DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN

HANDBOOK OF GRAPH THEORY

EDITED BY
JONATHAN L. GROSS
JAY YELLEN

CRC PRESS
Boca Raton London New York Washington, D.C.

DISCRETE MATHEMATICS AND ITS APPLICATIONS

Series Editor

Kenneth H. Rosen, Ph.D.

AT&T Laboratories
Middletown, New Jersey

Charles J. Colbourn and Jeffrey H. Dinitz, The CRC Handbook of Combinatorial Designs

Charalambos A. Charalambides, Enumerative Combinatorics

Steven Furino, Ying Miao, and Jianxing Yin, Frames and Resolvable Designs: Uses, Constructions, and Existence

Randy Goldberg and Lance Riek, A Practical Handbook of Speech Coders

Jacob E. Goodman and Joseph O'Rourke, Handbook of Discrete and Computational Geometry

Jonathan L. Gross and Jay Yellen, Graph Theory and Its Applications

Jonathan L. Gross and Jay Yellen, Handbook of Graph Theory

Darrel R. Hankerson, Greg A. Harris, and Peter D. Johnson, Introduction to Information Theory and Data Compression

Daryl D. Harms, Miroslav Kraetzl, Charles J. Colbourn, and John S. Devitt, Network Reliability: Experiments with a Symbolic Algebra Environment

David M. Jackson and Terry I. Visentin, An Atlas of Smaller Maps in Orientable and Nonorientable Surfaces

Richard E. Klima, Ernest Stitzinger, and Neil P. Sigmon, Abstract Algebra Applications with Maple

Patrick Knupp and Kambiz Salari, Verification of Computer Codes in Computational Science and Engineering

Donald L. Kreher and Douglas R. Stinson, Combinatorial Algorithms: Generation Enumeration and Search

Charles C. Lindner and Christopher A. Rodgers, Design Theory

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, Handbook of Applied Cryptography

Richard A. Mollin, Algebraic Number Theory

Richard A. Mollin, Fundamental Number Theory with Applications

Richard A. Mollin, An Introduction to Crytography

Richard A. Mollin, Quadratics

Continued Titles

Richard A. Mollin, RSA and Public-Key Cryptography

Kenneth H. Rosen, Handbook of Discrete and Combinatorial Mathematics

Douglas R. Shier and K.T. Wallenius, Applied Mathematical Modeling: A Multidisciplinary Approach

Douglas R. Stinson, Cryptography: Theory and Practice, Second Edition

Roberto Togneri and Christopher J. deSilva, Fundamentals of Information Theory and Coding Design

Lawrence C. Washington, Elliptic Curves: Number Theory and Cryptography

Library of Congress Cataloging-in-Publication Data

Handbook of graph theory / editors-in-chief, Jonathan L. Gross, Jay Yellen.
p. cm. — (Discrete mathematics and its applications)
Includes bibliographical references and index.
ISBN 1-58488-090-2 (alk. paper)
1. Graph theory—Handbooks, manuals, etc. I. Gross, Jonathan L. II. Yellen, Jay.
QA166.H36 2003
511'.5—dc22 2003065270

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

No claim to original U.S. Government works
International Standard Book Number 1-58488-090-2
Library of Congress Card Number 2003065270
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

PREFACE

Over the past forty years, graph theory has been one of the most rapidly growing areas of mathematics. Since 1960, more than 10,000 different authors have published papers classified as graph theory by *Math Reviews*, and for the past decade, over 1000 graph theory papers have been published each year.

This *Handbook* is intended to provide as comprehensive a view of graph theory as is feasible in a single volume. The 60 contributors to this volume collectively represent perhaps as much as 90% or more of the content areas in graph theory, including algorithmic and optimization approaches of graph theory as well as "pure" graph theory.

Format

In order to achieve this kind of comprehensiveness, we challenged our contributors to restrict their expository prose to a bare minimum by adhering to the ready-reference style of this CRC series, which emphasizes quick accessibility for the non-expert. We thank the contributors for responding so well to this challenge.

The 11 chapters of the *Handbook* are organized into 54 sections. Within each section, several major topics are presented. For each topic, there are lists of the essential definitions and facts, accompanied by examples, tables, remarks, and in some cases, conjectures and open problems. At the end of each section is a bibliography of references tied directly to that section. In many cases, these bibliographies are several pages long, providing extensive guides to the research literature and pointers to monographs.

In order that sections be reasonably self-contained, we encouraged contributors to include some definitions that may have appeared in earlier sections. Each contributor was asked to include a glossary with his or her section. These section glossaries were blended by the editors into 11 chapter glossaries.

Terminology and Notations

Graph theory has attracted mathematicians and scientists from diverse disciplines and, accordingly, is blessed (and cursed) with a proliferation of terminology and notations. Since the *Handbook* objective is to survey topics for persons whose expertise is elsewhere, either on other topics or outside of graph theory, we asked our contributors to tilt toward the general usage in the mathematical community, rather than staying strictly within the idioms of their specialities. But to understand the graph theory literature, it helps to accept the legacy of history. As editors, we tried to strike a balance between preserving the notation and terminology that evolved from each area's rich history and our desire create a cohesive, uniform body of material.

Some uniformity of usage came easily. In general, the word *graph* is used inclusively to refer to graphs with directed edges and/or to graphs with multi-edges and self-loops. In most sections, G denotes a graph and V and E denote its vertex- and edge-sets, respectively.

However, some words are used differently by different graph theory communities. For instance, to an algebraic graph theorist, a *Cayley graph* is simple, connected, and undirected, and to a topological graph theorist, it may be non-connected, possibly directed, and have multi-edges and/or self-loops. To some graph theorists, a *clique* is a complete subgraph, maximal under set inclusion, and to others maximality is not required.

Consistency in notation was also problematic. In graph coloring, the Greek letter χ denotes the chromatic number, and to an algebraic topologist, it means the euler characteristic.

Notes regarding terminology and notation were added to make explicit such differences and thereby improve cross-chapter compatibility.

Underscoring the difficulty of achieving uniformity is the co-editors' disagreement between themselves on when to place periods and commas inside and/or outside quotation marks. Yellen's concession on this issue comes with the expectation of yet-to-be-determined concessions granted to him by Gross on their next project.

Feedback

To see updates and to provide feedback and errata reports, visit the website *http://www.graphtheory.com*, and then follow the links to the webpage for this *Handbook*.

Acknowledgements

We would like to thank Bob Stern of CRC Press for his continued enthusiasm and patience during its gestation period and to thank Helena Redshaw, the Manager of Editorial Project Development at CRC, for her helpful support during the final hectic year. We would also like to thank Steve Maurer, who not only contributed a section to the *Handbook*, but also generously provided us with technical assistance at various stages of the project.

Jonathan Gross and Jay Yellen

Jonathan dedicates this book to Aaron, Jessie, Josh, Rena, and Alisa.

Jay dedicates this book to Betsey and Tara.

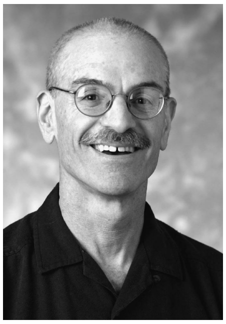
About the Editors

Jonathan Gross is Professor of Computer Science at Columbia University. His research in topology, graph theory, and cultural sociometry has earned him an Alfred P. Sloan Fellowship, an IBM Postdoctoral Fellowship, and various research grants from the Office of Naval Research, the National Science Foundation, and the Russell Sage Foundation.

Professor Gross has created and delivered numerous software-development short courses for Bell Laboratories and for IBM. These include mathematical methods for performance evaluation at the advanced level and for developing reusable software at a basic level. He has received several awards for outstanding teaching at Columbia University, including the career Great Teacher Award from the Society of Columbia Graduates. He appears on the Columbia Video Network and on the video network of the National Technological University.

His previous books include *Graph Theory and Its Applications*, coauthored with Jay Yellen, and *Topological Graph Theory*, coauthored with Thomas W. Tucker. Another previous book, *Measuring Culture*, coauthored with Steve Rayner, constructs network-theoretic tools for measuring sociological phenomena.

Prior to Columbia University, Professor Gross was in the Mathematics Department at Princeton University. His undergraduate work was at M.I.T., and he wrote his Ph.D. thesis on 3-dimensional topology at Dartmouth College.



Jay Yellen is Associate Professor of Mathematical Sciences at Rollins College. He received his B.S. and M.S. in Mathematics at Polytechnic University of New York and did his doctoral work in finite group theory at Colorado State University. Dr. Yellen has had regular faculty appointments at Allegheny College, State University of New York at Fredonia, and Florida Institute of Technology, where he was Chair of Operations Research from 1995 to 1999. He has had visiting appointments at Emory University, Georgia Institute of Technology, and Columbia University.

In addition to his book *Graph Theory and Its Applications*, coauthored with Jonathan L. Gross, he has written manuscripts used at IBM for two courses in discrete mathematics within the Principles of Computer Science Series and has contributed two sections to the *Handbook of Discrete and Combinatorial Mathematics*. He also has designed and conducted several summer workshops on creative problem-solving for secondary-school mathematics teachers, which were funded by the National Science Foundation and New York State. In April 2001 he received a Student's Choice Professor award at Rollins College.

Dr. Yellen has published research articles in character theory of finite groups, graph theory, power-system scheduling, and timetabling. His current research interests include graph theory, discrete optimization, and graph algorithms for software testing and course timetabling.

CONTRIBUTORS

Alfred V. Aho
Columbia University

Brian Alspach
University of Regina, Canada

Dan Archdeacon
University of Vermont

Lowell W. Beineke
Purdue University at Fort Wayne

Béla Bollobás
University of Memphis
Trinity College, Cambridge, UK

R. B. Borie
University of Alabama

Robert C. Brigham
University of Central Florida

Edmund Burke
University of Nottingham, UK

Gary Chartrand
Western Michigan University

Jianer Chen
Texas A&M University

Camil Demetrescu
Univ of Rome “La Sapienza”, Italy

A. K. Dewdney
University of Waterloo, Canada

Dominique de Werra
École Polytechnique Fédérale
de Lausanne, Switzerland

Michael Doob
University of Manitoba, Canada

Josep Fàbrega
Technical University of Catalonia, Spain

Ralph Faudree
University of Memphis

Irene Finocchi
University of Rome “Tor Vergata”, Italy

Miguel Angel Fiol
Technical University of Catalonia, Spain

Lisa Fleischer
IBM
Carnegie Mellon University

Herbert Fleischner
Technical University of Vienna, Austria

Alan Frieze
Carnegie Mellon University

Harold N. Gabow
University of Colorado

Mark Goldberg
Rensselaer Polytechnic Institute

Ronald J. Gould
Emory University

Jonathan L. Gross
Columbia University

Gregory Gutin
Royal Holloway, University of London, UK

Teresa W. Haynes
East Tennessee State University

Michael A. Henning
University of Natal, South Africa

Giuseppe F. Italiano
University of Rome “Tor Vergata”, Italy

Jeffrey Kingston
University of Sydney, Australia

Josef Lauri
University of Malta, Malta

Giuseppe Liotta
University of Perugia, Italy

Stephen B. Maurer
Swarthmore College

F. R. McMorris
Illinois Institute of Technology

Prakash Mirchandani
University of Pittsburgh

Seiya Negami
Yokohama National University, Japan

Vladimir Nikiforov
University of Memphis

James Oxley
Louisiana State University

R. Gary Parker
Georgia Institute of Technology

Tomaž Pisanski
University of Ljubljana, Slovenia

Michael Plummer
Vanderbilt University

Primož Potočnik
University of Ljubljana, Slovenia

K. B. Reid
California State University San Marcos

Douglas R. Shier
Clemson University

David Simchi-Levi
Massachusetts Institute of Technology

Clifford Stein
Columbia University

Paul K. Stockmeyer
College of William and Mary

Roberto Tamassia
Brown University

Krishnaiyan “KT” Thulasiraman
University of Oklahoma

C. A. Tovey
Georgia Institute of Technology

Alan Tucker
State University of New York Stony Brook

Thomas W. Tucker
Colgate University

Zsolt Tuza
Hungarian Academy of Sciences
University of Veszprém, Hungary

Andrew Vince
University of Florida

Mark E. Watkins
Syracuse University

Arthur T. White
Western Michigan University

Robin J. Wilson
The Open University, UK

Nicholas Wormald
University of Waterloo, Canada

Jay Yellen
Rollins College

Ping Zhang
Western Michigan University

CONTENTS

Preface

1. INTRODUCTION to GRAPHS 1

1.1 Fundamentals of Graph Theory . . . 2
– *Jonathan L. Gross and Jay Yellen*

1.2 Families of Graphs and Digraphs . . . 20
– *Lowell W. Beineke*

1.3 History of Graph Theory . . . 29
– *Robin J. Wilson*

Glossary . . . 50

2. GRAPH REPRESENTATION 56

2.1 Computer Representation of Graphs . . . 57
– *Alfred V. Aho*

2.2 The Graph Isomorphism Problem . . . 68
– *Mark Goldberg*

2.3 The Reconstruction Problem . . . 79
– *Josef Lauri*

2.4 Recursively Constructed Graphs . . . 99
– *R. B. Borie, R. Gary Parker, and C. A. Tovey*

Glossary . . . 119

3. DIRECTED GRAPHS 126

3.1 Basic Digraph Models and Properties . . . 127
– *Jay Yellen*

3.2 Directed Acyclic Graphs . . . 142
– *Stephen B. Maurer*

3.3 Tournaments . . . 156
– *K. B. Reid*

Glossary . . . 185

4. CONNECTIVITY and TRAVERSABILITY 193

4.1 Connectivity: Properties and Structure . . . 194
– *Josep Fàbrega and Miguel Angel Fiol*

4.2 Eulerian Graphs . . . 214
– *Herbert Fleischner*

4.3 Chinese Postman Problems . . . 237
– *R. Gary Parker*

4.4 DeBruijn Graphs and Sequences . . . 253
– *A. K. Dewdney*

4.5 Hamiltonian Graphs . . . 261
– *Ronald J. Gould*

4.6 Traveling Salesman Problems . . . 279
– *Gregory Gutin*

4.7 Further Topics in Connectivity . . . 300
– *Josep Fàbrega and Miguel Angel Fiol*

Glossary . . . 330

5. COLORINGS and RELATED TOPICS 340

5.1 Graph Coloring 341
– *Zsolt Tuza*

5.2 Further Topics in Graph Coloring 365
– *Zsolt Tuza*

5.3 Independent Sets and Cliques 389
– *Gregory Gutin*

5.4 Factors and Factorization 403
– *Michael Plummer*

5.5 Perfect Graphs 431
– *Alan Tucker*

5.6 Applications to Timetabling 445
– *Edmund Burke, Dominique de Werra, and Jeffrey Kingston*

Glossary 475

6. ALGEBRAIC GRAPH THEORY 484

6.1 Automorphisms 485
– *Mark E. Watkins*

6.2 Cayley Graphs 505
– *Brian Alspach*

6.3 Enumeration 516
– *Paul K. Stockmeyer*

6.4 Graphs and Vector Spaces 533
– *Krishnaiyan "KT" Thulasiraman*

6.5 Spectral Graph Theory 557
– *Michael Doob*

6.6 Matrioidal Methods in Graph Theory 574
– *James Oxley*

Glossary 599

7. TOPOLOGICAL GRAPH THEORY 610

7.1 Graphs on Surfaces 611
– *Tomaž Pisanski and Primož Potočnik*

7.2 Minimum and Maximum Imbeddings 625
– *Jianer Chen*

7.3 Genus Distributions 642
– *Jonathan L. Gross*

7.4 Voltage Graphs 661
– *Jonathan L. Gross*

7.5 The Genus of a Group 684
– *Thomas W. Tucker*

7.6 Maps 696
– *Andrew Vince*

7.7 Representativity 722
– *Dan Archdeacon*

7.8 Triangulations 737
– *Seiya Negami*

7.9 Graphs and Finite Geometries 761
– *Arthur T. White*

Glossary 770

8. ANALYTIC GRAPH THEORY 787

8.1 Extremal Graph Theory 788
– *Béla Bollobás and Vladimir Nikiforov*

8.2 Random Graphs 817
– *Nicholas Wormald*

8.3 Ramsey Graph Theory 837
– *Ralph Faudree*

8.4 The Probabilistic Method 860
– *Alan Frieze*

Glossary 868

9. GRAPHICAL MEASUREMENT 872

9.1 Distance in Graphs 873
– *Gary Chartrand and Ping Zhang*

9.2 Domination in Graphs 889
– *Teresa W. Haynes and Michael A. Henning*

9.3 Tolerance Graphs 910
– *F. R. McMorris*

9.4 Bandwidth 922
– *Robert C. Brigham*

Glossary 945

10. GRAPHS in COMPUTER SCIENCE 952

10.1 Searching 953
– *Harold N. Gabow*

10.2 Dynamic Graph Algorithms 985
– *Camil Demetrescu, Irene Finocchi, and Guiseppe F. Italiano*

10.3 Drawings of Graphs 1015
– *Giuseppe Liotta and Roberto Tamassia*

10.4 Algorithms on Recursively Constructed Graphs 1046
– *R. B. Borie, R. Gary Parker, and C. A. Tovey*

Glossary 1067

11. NETWORKS and FLOWS 1074

11.1 Maximum Flows 1075
– *Clifford Stein*

11.2 Minimum Cost Flows 1087
– *Lisa Fleischer*

11.3 Matchings and Assignments 1103
– *Douglas R. Shier*

11.4 Communication Network Design Models 1117
– *Prakash Mirchandani and David Simchi-Levi*

Glossary 1139

Chapter 1

INTRODUCTION TO GRAPHS

1.1 FUNDAMENTALS of GRAPH THEORY

Jonathan L. Gross, Columbia University
Jay Yellen, Rollins College

1.2 FAMILIES of GRAPHS and DIGRAPHS

Lowell W. Beineke, Purdue University at Fort Wayne

1.3 HISTORY of GRAPH THEORY

Robin J. Wilson, The Open University, UK

GLOSSARY

1.1 FUNDAMENTALS OF GRAPH THEORY

Jonathan L. Gross, Columbia University
Jay Yellen, Rollins College

1.1.1 Graphs and Digraphs
1.1.2 Degree and Distance
1.1.3 Basic Structural Concepts
1.1.4 Trees
References

Introduction

Configurations of nodes and connections occur in a great diversity of applications. They may represent physical networks, such as electrical circuits, roadways, or organic molecules. They are also used in representing less tangible interactions as might occur in ecosystems, sociological relationships, databases, or in the flow of control in a computer program.

1.1.1 Graphs and Digraphs

Any mathematical object involving points and connections between them may be called a *graph*. If all the connections are unidirectional, it is called a *digraph*. Our highly inclusive definition in this initial section of the *Handbook* permits fluent discussion of almost any particular modification of the basic model that has ever been called a graph.

Basic Terminology

DEFINITIONS

D1: A ***graph*** $G = (V, E)$ consists of two sets V and E.

- The elements of V are called ***vertices*** (or ***nodes***).
- The elements of E are called ***edges***.
- Each edge has a set of one or two vertices associated to it, which are called its ***endpoints***. An edge is said to ***join*** its endpoints.

NOTATION: The subscripted notations V_G and E_G (or $V(G)$ and $E(G)$) are used for the vertex- and edge-sets when G is not the only graph under consideration.

D2: If vertex v is an endpoint of edge e, then v is said to be ***incident*** on e, and e is incident on v.

D3: A vertex u is ***adjacent*** to vertex v if they are joined by an edge.

D4: Two adjacent vertices may be called ***neighbors***.

D5: ***Adjacent edges*** are two edges that have an endpoint in common.

D6: A ***proper edge*** is an edge that joins two distinct vertices.

D7: A ***multi-edge*** is a collection of two or more edges having identical endpoints.

D8: A ***simple adjacency*** between vertices occurs when there is exactly one edge between them.

D9: The ***edge-multiplicity*** between a pair of vertices u and v is the number of edges between them.

D10: A ***self-loop*** is an edge that joins a single endpoint to itself.

TERMINOLOGY: An alternative word for "self-loop" is "loop". This can be used in contexts in which "loop" has no other meanings.

TERMINOLOGY: In computer science, the word "graph" is commonly used either to mean a graph as defined here, or to mean a computer-represented data structure whose value is a graph.

EXAMPLE

E1: A line drawing of a graph $G = (V, E)$ is shown in Figure 1.1.1. It has vertex-set $V = \{u, v, w, x\}$ and edge-set $E = \{a, b, c, d, e, f\}$. The set $\{a, b\}$ is a multi-edge with endpoints u and v, and edge c is a self-loop.

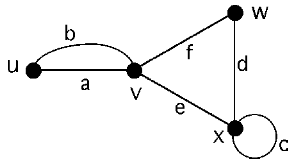


Figure 1.1.1 **A graph.**

REMARKS

R1: A graph is realized in a plane or in 3-space as a set of points, representing the vertices, and a set of curved or straight line segments, representing the edges. The curvature or length of such a line segment is irrelevant to the meaning. However, if a *direction* is indicated, that is significant.

R2: Occasionally, a graph is *parametrized* so that each edge is regarded as the homeomorphic image of the real interval $[0, 1]$ (except that for a self-loop, the endpoints 0 and 1 have the same image).

Simple Graphs

Most of theoretical graph theory is concerned with *simple* graphs. This is partly because many problems regarding general graphs can be reduced to problems about simple graphs.

DEFINITIONS

D11: A ***simple graph*** is a graph that has no self-loops or multi-edges.

D12: A ***trivial graph*** is a graph consisting of one vertex and no edges.

D13: A ***null graph*** is a graph whose vertex- and edge-sets are empty.

Edge Notation for Simple Adjacencies and for Multi-edges

NOTATION: An edge joining vertices u and v of a graph may be denoted by the juxtaposition uv if it is the only such edge. Occasionally, the ordered pair (u, v) is used in this situation, instead of uv. To avoid ambiguities when multi-edges exist, or whenever else desired, the edges of a general graph may be given their own names, as in Figure 1.1.1 above.

EXAMPLE

E2: The simple graph shown in Figure 1.1.2 has edge-set $E = \{uv, vw, vx, wx\}$.

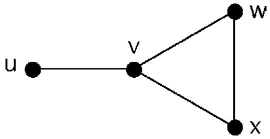


Figure 1.1.2 **A simple graph.**

General Graphs

Many applications require non-simple graphs as models. Moreover, some non-simple graphs serve an essential role in theoretical constructions, especially in constructing graph drawings (simple and non-simple) on surfaces (see Chapter 7).

TERMINOLOGY NOTE: Although the term "graph" means that self-loops and multi-edges are allowed, sometimes, for emphasis, the term ***general graph*** is used.

DEFINITIONS

D14: A ***loopless graph*** is a graph that has no self-loops. (It might have multi-edges.) Sometimes a loopless graph is referred to as a ***multigraph***.

D15: The ***dipole*** D_n is a loopless graph with two vertices and n edges joining them.

D16: The ***bouquet*** B_n is a graph with one vertex and n self-loops.

EXAMPLES

E3: The loopless graph in Figure 1.1.3 depicts the benzene molecule C_6H_6.

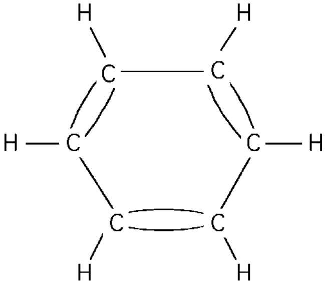


Figure 1.1.3 **Graph model for a benzene ring.**

E4: The dipole D_3 is shown in Figure 1.1.4.

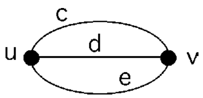


Figure 1.1.4 The loopless graph D_3.

E5: Two graphs with self-loops are shown in Figure 1.1.5.

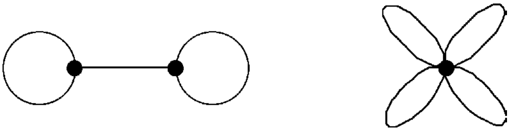


Figure 1.1.5 The dumbbell graph and the bouquet B_4.

Attributes

Allowing graphs to have additional attributes beyond vertices and edges enables them to serve as mathematical models for a wide variety of applications. Two of the most common additional edge attributes, both described in great detail later in the *Handbook*, are edge *direction* (e.g., Chapters 3 and 11) and edge *weight* (e.g., Chapters 4 and 11). Another common attribute (for edges or vertices) is *color*. Graph coloring is discussed in Chapter 5.

DEFINITIONS

D17: A ***vertex attribute*** is a function from the vertex-set to some set of possible attribute values.

D18: An ***edge attribute*** is a function from the edge-set to some set of possible attribute values.

Digraphs

An edge between two vertices creates a connection in two opposite senses at once. Assigning a direction makes one of these senses *forward* and the other *backward*. Viewing direction as an edge attribute is partly motivated by its impact on computer implementations of graph algorithms. Moreover, from a mathematical perspective, regarding *directed graphs* as augmented graphs makes it easier to view certain results that tend to be established separately for graphs and for digraphs as a single result that applies to both. The attribute of edge direction is developed extensively in Chapter 3 and elsewhere in this *Handbook*.

DEFINITIONS

D19: A ***directed edge*** (or ***arc***) is an edge e, one of whose endpoints is designated as the ***tail***, and whose other endpoint is designated as the ***head***. They are denoted $head(e)$ and $tail(e)$, respectively.

TERMINOLOGY: A directed edge is said to be ***directed from*** its tail and ***directed to*** its head. (The tail and the head of a directed self-loop are the same vertex.)

NOTATION: In a line drawing, the arrow points toward the head.

D20: A ***multi-arc*** is a set of two or more arcs having the same tail and same head.

D21: A ***digraph*** (or ***directed graph***) is a graph each of whose edges is directed.

D22: A ***simple digraph*** is a digraph with no self-loops and no multi-arcs.

D23: A ***mixed graph*** (or ***partially directed graph***) is a graph that has both undirected and directed edges. In a mixed graph, using the unmodified term "edge" avoids specifying whether the edge is directed or undirected.

D24: The ***underlying graph*** of a directed or partially directed graph G is the graph that results from removing all the designations of *head* and *tail* from the directed edges of G (i.e., deleting all the edge-directions).

Ordered-Pair Representation of Arcs

NOTATION: In a simple digraph, an arc from vertex u to vertex v is commonly denoted (u, v) (or sometimes uv). When multi-arcs are possible, using distinct names is often necessary.

COMPUTATIONAL NOTE: (*A caution to software designers*) From the perspective of object-oriented software design, the ordered-pair representation of arcs in a digraph treats digraphs as a different class of objects from graphs. This could seriously undermine *software reuse*. Large portions of computer code might have to be rewritten in order to adapt an algorithm that was originally designed for a digraph to work on an undirected graph.

The ordered-pair representation could also prove awkward in implementing algorithms for which the graphs or digraphs are *dynamic* structures (i.e., they change during the algorithm). Whenever the direction on a particular edge must be reversed, the associated ordered pair has to be deleted and replaced by its reverse. Even worse, if a directed edge is to become undirected, then an ordered pair must be replaced with an unordered pair. Similarly, the undirected and directed edges of a partially directed graph would require two different types of objects.

EXAMPLES

E6: The digraph on the left in Figure 1.1.6 has the undirected graph on the right as its underlying graph. The digraph has two multi-arcs: $\{a, b\}$ and $\{f, h\}$.

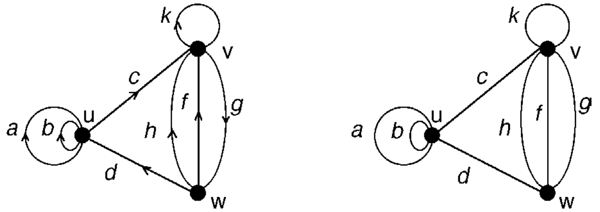


Figure 1.1.6 **A digraph and its underlying graph.**

E7: A simple digraph can have one arc in each direction between two vertices.

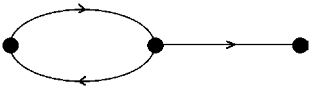


Figure 1.1.7 **A simple digraph whose underlying graph is not simple.**

Vertex-Coloring

When the vertex-set of a graph is partitioned, the cells of the partition are commonly assigned distinct *colors*. This is developed at length in Chapter 5.

DEFINITIONS

D25: A ***vertex-coloring*** of a graph G is a function from its vertex-set V_G vertices to a set C whose elements are called *colors*.

D26: A vertex-coloring is ***proper*** if two adjacent vertices are always assigned different colors.

D27: A graph is ***c-colorable*** if it has a proper vertex-coloring with c or fewer colors.

D28: The (***vertex***) ***chromatic number of a graph*** G, denoted $\chi(G)$, is the smallest number c of colors such that G is c-colorable.

REMARK

R3: Definitions of ***edge-coloring***, ***c-edge-colorable***, and ***edge-chromatic number***, denoted $\chi'(G)$, are obtained by simply replacing the word "vertices" in the definitions above with the word "edges".

EXAMPLE

E8: The graph G in Figure 1.1.8 is shown with a 3-coloring of its vertex-set. Since it is not 2-colorable, its chromatic number is 3. Also, the graph is easily seen to be 3-edge-colorable and clearly is not 2-edge-colorable; hence, $\chi'(G) = 3$.

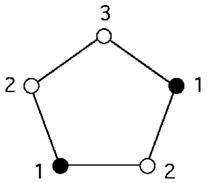


Figure 1.1.8 **A graph G with $\chi(G) = \chi'(G) = 3$.**

1.1.2 Degree and Distance

Two of the most fundamental notions in graph theory are those of the *degree* of a vertex and the *distance* between two vertices. Distance is developed fully in §9.1.

Degree

DEFINITIONS

D29: The ***degree*** (or ***valence***) of a vertex v in a graph G, denoted $deg(v)$, is the number of proper edges incident on v plus twice the number of self-loops. (For simple graphs, of course, the degree is simply the number of neighbors.)

TERMINOLOGY: Applications of graph theory to physical chemistry motivate the use of the term ***valence*** as an alternative to *degree*. Thus, a vertex of degree d is also called a ***d*-valent vertex**.

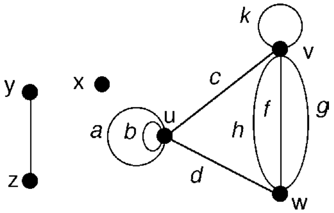
D30: The ***degree sequence*** of a graph is the sequence formed by arranging the vertex degrees into non-decreasing order.

D31: The ***indegree*** of a vertex v in a digraph is the number of arcs directed to v; the ***outdegree*** of vertex v is the number of arcs directed from v. Each self-loop at v counts one toward the indegree of v and one toward the outdegree.

D32: An ***isolated vertex*** in a graph is a vertex of degree 0.

EXAMPLES

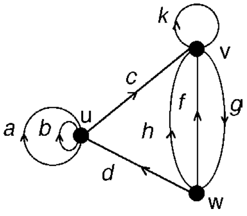
E9: Figure 1.1.9 shows a graph and its degree sequence.



$< 0, 1, 1, 4, 6, 6, >$
x y z w u v

Figure 1.1.9 **A graph and its degree sequence.**

E10: Figure 1.1.10 shows the indegrees and outdegrees of a digraph.



vertex	u	v	w
indegree	3	4	1
outdegree	3	2	3

Figure 1.1.10 **The indegrees and outdegrees of the vertices of a digraph.**

FACTS

For proofs of the following elementary facts, see [GrYe99, §1.1] or other basic texts.

F1: (Euler) The sum of the degrees of the vertices of a graph is twice the number of edges.

F2: In every graph, the number of vertices having odd degree is an even number.

F3: A non-trivial simple graph G must have at least one pair of vertices whose degrees are equal.

F4: In a digraph, the sum of the indegrees and the sum of the outdegrees both equal the number of edges.

F5: The degree sequence of a graph is a finite, non-decreasing sequence of nonnegative integers whose sum is even.

F6: Conversely, any non-decreasing, nonnegative sequence of integers whose sum is even is the degree sequence of some graph, but not necessarily of a simple graph.

Walks, Trails, and Paths

DEFINITIONS

D33: A ***walk*** in a graph G is an alternating sequence of vertices and edges,

$$W = v_0, e_1, v_1, e_1, \ldots, e_n, v_n$$

such that for $j = 1, \ldots, n$, the vertices v_{j-1} and v_j are the endpoints of the edge e_j. If, moreover, the edge e_j is directed from v_{j-1} to v_j, then W is a ***directed walk***.

- In a simple graph, a walk may be represented simply by listing a sequence of vertices: $W = v_0, v_1, \ldots, v_n$ such that for $j = 1, \ldots, n$, the vertices v_{j-1} and v_j are adjacent.
- The ***initial vertex*** is v_0.
- The ***final vertex*** (or ***terminal vertex***) is v_n.
- An ***internal vertex*** is a vertex that is neither initial nor final.

D34: The ***length of a walk*** is the number of edges (counting repetitions).

D35: A walk is ***closed*** if the initial vertex is also the final vertex; otherwise, it is ***open***.

D36: A ***trail*** in a graph is a walk such that no edge occurs more than once.

D37: An ***eulerian trail*** in a graph G is a walk that contains each edge of G *exactly* once. (See §4.2.)

D38: A ***path*** in a graph is a trail such that no internal vertex is repeated.

D39: A ***cycle*** is a closed path of length at least 1.

D40: A ***trivial*** walk, trail, or path consists of a single vertex and no edges.

EXAMPLE

E11: In the graph shown in Figure 1.1.11, the vertex sequence $\langle u, v, x, v, z \rangle$ represents a walk that is not a trail, and the vertex sequence $\langle u, v, x, y, v, z \rangle$ represents a trail that is not a path.

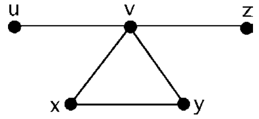


Figure 1.1.11

Distance and Connectivity

D41: The ***distance between two vertices*** in a graph is the length of the shortest walk between them.

D42: The ***directed distance from*** a vertex u ***to*** a vertex v in a digraph is the length of the shortest directed walk from u to v.

D43: A graph is ***connected*** if between every pair of vertices there is a walk.

D44: A digraph is ***(weakly) connected*** if its underlying graph is connected.

D45: A digraph is ***strongly connected*** if from each vertex to each other vertex there is a directed walk.

D46: The ***eccentricity*** of a vertex v in a connected graph is its distance to a vertex farthest from v.

D47: The ***radius*** of a connected graph is its minimum eccentricity.

D48: The ***diameter*** of a connected graph is its maximum eccentricity.

EXAMPLE

E12: The digraph shown on the left in Figure 1.1.12 is strongly connected; the digraph on the right is connected but not strongly connected.

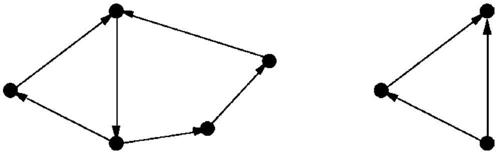


Figure 1.1.12 **A strongly connected digraph and a weakly connected one.**

1.1.3 Basic Structural Concepts

We are concerned with the possible equivalence of two graphs, with the symmetries of an individual graph, and with the possible appearance of one graph within another graph.

Isomorphism

In concept, two graphs are *isomorphic* if they are structurally identical, which means that they correspond in all structural details. A formal vertex-to-vertex and edge-to-edge correspondence is called an *isomorphism.*

DEFINITIONS

D49: An ***isomorphism between two simple graphs*** G and H is a vertex bijection $\phi : V_G \to V_H$ such that for $u, v \in V_G$, the vertex u is adjacent to the vertex v in graph G if and only if $\phi(u)$ is adjacent to $\phi(v)$ in graph H. Implicitly, there is also an edge bijection $E_G \to E_H$ such that $uv \mapsto \phi(u)\phi(v)$.

D50: An ***isomorphism between two general graphs*** G and H is a pair of bijections $\phi_V : V_G \to V_H$ and $\phi_E : E_G \to E_H$ such that for every pair of vertices $u, v \in V_G$, the set of edges in E_G joining u and v is mapped bijectively to the set of edges in E_H joining the vertices $\phi(u)$ and $\phi(v)$.

D51: We say that G and H are ***isomorphic graphs*** and we write $G \cong H$ if there is an isomorphism $G \to H$.

D52: An ***adjacency matrix*** for a simple graph G whose vertices are explicitly ordered $v_1, v_2, \ldots, v_n$ is the $n \times n$ matrix A_G such that

$$A_G(i,j) = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

D53: A property associated with all graphs is an ***isomorphism invariant*** if it has the same value (or is the same) for any two isomorphic graphs.

EXAMPLES

E13: The two graphs in Figure 1.1.13 are isomorphic under the mapping

$$u_1 \mapsto v_1 \quad u_2 \mapsto v_1 \quad u_3 \mapsto v_4 \quad u_4 \mapsto v_3$$

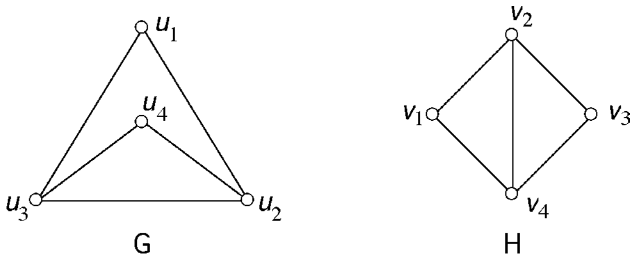


Figure 1.1.13 **Two isomorphic graphs.**

If one flips vertex u_4 of graph G downward to the bottom and rotates the figure a quarter-turn counterclockwise, then the resulting image of graph G "looks just like" graph H. Their adjacency matrices are:

$$A_G = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array} \qquad A_H = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

We observe that transposing rows u_3 and u_4 and also transposing columns u_3 and u_4 transforms the matrix A_G into matrix A_H.

E14: The two graphs in Figure 1.1.14 are isomorphic, even if the drawings look quite different. The vertex-labels indicate an isomorphism.

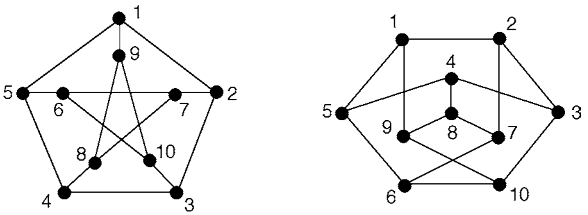


Figure 1.1.14 **Two isomorphic graphs that look quite different.**

E15: Figure 1.1.15 shows two non-isomorphic graphs with identical degree sequences. (It is easy to show that connectedness is an isomorphism invariant.)

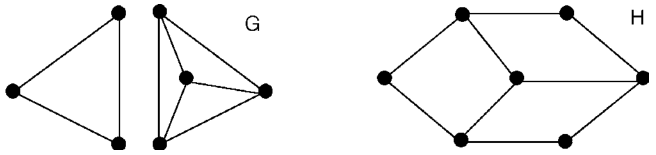


Figure 1.1.15 **Two graphs whose degree sequences are both $\langle 2,2,2,3,3,3,3\rangle$.**

FACTS

F7: Considering all possible bijections of the vertex-sets of two n-vertex graphs requires $O(n!)$ steps.

F8: Although some fast heuristics are known (see §2.2), there is no known polynomial-time algorithm for testing graph isomorphism.

F9: The number of vertices, the number of edges, and the degree sequence are all isomorphism invariants. On the other hand, having the same values for all three of these invariants does not imply that two graphs are isomorphic, as illustrated by Example 15.

F10: Each row sum (and column sum) in an adjacency matrix equals the degree of the corresponding vertex.

Automorphisms

The notion of symmetry in a graph is formalized in terms of isomorphisms of the graph to itself.

DEFINITIONS

D54: A ***graph automorphism*** is an isomorphism of the graph to itself.

D55: The ***orbit of a vertex*** u of a graph G is the set of all vertices $v \in V_G$ such that there is an automorphism ϕ such that $\phi(u) = v$.

D56: The ***orbit of an edge*** d of a graph G is the set of all edges $e \in E_G$ such that there is an automorphism ϕ such that $\phi(d) = e$.

D57: A graph is ***vertex-transitive*** if all the vertices are in the same orbit.

D58: A graph is ***edge-transitive*** if all the edges are in the same orbit.

FACTS

F11: The vertex orbits partition the vertex-set of a graph.

F12: The edge orbits partition the edge-set of a graph.

EXAMPLE

E16: For the graph on the left in Figure 1.1.16, the vertex orbits are $\{u_1, u_4\}$ and $\{u_2, u_3\}$, and the edge orbits are $\{u_1u_2, u_1u_3, u_2u_4, u_3u_4\}$ and $\{u_2u_3\}$. The graph on the right is vertex-transitive and edge-transitive.

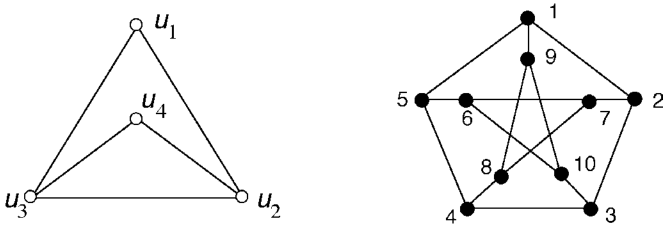


Figure 1.1.16

Subgraphs

DEFINITIONS

D59: A subgraph of a graph G is a graph H such that $V_H \subset V_G$ and $E_H \subset E_G$. (Usually, any graph isomorphic to a subgraph of G is also said to be a subgraph of G.)

D60: In a graph G, the ***induced subgraph*** on a set of vertices $W = \{w_1, \ldots, w_k\}$, denoted $G(W)$, has W as its vertex-set, and it contains every edge of G whose endpoints are in W. That is,

$$V(G(W)) = W \text{ and } E(G(W)) = \{e \in E(G) \mid \text{the endpoints of edge } e \text{ are in } W\}$$

D61: A subgraph H of a graph G is a ***spanning subgraph*** if $V(H) = V(G)$. (Also, if H is isomorphic to a spanning subgraph of G, we may say that H spans G.)

D62: A ***component*** of a graph G is a connected subgraph H such that no subgraph of G that properly contains H is connected. In other words, a component is a *maximal* connected subgraph.

EXAMPLE

E17: For the graph G in Figure 1.1.17, H_1 is a spanning subgraph but not an induced subgraph, and H_2 is an induced subgraph but not a spanning subgraph.

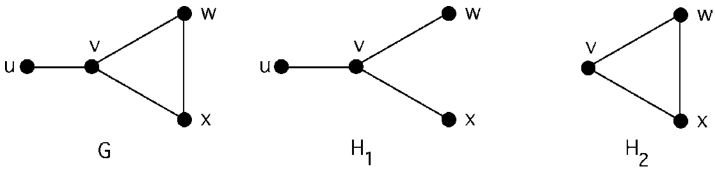


Figure 1.1.17 **A spanning subgraph H_1 and an induced subgraph H_2.**

FACTS

F13: Let $\phi : G \to H$ be a graph isomorphism, and let J be a subgraph of G. Then the restriction of ϕ to the subgraph J is an isomorphism onto its image $\phi(J)$.

F14: If a graph J is a subgraph of a graph G but not a subgraph of a graph H, then $G \not\cong H$. This is a corollary of Fact 13.

Graph Operations

The operations of adding and deleting vertices and edges of a graph are regarded as *primary operations*, because they are the foundation for other operations, which may be called *secondary operations*.

DEFINITIONS

D63: The operation of ***adding the vertex*** u to a graph $G = (V, E)$, such that $u \notin V$, yields a new graph with vertex set $V \cup \{u\}$ and edge set E, which is denoted $G \cup \{u\}$. (The new vertex u has no neighbors.)

D64: The operation of ***deleting the vertex*** u from a graph $G = (V, E)$ not only removes the vertex u but also removes every edge of which u is an endpoint. The resulting graph is denoted $G - u$.

D65: The operation of ***adding an edge*** d (or uv) to a graph $G = (V, E)$ joining the vertices u and v yields a new graph with vertex set V and edge set $E \cup \{d\}$ (or $E \cup \{uv\}$), which is denoted $G \cup \{d\}$ (or $G \cup \{uv\}$).

D66: The operation of ***deleting an edge*** d (or uv) from a graph $G = (V, E)$ removes only that edge. The resulting graph is denoted $G - d$ (or $G - uv$).

D67: A ***cut-vertex*** (or ***cutpoint***) is a vertex whose removal increases the number of components.

D68: A ***cut-edge*** is an edge whose removal increases the number of components.

D69: The ***edge-complement*** of a simple graph G is the graph $\overline{G}$ (alternatively denoted G^c) that has the same vertex set as G, such that uv is an edge of $\overline{G}$ if and only if it is *not* an edge of G.

D70: The ***join*** (or ***suspension***) of two graphs G and H is denoted by $G + H$.

$$V(G+H) = V(G) \cup V(H)$$
$$E(G+H) = E(G) \cup E(H) \cup \{uv \mid u \in V(G) \text{ and } v \in V(H)\}$$

D71: The ***cartesian product*** (or ***product***) of two graphs G and H is denoted by $G \times H$.

$$V(G \times H) = V(G) \times V(H)$$
$$E(G \times H) = E(G) \times V(H) \cup V(G) \times E(H)$$

The endpoints of the edge $(d, v) \in E(G) \times V(H)$ are the vertices (x, v) and (y, v), where x and y are the endpoints of edge $d \in E(G)$. The endpoints of the edge $(u, e) \in V(G) \times E(H)$ are the vertices (u, s) and (u, t), where s and t are the endpoints of edge $e \in E(H)$.

D72: The ***graph union*** of two graphs G and H is the graph $G \cup H$ whose vertex-set and edge-set are the disjoint unions, respectively, of the vertex-sets and the edge-sets of G and H.

D73: The **m-*fold self-union*** mG is the iterated disjoint union $G \cup \cdots \cup G$ of m copies of the graph G.

EXAMPLES

E18: Figure 1.1.18 illustrates the operation of edge-complementation.

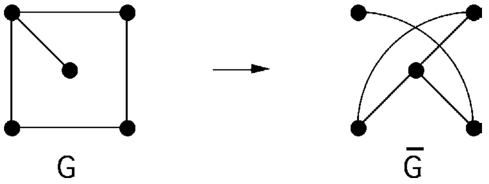


Figure 1.1.18 **Edge-complementation.**

E19: Figure 1.1.19 illustrates the join operation.

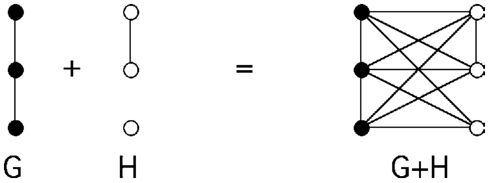


Figure 1.1.19 **Join operation.**

E20: Figure 1.1.20 illustrates the product operation.

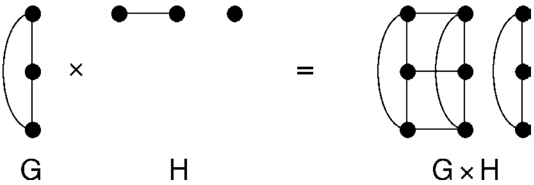


Figure 1.1.20 **Cartesian product.**

E21: In Figure 1.1.19 above, the vertex in the upper left corner of the drawing of the graph G is a cut-vertex, and the edge from that vertex to the center vertex is a cut-edge.

1.1.4 Trees

Trees are important to the structural understanding of graphs and to the algorithmics of information processing, and they play a central role in the design and analysis of connected networks. A standard characterization theorem for trees appears in §1.2.

Acyclic Graphs

DEFINITIONS

D74: A ***tree*** is a connected graph with no cycles (i.e., *acyclic*).

D75: A ***forest*** is a (not necessarily connected) graph with no cycles.

D76: A ***central vertex*** in a graph is a vertex whose eccentricity equals the radius of the graph.

D77: The ***center*** of a graph is the subgraph induced on its set of central vertices.

TERMINOLOGY NOTE: Classically (see §1.3), the words *center* and *bicenter* were used to mean the set of central vertices of a tree, when there was only one vertex or two vertices, respectively. (See Fact 15 below.)

EXAMPLE

E22: The graph on the left in Figure 1.1.21 is a tree; the other two graphs are not.

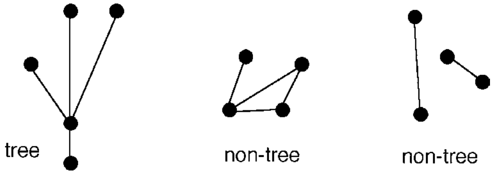


Figure 1.1.21 A tree and two non-trees.

FACT

F15: The center of a tree is isomorphic to K_1 or to K_2. (See §1.3 for information about the historical context of this fact.)

Trees as Subgraphs

Several different problem-solving algorithms involve growing a tree within a graph, one edge and one vertex at a time. All these techniques are refinements and extensions of the same basic tree-growing scheme given in this section.

DEFINITIONS

TERMINOLOGY: For a given tree T in a graph G, the edges and vertices of T are called ***tree edges*** and ***tree vertices***, and the edges and vertices of G that are not in T are called ***non-tree edges*** and ***non-tree vertices***.

D78: A ***frontier edge*** for a given tree T in a graph is a non-tree edge with one endpoint in T and one endpoint not in T.

D79: A ***spanning tree*** of a graph G is a spanning subgraph of G that is a tree.

EXAMPLE

E23: For the graph in Figure 1.1.22, the tree edges of a tree T are drawn in bold. The tree vertices are black, and the non-tree vertices are white. The frontier edges for T, appearing as dashed lines, are edges a, b, c, and d. The plain edges are the non-tree edges that are not frontier edges for T.

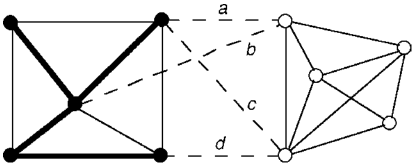


Figure 1.1.22 A tree with frontier edges a, b, c, and d.

Observe that when any one of the frontier edges in Figure 1.1.22 is added to the tree T, the resulting subgraph is still a tree. This property holds in general, and applying it iteratively forms the core of the tree-growing scheme of this section.

FACT

F16: Let T be a tree in a graph G, and let e be a frontier edge for T. Then the subgraph of G formed by adding edge e to tree T is a tree. (Formally, adding frontier edge e to a tree involves adding a new vertex to current tree T, i.e., its non-tree endpoint.)

Basic Tree-Growing Algorithm

The basic tree-growing scheme uses vertex labels to keep track of the order in which vertices are added to the tree.

TERMINOLOGY: A ***standard (0-based) vertex-labeling*** of an n-vertex graph is a one-to-one assignment of the integers $0, 1, \ldots, n-1$ to the vertices of that graph.

Algorithm 1.1.1: Basic Tree-Growing with Vertex Labels

Input: a graph G and a starting vertex $v \in V_G$.
Output: a spanning tree T of $C_G(v)$ and a standard vertex-labeling of $C_G(v)$.

```
Initialize tree $T$ as vertex $v$.
Write label 0 on vertex $v$.
Initialize label counter $i := 1$
While tree $T$ does not yet span component $C_G(v)$
    Choose a frontier edge $e$ for tree $T$.
    Let $w$ be the endpoint of edge $e$ that lies outside of $T$.
    Add edge $e$ and vertex $w$ to tree $T$.
    Write label $i$ on vertex $w$.
    $i := i + 1$
Return tree $T$ and vertex-labeling of $C_G(v)$.
```

REMARK

R4: [**Uniqueness of the Output Tree from Tree-Growing**]
Without a rule for choosing a frontier edge (including a way to break ties), the output tree from Algorithm 1.1.1 would not be unique (in which case, many computer scientists would hesitate to use the term *algorithm*). The uniqueness of the output depends on some *default priority* based on the ordering of the edges (and vertices) in the data

structure chosen to implement the algorithm. The default priority is used whenever no other rule is given and as a way of breaking ties left from other rules.

FACTS

F17: If an execution of the basic tree-growing algorithm starts at vertex v of a graph G, then the subgraph consisting of the labeled vertices and tree edges is a spanning tree of the component $C_G(v)$.

F18: A graph is connected if and only if the basic tree-growing algorithm labels all its vertices.

Prioritizing the Edge Selection

The edge-prioritized tree-growing algorithm below is a refinement of basic tree-growing.

Algorithm 1.1.2: Edge-Prioritized Tree-Growing

Input: a connected graph G, a starting vertex $v \in V_G$,
and a rule for prioritizing frontier edges.
Output: a spanning tree T and a standard vertex-labeling of V_G.

```
Initialize tree $T$ as vertex $v$.
Initialize the set of frontier edges for tree $T$ as empty.
Write label 0 on vertex $v$.
Initialize label counter $i := 1$
While tree $T$ does not yet span $G$
    Update the set of frontier edges for $T$.
    Let $e$ be the frontier edge for $T$ of highest priority.
    Let $w$ be the unlabeled endpoint of edge $e$.
    Add edge $e$ (and vertex $w$) to tree $T$.
    Write label $i$ on vertex $w$.
    $i := i + 1$
Return tree $T$ with its vertex-labeling.
```

FACT

F19: Different rules for prioritizing the frontier edges give rise to different spanning trees: the *depth-first search* tree (*last-in-first-out* priority), the *breadth-first search* tree (*first-in-first-out* priority), the *Prim tree* (*least-cost* priority), and the *Dijkstra tree* (*closest-to-root* priority). (See §10.1.)

References

[Be85] C. Berge, *Graphs*, North-Holland, 1985.

[Bo98] B. Bollobás, *Modern Graph Theory*, Springer, 1998.

[ChLe96] G. Chartrand and L. Lesniak, *Graphs and Digraphs*, Third Edition, Chapman and Hall, 1996. (First Edition, PWS Publishers, 1979.)

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Ha94] F. Harary, *Graph Theory*, Perseus reprint, 1994. (First Edition, Addison-Wesley, 1969.)

[ThSw92] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, John Wiley & Sons, 1992.

[Tu00] W. T. Tutte, *Graph Theory*, Cambridge, 2000.

[We01] D. B. West, *Introduction to Graph Theory*, Second Edition, Prentice-Hall, 2001. (First Edition, 1996.)

1.2 FAMILIES OF GRAPHS AND DIGRAPHS

Lowell W. Beineke, Purdue University at Fort Wayne

1.2.1 Building Blocks
1.2.2 Symmetry
1.2.3 Integer-Valued Invariants
1.2.4 Criterion Qualification
References

Introduction

Whenever a property of graphs is defined, a family of graphs — those with that property — results. Consequently, we focus on basic families. Along with the definitions of families, we include characterizations where appropriate. [ReWi98] offers a detailed catalog of the members of various graph and digraph families.

1.2.1 Building Blocks

Some simple graphs have as few edges or as many as possible for a given number of vertices. Some multigraphs and general graphs have as few vertices as possible for a given number of edges.

DEFINITIONS

D1: A simple graph is a ***complete graph*** if every pair of vertices is joined by an edge. The complete graph with n vertices is denoted K_n.

D2: The ***empty graph*** $\overline{K_n}$ is defined to be the graph with n vertices and no edges.

D3: The ***null graph*** K_0 is the graph with no vertices or edges.

D4: The ***trivial graph*** K_1 is the graph with one vertex and no edges.

D5: The ***bouquet*** B_n is the general graph with one vertex and n self-loops.

D6: The ***dipole*** D_n is the multigraph with two vertices and n edges.

D7: A simple digraph is a ***complete digraph*** if between every pair of vertices there is an arc in each direction. The complete digraph with n vertices is denoted $\overleftrightarrow{K}_n$.

D8: The ***path graph*** P_n is the n-vertex graph with $n-1$ edges, all on a single open path. (Quite commonly elsewhere, the subscript of the notation P_n denotes the number of edges.)

D9: The ***cycle graph*** C_n is the n-vertex graph with n edges, all on a single cycle.

REMARKS

R1: Although the empty graph may seem to some a “pointless” concept, it is the default initial value in computer representations of graph-valued variables.

R2: Whereas a "path" and a "cycle" are alternating sequences of vertices and edges, a "path graph" and a "cycle graph" are kinds of graphs.

EXAMPLES

E1:

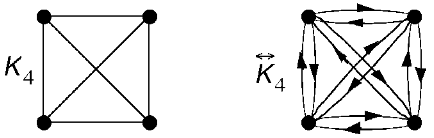


Figure 1.2.1 **A complete graph and a complete digraph.**

E2:

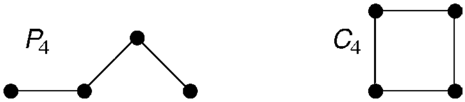


Figure 1.2.2 **A path graph and a cycle graph.**

1.2.2 Symmetry

Graphs with various kinds of symmetry are of particular interest.

Local Symmetry: Regularity

Regularity of a graph is an elementary form of local symmetry.

DEFINITIONS

D10: A graph is ***regular*** if every vertex is of the same degree.

- It is **k-*regular*** if every vertex is of degree k.

D11: A **k-*factor*** of a graph G is a k-regular spanning subgraph.

FACT

F1: All *vertex-transitive* graphs (see §1.1) are regular.

EXAMPLES

E3: For $k = 0, 1, 2, 3$, there is exactly one k-regular simple graph with 4 vertices.

E4: The only regular simple graphs with 5 vertices are the empty graph $\overline{K_5}$ (degree 0), the cycle graph C_5 (degree 2), and the complete graph K_5 (degree 4).

E5: [ReWi98] There are exactly two 3-regular simple graphs with 6 vertices.

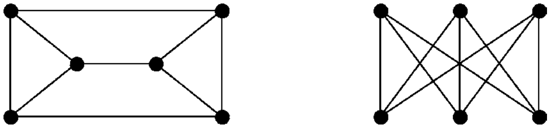


Figure 1.2.3 **The two 3-regular simple graphs with 6 vertices.**

E6: The disjoint union of the complete graphs K_3 and K_4 is a 2-regular simple 7-vertex graph that is not vertex-transitive. Its edge-complement is a 4-regular connected simple 7-vertex graph that is not vertex-transitive.

E7: Of the five 3-regular connected simple graphs with 8 vertices, two are vertex-transitive.

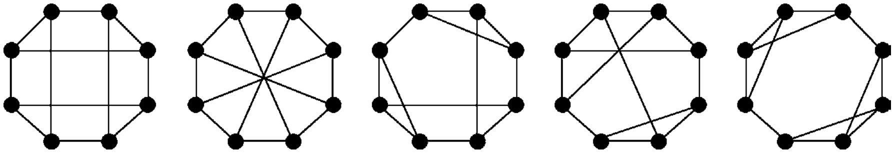


Figure 1.2.4 The five 3-regular connected simple graphs with 8 vertices.

Global Symmetry: Vertex-Transitivity

Often vertex-transitivity arises from algebra or geometry. See §6.2 for further discussion of Cayley graphs and circulant graphs.

DEFINITIONS

D12: The ***Cayley graph*** $C(\mathcal{A}, X)$ for a group $\mathcal{A}$ with generating set X has the elements of $\mathcal{A}$ as vertices and has an edge directed from a to ax for every $a \in \mathcal{A}$ and $x \in X$. We assume that vertices are labeled by elements of $\mathcal{A}$ and that edges are labeled by elements of X.

- We note that an involution x gives rise to a pair of oppositely directed edges between a and ax, for each $a \in \mathcal{A}$; sometimes we identify each such pair of directed edges to a single undirected edge labeled x.

D13: A ***circulant graph*** $\mathrm{Circ}(n;\, X)$ is defined for a positive integer n and a subset X of the integers $1, 2, \ldots, \lfloor \frac{n}{2} \rfloor$, called the ***connections***.

- The vertex set is $\mathbb{Z}_n$, the integers modulo n.
- There is an edge joining two vertices j and k if and only if the difference $|j - k|$ is in the set X. A circulant graph is a special case of a Cayley graph; an involution in the connection set gives rise to a single edge.

D14: The 1-***skeleton*** (often in graph theory, the ***skeleton***) of a k-complex K is the graph consisting of the vertices and the edges of K.

D15: The **d-*hypercube graph*** Q_d (or **d-*cube graph***) is the 1-skeleton of the d-dimensional hypercube $\{(x_1, \ldots, x_n) \mid 0 \leq x_j \leq 1\}$. This graph has 2^d vertices and is regular of degree d.

D16: The **d-*octahedral graph*** $\mathcal{O}_d$ is defined recursively:

$$\mathcal{O}_d = \begin{cases} \overline{K_2} & \text{if } n = 1 \\ \mathcal{O}_{d-1} + \overline{K_2} & \text{if } n \geq 2 \end{cases}$$

D17: The ***Petersen graph*** is the 10-vertex 3-regular graph depicted in Figure 1.2.5.

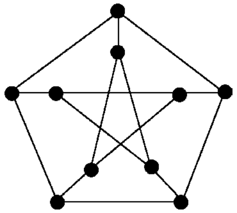


Figure 1.2.5 **The Petersen graph.**

EXAMPLES

E8: The ***n-simplex*** is the convex hull of $n+1$ affinely independent points in n-dimensional space. Its 1-skeleton is isomorphic to the complete graph K_n.

E9: A ***Platonic graph*** is the 1-skeleton of one of the five Platonic solids: the tetrahedron, the cube, the octahedron, the dodecahedron, and the icosahedron.

E10: The Petersen graph is vertex-transitive, since there is an automorphism that swaps the pentagram (i.e. the star) with the pentagon. It is not a Cayley graph of either of the two groups of order 10, i.e., of the cyclic group $\mathbb{Z}_{10}$ or of the dihedral group $\mathbb{D}_5$, and thus, not a Cayley graph.

E11: The octahedral graph $\mathcal{O}_d$ is isomorphic to $\overline{dK_2}$.

FACTS

F2: [Hypercube Characterization Theorem] The graph whose vertices are the binary sequences of length d in which two vertices are adjacent if their sequences differ in exactly one place is isomorphic to Q_d.

F3: We can construct the d-dimensional hypercube Q_d recursively, using the cartesian product operation:

$$Q_d = \begin{cases} K_1 & \text{if } d = 0 \\ Q_{d-1} \times K_2 & \text{if } d \geq 1 \end{cases}$$

1.2.3 Integer-Valued Invariants

Some of the most useful graph properties are provided by integer-valued invariants of isomorphism type. Such invariants partition all graphs into an infinite list of subclasses. Often the subclasses with low invariant values are of special interest.

Cycle Rank

The connected graphs of *cycle rank* 0 are of great special interest, since they are the *trees* (see §1.1).

DEFINITION

D18: The ***cycle rank*** of a connected graph $G = (V, E)$ is the number $|E| - |V| + 1$. (See §6.4 for an interpretation of cycle rank as the rank of a vector space.) More generally, for a graph G with $c(G)$ components, the cycle rank is the number $|E(G)| - |V(G)| + c(G)$.

EXAMPLE

E12: The connected graphs of cycle rank 0 are the trees.

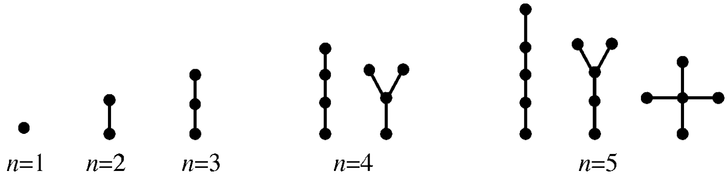


Figure 1.2.6 **The trees with up to five vertices.**

FACTS

F4: [Tree Characterization Theorem] The following statements are equivalent for a graph T with n vertices (e.g., see [GrYe99, Theorem 3.1.11]):

- T is a tree (that is, G is connected and has no cycles).
- T is connected and has $n - 1$ edges.
- T has no cycles and has $n - 1$ edges.
- Any two vertices of T are connected by exactly one path.

F5: [Inductive (Recursive) Definition of Trees] Let $\mathcal{T}$ be the family of graphs defined as follows:

(i) $K_1 \in \mathcal{T}$.

(ii) If $T \in \mathcal{T}$ and T' can be obtained by adding a new vertex and joining it to a vertex of T, then $T' \in \mathcal{T}$.

Then $\mathcal{T}$ is the family of all trees.
(Several more classes of recursively defined graphs are presented in §2.4.)

F6: The cycle rank of a graph is the sum of the cycle ranks of its components.

F7: A forest is a graph such that every component is a tree.

Chromatic Number and k-Partite Graphs

In a proper coloring of a graph, no two vertices with the same color are adjacent, and thus, every edges joins vertices in different color classes. The graphs with a proper 2-coloring are of special interest. Graph coloring is covered extensively in §5.1 and §5.2.

DEFINITIONS

D19: A simple graph or multigraph is ***bipartite*** if its vertices can be partitioned into two sets (called ***partite sets***) in such a way, that no edge joins two vertices in the same set. (For technical reasons, this includes the graph K_1 in this definition.) If r and s are the orders of the partite sets, then the graph is said to be an ***r-by-s bipartite graph***.

D20: A ***complete bipartite graph*** is a simple bipartite graph in which each vertex in one partite set is adjacent to all the vertices in the other partite set. If the two partite sets have cardinalities r and s, then this graph is denoted $K_{r,s}$.

D21: A graph is ***k-partite*** if its vertices can be partitioned into k sets (called ***partite sets***) in such a way that no edge joins two vertices in the same set.

D22: A ***complete k-partite graph*** is a simple k-partite graph in which two vertices are adjacent if and only if they are in different partite sets. All such graphs are called ***complete multipartite graphs***. If the k partite sets have orders $n_1, n_2, \ldots, n_k$, then the graph is denoted $K_{n_1,n_2,\ldots,n_k}$, and if each partite set has order r, then $K_{k(r)}$.

EXAMPLES

E13: Every tree is bipartite.

E14: Every cycle with an even number of vertices is bipartite, and no cycle with an odd number is bipartite.

E15: The complete d-partite graph $K_{d(2)}$ is isomorphic to the d-octahedral graph $\mathcal{O}_d$.

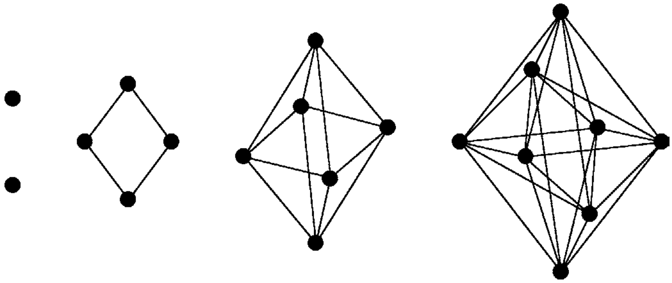


Figure 1.2.7 **The complete d-partite graphs $K_{d(2)}$, for $d = 1, \ldots, 4$.**

FACTS

F8: [Bipartite Graph Characterization Theorem] A graph is bipartite if and only if the length of each of its cycles is even (e.g., see [GrYe99, Theorem 1.5.3]).

F9: A graph is k-colorable if and only if it is k-partite.

F10: For $k \geq 3$, the problem of deciding whether a graph is k-partite is NP-complete.

k-Connectivity and k-Edge-Connectivity

Graphs can be categorized according to their *connectivity* and their *edge-connectivity.* There are analogues for strong connectedness in digraphs. See §4.1 and §4.7 for extensive coverage of connectivity.

DEFINITIONS

D23: The ***(vertex-)connectivity*** of a graph G, denoted $\kappa_v(G)$, is the minimum number of vertices whose removal from G leaves a non-connected or trivial graph.

D24: The ***edge-connectivity*** of a nontrivial graph G, denoted $\kappa_e(G)$ is the minimum number of edges whose removal from G results on a non-connected graph.

NOTATION: The subscripted "G" is often suppressed when the graph G is understood. Elsewhere, the notation κ and λ are used instead of κ_v and κ_e, respectively.

D25: A graph G with connectivity $\kappa_v \geq k \geq 1$ is called ***k*-connected**. Equivalently, G is k-connected if the removal of $k-1$ or fewer vertices leaves neither a non-connected graph nor a trivial one.

D26: A graph G with edge-connectivity $\kappa_e \geq k \geq 1$ is called ***k-edge-connected***. That is, the removal of $k-1$ or fewer edges from a k-edge-connected graph results in a connected graph.

D27: A digraph is ***strongly k-connected*** (or ***k-strong***) if the result of removing any set of fewer than k vertices is strongly connected and nontrivial.

D28: A digraph is ***strongly k-arc-connected*** (or ***k-arc-strong***) if the result of removing any set of fewer than k arcs is strongly connected and nontrivial.

Minimum Genus

Graphs can be categorized according to their topological properties.

DEFINITIONS

D29: The ***minimum genus*** (or simply the ***genus***) of a connected graph G is the smallest number g such that G can be drawn on the orientable surface S_g (see §7.1) without any edge-crossings.

D30: A graph of genus 0 is ***planar***.

1.2.4 Criterion Qualification

A graph family is also specified as the set of all graphs or digraphs that match a stated criterion, e.g., traversability and various forms of minimality and maximality.

DEFINITIONS

D31: A graph is ***eulerian*** if it has a closed walk that contains every edge exactly once. (See §1.3 for the history of eulerian graphs and §4.2 for an extensive discussion.)

D32: A graph is ***hamiltonian*** if it has a spanning cycle. (See §1.3 for the history of hamiltonian graphs and §4.5 for an extensive discussion.)

D33: A k-chromatic graph is ***critically k-chromatic*** if its chromatic number would decrease if any edge were removed. (See §5.1.)

D34: A k-connected graph is ***critically k-connected*** if its connectivity would decrease if any vertex were removed. (See §4.1.)

D35: A k-edge-connected graph is ***critically k-edge-connected*** if its edge-connectivity would decrease if any edge were removed. (See §4.1.)

D36: The ***line graph*** $L(G)$ of a graph G has the edges of G as its vertices; two vertices of $L(G)$ are adjacent if the edges in G to which they correspond have a common vertex. Also, a graph H is said to be a ***line graph*** if there exists a graph G such that H is isomorphic to $L(G)$.

D37: A ***tournament*** is a digraph in which there is exactly one arc between each pair of vertices. (See §3.3.)

EXAMPLE

E16:

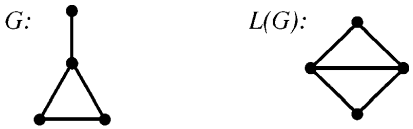


Figure 1.2.8 **A graph and its line graph.**

FACTS

F11: [Line Graph Characterization] The following statements are equivalent:

- G is a line graph.
- [Kr43] The edges of G can be partitioned into complete subgraphs in such a way that no vertex is in more than two.
- [Be70] None of the nine graphs in Figure 1.2.9 is an induced subgraph of G.

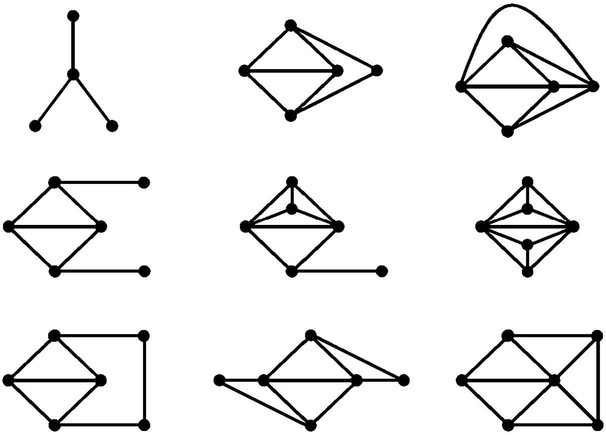


Figure 1.2.9 **The nine forbidden induced subgraphs.**

F12: A strongly connected tournament contains a directed spanning cycle.

EXAMPLE

E17:

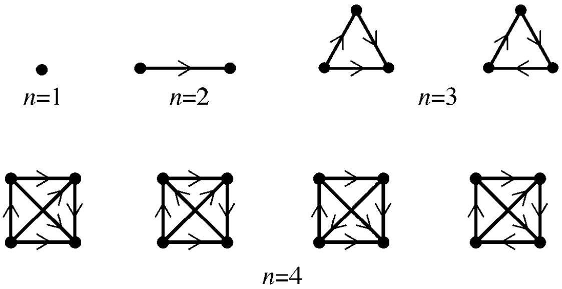


Figure 1.2.10 **All tournaments with one to four vertices.**

References

[Be70] L. W. Beineke, Characterizations of derived graphs, *J. Combin. Theory* 9 (1970), 129–135.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Kr43] J. Krausz, Dèmonstation nouvelle d'une théorème de Whitney sur les réseaux, *Mat. Fiz. Lapok* 50 (1943), 75–89.

[ReWi98] R. C. Read and R. J. Wilson, *An Atlas of Graphs*, Oxford University Press, 1998.

1.3 HISTORY OF GRAPH THEORY

Robin J. Wilson, The Open University, UK

1.3.1 Traversability
1.3.2 Trees
1.3.3 Topological Graphs
1.3.4 Graph Colorings
1.3.5 Graph Algorithms
References

Introduction

Although the first mention of a graph was not until 1878, graph-theoretical ideas can be traced back to 1735 when Leonhard Euler (1707-83) presented his solution of the Königsberg bridges problem. This chapter summarizes some important strands in the development of graph theory since that time. Further information can be found in [BiLlWi98] or [Wi99].

1.3.1 Traversability

The origins of graph theory can be traced back to Euler's work on the Königsberg bridges problem (1735), which subsequently led to the concept of an *eulerian graph.* The study of cycles on polyhedra by the Revd. Thomas Penyngton Kirkman (1806-95) and Sir William Rowan Hamilton (1805-65) led to the concept of a *hamiltonian graph.*

The Königsberg Bridges Problem

The *Königsberg bridges problem*, pictured in Figure 1.3.1, asks whether there is a continuous walk that crosses each of the seven bridges of Königsberg exactly once — and if so, whether a closed walk can be found. See §4.2 for more extensive discussion of issues concerning *eulerian graphs.*

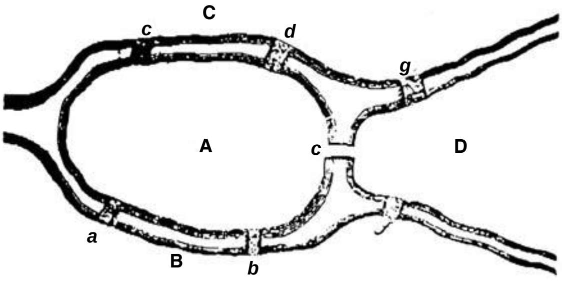


Figure 1.3.1 The seven bridges of Königsberg.

FACTS [BiLlWi98, Chapter 1]

F1: On 26 August 1735 Leonhard Euler presented a paper on "The solution of a problem relating to the geometry of position" to the Academy of Sciences of St. Petersburg, Russia, proving that there is no such continuous walk across the seven bridges.

F2: In 1736, Euler communicated his solution to several other mathematicians, outlining his views on the nature of the problem and on its situation in the geometry of position.

F3: Euler [Eu:1736] sent his solution of the problem to the *Commentarii Academii Scientiarum Imperialis Petropolitanae* under the title "Solutio problematis ad geometriam ad geometriam situs pertinentis". Although dated 1736, it did not appear until 1741, and it was later republished in the new edition of the *Commentarii* in 1752.

F4: Euler's paper is divided into 21 sections, of which nine are on the Königsberg bridges problem, and the remainder are concerned with general arrangements of bridges and land areas.

F5: Euler did not draw a graph in order to solve the problem, but he reformulated the problem as one of trying to find a sequence of eight letters A, B, C, or D (the land areas) such that the pairs AB and AC are adjacent twice (corresponding to the two bridges between A and B and between A and C), and the pairs AD, BD, and CD are adjacent just once. He showed by a counting argument that no such sequence exists, thereby proving that the Königsberg bridges problem has no solution.

F6: In discussing the general problem, Euler observed that the number of bridges written next to the letters A, B, C, etc. together add up to twice the number of bridges. This is the first appearance of what some graph-theorists now call the "handshaking lemma", that the sum of the vertex-degrees in a graph is equal to twice the number of edges.

F7: Euler's main conclusions for the general situation were as follows:

- If there are more than two areas to which an odd number of bridges lead, then such a journey is impossible.
- If the number of bridges is odd for exactly two areas, then the journey is possible if it starts in either of these two areas.
- If, finally, there are no areas to which an odd number of bridges lead, then the required journey can be accomplished starting from any area.

These results correspond to the conditions under which a graph has an eulerian, or semi-eulerian, trail.

F8: Euler noted the converse result, that if the above conditions are satisfied, then a route is possible, and gave a heuristic reason why this should be so, but did not prove it. A valid demonstration did not appear until a related result was proved by C. Hierholzer [Hi:1873] in 1873.

Diagram-Tracing Puzzles

A related area of study was that of *diagram-tracing puzzles*, where one is required to draw a given diagram with the fewest possible number of connected strokes. Such puzzles can be traced back many hundreds of years – for example, there are some early African examples.

FACTS [BiLlWi98, Chapter 1]

F9: In 1809 L. Poinsot [Po:1809] wrote a memoir on polygons and polyhedra in which he posed the following problem:

> Given some points situated at random in space, it is required to arrange a single flexible thread uniting them two by two in all possible ways, so that the two ends of the thread join up and the total length is equal to the sum of all the mutual distances.

Poinsot noted that a solution is possible only when the number of points is odd, and gave a method for finding such an arrangement for each possible value. In modern terminology, the question is concerned with eulerian trails in complete graphs of odd order.

F10: Other diagram-tracing puzzles were posed and solved by T. Clausen [Cl:1844] and J. B. Listing [Li:1847]. The latter appeared in the book *Vorstudien zur Topologie*, the first place that the word "topology" appeared in print.

F11: In 1849, O. Terquem asked for the number of ways of laying out a complete ring of dominoes. This is essentially the problem of determining the number of eulerian tours in the complete graph K_7, and was solved by M. Reiss [Re:1871-3] and later by G. Tarry.

F12: The connection between the Königsberg bridges problem and diagram-tracing puzzles was not recognized until the end of the 19th century. It was pointed out by W. W. Rouse Ball [Ro:1892] in *Mathematical Recreations and Problems*. Rouse Ball seems to have been the first to use the graph in Figure 1.3.2 to solve the problem.

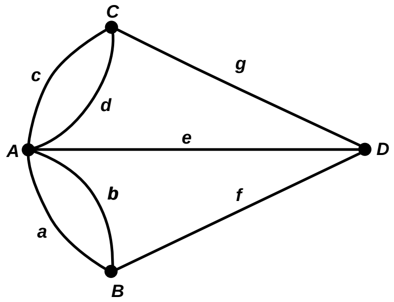


Figure 1.3.2 **The graph of the Königsberg bridges problem.**

Hamiltonian Graphs

A type of graph problem that superficially resembles the eulerian problem is that of finding a cycle that passes just once through each vertex of a given graph. Because of Hamilton's influence, such graphs are now called *hamiltonian graphs* (see §4.5), instead of more justly being named after Kirkman, who considered the more general problem prior to Hamilton's consideration of the dodecahedron, as discussed below.

FACTS [BiLlWi98, Chapter 2]

F13: An early example of such a problem is the *knight's tour problem*, of finding a succession of knight's moves on a chessboard, visiting each of the 64 squares just once and returning to the starting point. This problem can be dated back many hundreds of years, and systematic solutions were given by Euler [Eu:1759], A.-T. Vandermonde [Va:1771] and others.

F14: In 1855 Kirkman [Ki:1855] wrote a paper investigating those polyhedra for which one can find a cycle passing through all the vertices just once. He proved that every polyhedron with even-sided faces and an odd number of vertices has no such cycle, and gave as an example the polyhedron obtained by "cutting in two the cell of a bee" (see Figure 1.3.3).

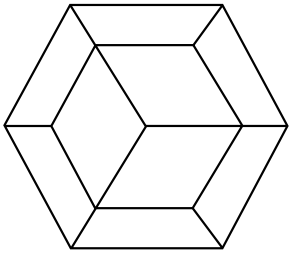


Figure 1.3.3 Kirkman's "cell of a bee" example.

F15: Arising from his work on non-commutative algebra, Hamilton considered cycles passing through all the vertices of a dodecahedron. He subsequently invented a game, called the icosian game (see Figure 1.3.4), in which the player was challenged to find such cycles on a solid dodecahedron, satisfying certain extra conditions.

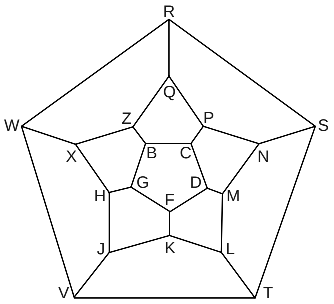


Figure 1.3.4 Hamilton's icosian game.

F16: In 1884, P. G. Tait asserted that every 3-valent polyhedron has a hamiltonian cycle. This assertion was subsequently proven false by W. T. Tutte [Tu46] in 1946 (see Figure 1.3.5).

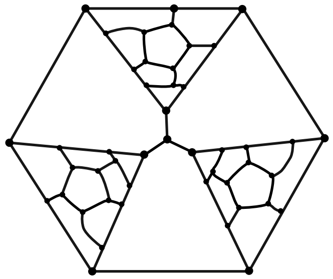


Figure 1.3.5 Tutte's 3-valent non-hamiltonian polyhedron.

F17: Sufficient conditions for a graph to be hamiltonian were later obtained by G. A. Dirac [Di52], O. Ore [Or60], J. A. Bondy and V. Chvátal in 1976, and others.

F18: Hamiltonian digraphs have also been investigated, by A. Ghouila-Houri (1960), H. Meyniel (1973), and others.

1.3.2 Trees

The concept of a tree, a connected graph without cycles, appeared implicitly in the work of Gustav Kirchhoff (1824-87), who employed graph-theoretical ideas in the calculation of currents in electrical networks. Later, trees were used by Arthur Cayley (1821-95), James Joseph Sylvester (1806-97), Georg Pólya (1887-1985), and others, in the solution of problems involving the enumeration of certain chemical molecules.

Counting Trees

Enumeration techniques involving trees first arose in connection with a problem in the differential calculus, but they soon came to be fundamental tools in the counting of chemical molecules, as well as providing a fascinating topic of interest in their own right. Enumeration of various kinds of graphs is discussed in §6.3.

FACTS [BiLlWi98, Chapter 3] [PoRe87]

F19: While working on a problem inspired by some work of Sylvester on "differential transformation and the reversion of serieses", Cayley [Ca:1857] was led to the enumeration of rooted trees.

F20: Cayley's method was to take a rooted tree and remove its root, thereby obtaining a number of smaller rooted trees (see Figure 1.3.6).

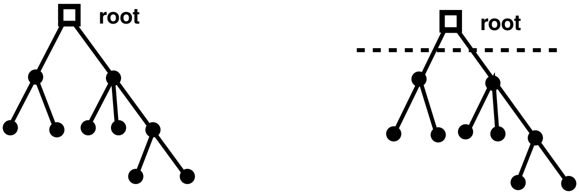


Figure 1.3.6 **Splitting a rooted tree.**

Letting A_n be the number of rooted trees with n branches, Cayley proved that the generating function

$$1 + A_1x + A_2x^2 + A_3x^3 + \ldots$$

is equal to the product

$$(1-x)^{-1} \cdot (1-x^2)^{-A_1} \cdot (1-x^3)^{-A_2} \cdot \ldots$$

Using this equality, he was able to calculate the numbers A_n one at a time.

F21: Around 1870, Sylvester and C. Jordan independently defined the *center/bicenter* and the *centroid/bicentroid* of a tree.

F22: In 1874, Cayley [Ca:1874] found a method for solving the more difficult problem of counting unrooted trees. This method, which he applied to chemical molecules, consisted essentially of starting at the center or centroid of the tree or molecule and working outwards.

F23: In 1889, Cayley [Ca:1889] presented his n^{n-2} formula for the number of labeled trees with n vertices. He explained why the formula holds when $n = 6$, but he did not give a proof in general. The first correct proof was given by H. Prüfer [Pr18]: his method was to establish a one-to-one correspondence between such labeled trees and sequences of length $n - 2$ formed from the numbers $1, 2, \ldots, n$.

F24: In a fundamental paper of 1937, Pólya [Pó37] combined the classical idea of a generating function with that of a permutation to obtain a powerful theorem that enabled him to enumerate certain types of configuration under the action of a group of symmetries. Some of Pólya's work was anticipated by J. H. Redfield [Re27], but Redfield's paper was obscure and had no influence on the development of the subject.

F25: Later results on the enumeration of trees were derived by R. Otter [Ot48] and others. The field of graphical enumeration (see [HaPa73]) was subsequently further developed by F. Harary [Ha55], R. C. Read [Re63], and others.

Chemical Trees

By 1850 it was already known that chemical elements combine in fixed proportions. Chemical formulas such as CH_4 (methane) and C_2H_5OH (ethanol) were known, but it was not understood how the elements combine to form such substances. Around this time, chemical ideas of valency began to be established, particularly when Alexander Crum Brown presented his graphic formulae for representing molecules. Figure 1.3.7 presents his representation of ethanol, the usual drawing, and the corresponding tree graph.

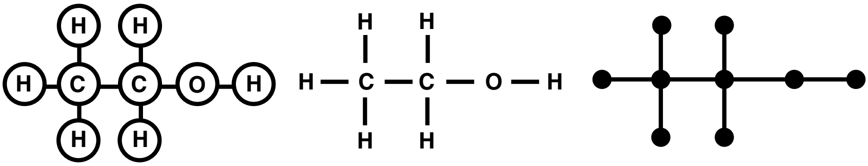


Figure 1.3.7 Representations of ethanol.

FACTS [BiLlWi98, Chapter 4]

F26: Crum Brown's graphic notation explained for the first time the phenomenon of isomerism, whereby there exist pairs of molecules (isomers) with the same chemical formula but different chemical properties. Figure 1.3.8 shows isomers with chemical formula C_4H_{10}.

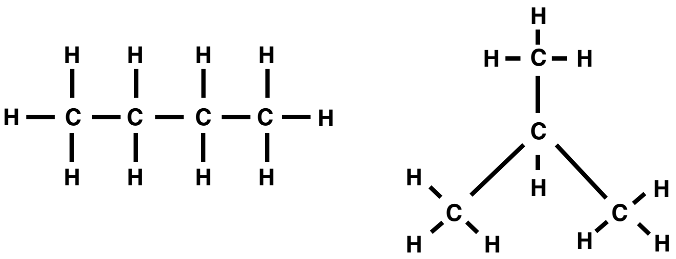


Figure 1.3.8 Two isomers: butane and isobutane.

F27: Cayley [Ca:1874] used tree-counting methods to enumerate paraffins (alkanes) with up to 11 carbon atoms, as well as various other families of molecules; the followiing table gives the number of isomers of alkanes for $n = 1, \ldots, 8$.

Formula	CH_4	C_2H_6	C_3H_8	C_4H_{10}	C_5H_{12}	C_6H_{14}	C_7H_{16}	C_8H_{18}
Number	1	1	1	2	3	5	9	18

F28: W. K. Clifford and Sylvester believed that a connection could be made between chemical atoms and binary quantics in invariant theory, a topic to which Cayley and Sylvester had made significant contributions. In 1878, Sylvester [Sy:1877-8] wrote a short note in *Nature* about this supposed connection, remarking that:

> Every invariant and covariant thus becomes expressible by a *graph* precisely identical with a Kekuléan diagram or chemicograph.

This was the first appearance of the word *graph* in the graph-theoretic sense.

F29: In 1878, Sylvester [Sy:1878] wrote a lengthy article on the graphic approach to chemical molecules and invariant theory in the first volume of the *American Journal of Mathematics*, which he had recently founded.

F30: Little progress was made on the enumeration of isomers until the 1920s and 1930s. A. C. Lunn and J. K. Senior [LuSe29] recognized the importance of permutation groups for this area, and Pólya's above-mentioned paper solved the counting problem for several families of molecules.

1.3.3 Topological Graphs

Euler's polyhedron formula [Eu:1750] was the foundation for topological graph theory, since it holds also for planar graphs. It was later extended to surfaces other than the sphere. In 1930, a fundamental characterization of graphs imbeddable in the sphere was given by Kazimierz Kuratowski (1896-1980), and recent work – notably by Neil Robertson, Paul Seymour, and others – has extended these results to the higher order surfaces.

Euler's Polyhedron Formula

The Greeks were familiar with the five regular solids, but there is no evidence that they knew the simple connection between the numbers V of vertices, E of edges, and F of faces of a polyhedron:

$$V - E + F = 2$$

In the 17th century, René Descartes studied polyhedra, and he obtained results from which Euler's formula could later be derived. However, since Descartes had no concept of an edge, he was unable to make this deduction.

FACTS [BiLlWi98, Chapter 5] [Cr99]

F31: The first appearance of the polyhedron formula appeared in a letter, dated 14 November 1750, from Euler to C. Goldbach. Denoting the number of faces, solid angles (vertices) and joints (edges) by $\mathcal{H}$, $\mathcal{S}$, and $\mathcal{A}$,

- In every solid enclosed by plane faces the aggregate of the number of faces and the number of solid angles exceeds by two the number of edges, or $\mathcal{H} + \mathcal{S} = \mathcal{A} + 2$.

F32: Euler was unable to prove his formula. In 1752 he attempted a proof by dissection, but it was deficient. The first valid proof was given by A.-M. Legendre [Le:1794] in 1794, using metrical properties of spherical polygons.

F33: In 1813, A.-L. Cauchy [Ca:1813] obtained a proof of Euler's formula by stereographically projecting the polyhedron onto a plane and considering a triangulation of the resulting planar graph.

F34: Around the same time, S.-A.-J. Lhuilier [Lh:1811] gave a topological proof that there are only five regular convex polyhedra, and he anticipated the idea of duality by noting that four of them occur in reciprocal pairs. He also found three types of polyhedron for which Euler's formula fails – those with indentations in their faces, those with an interior cavity, and ring-shaped polyhedra drawn on a torus (that is, polyhedra with a 'tunnel' through them). For such ring-shaped polyhedra, Lhuilier derived the formula

$$V - E + F = 0$$

and extended his discussion to prove that, if g is the number of tunnels in a surface on which a polyhedral map is drawn, then

$$V - E + F = 2 - 2g$$

The number g is now called the *genus of the surface*, and the quantity $2 - 2g$ is called the *Euler characteristic*. (See §7.1.)

F35: In 1861-2, Listing [Li:1861-2] wrote *Der Census räumliche Complexe*, an extensive investigation into complexes, and studied how their topological properties affect the generalization above of Euler's formula. This work proved to be influential in the subsequent development of topology. In particular, H. Poincaré took up Listing's ideas in his papers of 1895-1904 that laid the foundations for algebraic topology.

F36: The work of Poincaré's work was instantly successful, and it appeared in an article by M. Dehn and P. Heegaard [DeHe07] on analysis situs (topology) in the ten-volume *Encyklopädie der Mathematischen Wissenschaften*. His ideas were further developed by O. Veblen [Ve22] in a series of colloquium lectures on analysis situs for the American Mathematical Society in 1916.

Planar Graphs

The study of planar graphs originated in two recreational problems involving the complete graph K_5 and the complete bipartite graph $K_{3,3}$. These graphs (shown in Figure 1.3.9) proved to be the main obstacles to planarity, as was subsequently demonstrated by Kuratowski.

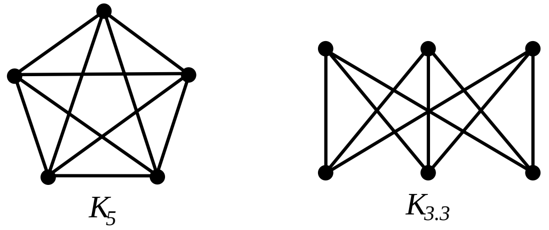


Figure 1.3.9 **The Kuratowski graphs K_5 and $K_{3,3}$**

FACTS [BiLlWi98, Chapter 8]

F37: Around the year 1840, A. F. Möbius presented the following puzzle to his students:

> There was once a king with five sons. In his will he stated that after his death the sons should divide the kingdom into five regions so that the boundary of each region should have a frontier line in common with each of the other four regions. Can the terms of the will be satisfied?

This question asks whether one can draw five mutually neighboring regions in the plane. The connection with graph theory can be seen from its dual version, later formulated by H. Tietze:

> The king further stated that the sons should join the five capital cities of his kingdom by roads so that no two roads intersect. Can this be done?

In this dual formulation, the problem is that of deciding whether the graph K_5 is planar.

F38: An old problem, whose origins are obscure, is the *utilities problem*, or *gas-water-electricity problem*, mentioned by H. Dudeney [Du13] in the *Strand Magazine* of 1913:

> The puzzle is to lay on water, gas, and electricity, from W, G, and E, to each of the three houses, A, B, and C, without any pipe crossing another (see Figure 1.3.10).

This problem is that of deciding whether $K_{3,3}$ is planar.

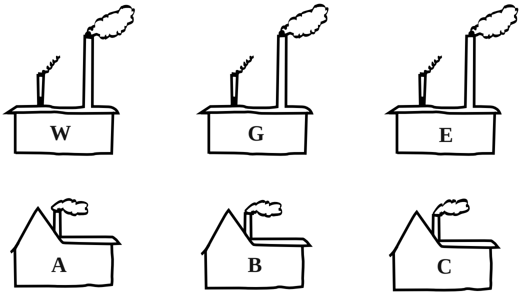


Figure 1.3.10 **The gas-water-electricity problem**

F39: In 1930 Kuratowski [Ku30] published a paper proving that every planar graph has a subgraph homeomorphic to K_5 or $K_{3,3}$; this result was obtained independently by O. Frink and P. A. Smith.

F40: In 1931 H. Whitney [Wh31] discovered an abstract definition of duality that is purely combinatorial and agrees with the geometrical definition of duality for planar graphs. He proved that, with this general definition of duality, a graph is planar if and only if it has an abstract dual. Related results were obtained by S. MacLane and others.

F41: In 1935 Whitney [Wh35] generalized the idea of independence in graphs and vector spaces to the concept of a matroid. The dual of a matroid extends and clarifies the duality of planar graphs, and Tutte [Tu59] used these ideas in the late-1950s to obtain a Kuratowski-type criterion for a matroid to arise from a graph (see §6.6).

Graphs on Higher Surfaces

A graph drawn without crossings on a plane corresponds (by stereographic projection) to a graph similarly drawn on the surface of a sphere. This leads to the idea of graphs drawn on surfaces other than the sphere. The initial work in this area was carried out, in the context of coloring maps, by Percy Heawood (1861-1955) and Lothar Heffter (1862-1962) for orientable surfaces, and by Heinrich Tietze (1880-1964) for non-orientable surfaces, but the basic problems in the area were not solved until Gerhard Ringel and Ted Youngs solved the Heawood conjecture in the 1960s and Neil Robertson and Paul Seymour generalized Kuratowski's theorem to other surfaces in the 1980s.

FACTS [BiLlWi98, Chapter 7; Ri74]

F42: In 1890, Heawood [He:1890] presented an imbedding of the complete graph K_7 on a torus. He also derived a formula for the genus of a surface on which a given complete graph can be imbedded, but his attempted proof of this formula was deficient.

F43: In 1891, L. Heffter [He:1891] investigated the imbedding of complete graphs on orientable surfaces other than the sphere and the torus, and he proved that Heawood's formula is correct for orientable surfaces of low genus and certain other surfaces.

F44: In 1910, H. Tietze [Ti10] extended Heffter's considerations to certain non-orientable surfaces, such as the Möbius band and the projective plane, and stated a corresponding Heawood formula. He was unable to prove it for the Klein bottle, but this case was settled by P. Franklin [Fr34] in 1934, who found that it was an exception to the formula. In 1935, I. Kagno [Ka35] proved the formula for surfaces of non-orientable genus 3, 4, and 6.

F45: The Heawood formula for general non-orientable surfaces was proved in 1952 by Ringel. The proof for orientable surfaces proved to be much more difficult, involving 300 pages of consideration of 12 separate cases. Most of these were settled in the mid-1960s, and the proof was completed in 1968 by Ringel and Youngs [RiYo68], using W. Gustin's [Gu63] combinatorial inspiration in 1963 of a *current graph*. Since then, the transformation by J. L. Gross [Gr74] of numerous types of specialized combinatorial current graphs into a unified topological object, with its dualization to a *voltage graph* (see §7.4), has led to simpler solutions (see Gross and T. W. Tucker [GrTu74]).

F46: In a sequence of papers in the 1980s of great mathematical depth, Robertson and Seymour [RoSe85] proved that, for each orientable genus g, the set of "forbidden subgraphs" is finite (see §7.7). However, apart from the sphere, the number of forbidden subgraphs runs into hundreds, even for the torus. For non-orientable surfaces, there is a similar result, and in 1979 H. H. Glover, J. P. Huneke, and C. S. Wang [GlHuWa79] obtained a set of 103 forbidden subgraphs for the projective plane.

1.3.4 Graph Colorings

Early work on colorings concerned the coloring of the countries of a map and, in particular, the celebrated four-color problem. This was first posed by Francis Guthrie in 1852, and a celebrated (incorrect) "proof" by Alfred Bray Kempe appeared in 1879. It was eventually proved by Kenneth Appel and Wolfgang Haken in 1976, building on the earlier work of Kempe, George Birkhoff, Heinrich Heesch, and others, and a simpler proof was subsequently produced by Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas [1994]. Meanwhile, attention had turned to the dual problem of coloring the vertices of a planar graph and of graphs in general. There was also a parallel development in the coloring of the edges of a graph, starting with a result of Tait [1880], and leading to a fundamental theorem of V. G. Vizing in 1964. As mentioned earlier, the corresponding problem of coloring maps on other surfaces was settled by Ringel and Youngs in 1968.

The Four-Color Problem

Many developments in graph theory can be traced back to attempts to solve the celebrated four-color problem on the coloring of maps.

FACTS [BiLlWi98, Chapter 6] [Wi02]

F47: The earliest known mention of the four-color problem occurs in a letter from A. De Morgan to Hamilton, dated 23 October 1852. De Morgan described how a student had asked him whether every map can be colored with just four colors in such a way that neighbouring countries are colored differently. The student later identified himself as Frederick Guthrie, giving credit for the problem to his brother Francis, who formulated it while coloring the counties of a map of England. Hamilton was not interested in the problem.

F48: De Morgan wrote to various friends, outlining the problem and trying to describe where the difficulty lies. On 10 April 1860, the problem first appeared in print, in an unsigned book review in the *Athenaeum*, written by De Morgan. This review was read in the U.S. by C. S. Peirce, who developed a life-long interest in the problem.

F49: On 13 June 1878, at a meeting of the London Mathematical Society, Cayley asked whether the problem had been solved. Shortly after, he published a short note describing where the difficulty might lie, and he showed that it is sufficient to restrict one's attention to trivalent maps.

F50: In 1879, Kempe [Ke:1879], a former Cambridge student of Cayley, published a purported proof of the four-color theorem in the *American Journal of Mathematics*, which had recently been founded by Sylvester. Kempe showed that every map must contain a country with at most five neighbours, and he showed how any coloring of the rest of the map can be extended to include such a country. His solution included a new technique, now known as a *Kempe-chain* argument, in which the colors in a two-colored section of the map are interchanged. Kempe's proof for a map that contains a digon, triangle, or quadrilateral was correct, but his argument for the pentagon (where he used two simultaneous color-interchanges) was fallacious.

F51: In 1880, Tait [Te:1878-80] presented "improved proofs" of the four-color theorem, all of them fallacious. Other people interested in the four-color problem at this time were C. L. Dodgson (Lewis Carroll), F. Temple (Bishop of London), and the Victorian educator J. M. Wilson.

F52: In 1890, Heawood [He:1890] published a paper in the *Quarterly Journal of Pure and Applied Mathematics*, pointing out the error in Kempe's proof, salvaging enough to deduce the five-color theorem, and generalizing the problem in various ways, such as for other surfaces (see §1.1.3). Heawood subsequently published another six papers on the problem, the last while he was in his 90th year. Kempe admitted his error, but he was unable to put it right.

F53: During the first half of the 20th century two ideas emerged, each of which finds its origin in Kempe's paper. The first is that of an *unavoidable set* — a set of configurations, at least one of which must appear in any map. Unavoidable sets were produced by P. Wernicke [We:1904] (see Figure 1.3.11), by P. Franklin, and by H. Lebesgue.

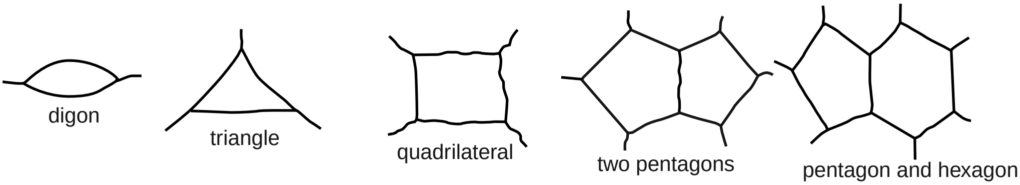


Figure 1.3.11 **Wernicke's unavoidable set**

The second is that of a *reducible configuration* — a configuration of countries with the property that any coloring of the rest of the map can be extended to the configuration: no such configuration can appear in any counter-example to the four-color theorem. Birkhoff [Bi:1913] showed that the arrangement of four pentagons in Figure 1.3.12 (known as the Birkhoff diamond) is a reducible configuration.

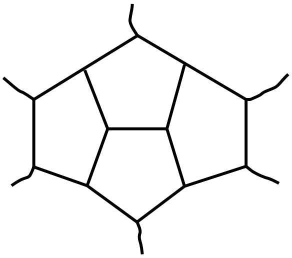


Figure 1.3.12 **The Birkhoff diamond**

F54: In 1912, Birkhoff [Bi12] investigated the number of ways of coloring a given map with k colors, and he showed that this is always a polynomial in k, now called the *chromatic polynomial* of the map.

F55: In 1922, Franklin [Fr22] presented further unavoidable sets and reducible configurations, and he deduced that the four-color theorem is true for all maps with up to 25 countries. This number was later increased several times.

F56: Around 1950 Heesch started to search for an unavoidable set of reducible configurations. Over the next few years, Heesch [He69] produced thousands of reducible configurations.

F57: In 1976, Appel and Haken [ApHa77, ApHaKo77], with the assistance of J. Koch, obtained an unavoidable set of 1482 reducible configurations, thereby proving the four-color theorem. Their solution required substantial use of a computer to test the configurations for reducibility.

F58: Around 1994, Robertson, Sanders, Seymour, and Thomas [RoSaSeTh97] produced a more systematic proof. Using a computer to assist with both the unavoidable set and the reducible configuration parts of the solution, they systematized the Appel-Haken approach, and they obtained an unavoidable set of 633 reducible configurations.

Other Graph Coloring Problems

Arising from work on the four-color problem, progress was being made on other graph problems involving the coloring of edges or vertices.

FACTS [BiLlWi98, Chapter 6] [FiWi77] [JeTo95]

F59: In his 1879 paper on the coloring of maps, Kempe [Ke:1879] outlined the dual problem of coloring the vertices of a planar graph in such a way that adjacent vertices are colored differently. This dual approach to map-coloring was later taken up by H. Whitney in a fundamental paper of 1932 and by most subsequent workers on the four-color problem.

F60: In 1880, Tait [Ta:1878-80] proved that the four-color theorem is equivalent to the statement that the edges of every trivalent map can be colored with three colors in such a way that each color appears at every vertex.

F61: In 1916, D. König [Kö16] proved that the edges of any bipartite graph with maximum degree d can be colored with d colors. (See §11.3.)

F62: The idea of coloring the vertices of a graph so that adjacent vertices are colored differently developed a life of its own in the 1930s, mainly through the work of Whitney, who wrote his Ph.D. thesis on the coloring of graphs.

F63: In 1941, L. Brooks [Br41] proved that the chromatic number of any simple graph with maximum degree d is at most $d + 1$, with equality only for odd cycles and odd complete graphs. (See §5.1.)

F64: In the 1950s, substantial progress on vertex-colorings was made by G. A. Dirac, who introduced the idea of a *critical graph*.

F65: In 1964, V. G. Vizing [Vi64] proved that the edges of any simple graph with maximum degree d can always be colored with $d + 1$ colors. In the following year, Vizing produced many further results on edge-colorings.

F66: The concepts of the chromatic number and edge-chromatic number of a graph have been generalized by a number of writers — for example, M. Behzad and others introduced total colorings in the 1960s, and P. Erdős and others introduced list colorings.

Factorization

A graph is *k-regular* if each of its vertices has degree k. Such graphs can sometimes be split into regular subgraphs, each with the same vertex-set as the original graph. A *k-factor* in a graph is a k-regular subgraph that contains all the vertices of the original graph. Fundamental work on factors in graphs was carried out by Julius Petersen [1839-1910] and W. T. Tutte [1914-2002]. (See §5.4.)

FACTS [BiLlWi98, Chapter 10]

F67: In 1891, Petersen [Pe:1891] wrote a fundamental paper on the factorization of regular graphs, arising from a problem in the theory of invariants. In this paper he proved that if k is even, then any k-regular graph can be split into 2-factors. He also proved that any 3-regular graph possesses a 1-factor, provided that it has not more than two "leaves"; a leaf is a subgraph joined to the rest of the graph by a single edge.

F68: In 1898, Petersen [Pe:1898] produced a trivalent graph with no leaves, now called the *Petersen graph* (see Figure 1.3.13), which cannot be split into three 1-factors; it can, however, be split into a 1-factor (the spokes) and a 2-factor (the pentagon and pentagram).

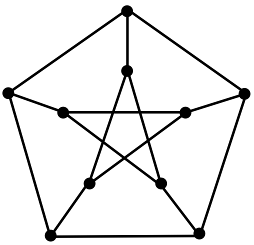


Figure 1.3.13 The Petersen graph

F69: In 1947, Tutte [Tu47] produced a characterization of graphs that contain a 1-factor. Five years later he extended his result to a characterization of graphs that contain a k-factor, for any k.

1.3.5 Graph Algorithms

Graph theory algorithms can be traced back to the 19th century, when Fleury gave a systematic method for tracing an eulerian graph and G. Tarry showed how to escape from a maze (see §4.2). The 20th century saw algorithmic solutions to such problems as the minimum connector problem, the shortest and longest path problems, and the *Chinese Postman Problem* (see §4.3), as well as to a number of problems arising in operational research. In each of these problems we are given a network, or weighted graph, to each edge (and/or vertex) of which has been assigned a number, such as its length or the time taken to traverse it.

FACTS [Da82] [LLRS85] [LoPl86]

F70: The *Traveling Salesman Problem*, in which a salesman wishes to make a cyclic tour of a number of cities in minimum time or distance, appeared in rudimentary form in 1831. It reappeared in mathematical circles in the early 1930s, at Princeton, and was later popularized at the RAND Corporation. This led to a fundamental paper of G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson [DaFuJo54] that included the solution of a traveling salesman problem with 49 cities. In the 1980s a problem with 2392 cities was settled by Padberg and Rinaldi [PaRi87]. (See §4.6.)

F71: The greedy algorithm for the *minimum connector problem*, in which one seeks a minimum-length spanning tree in a weighted graph, can be traced back to O. Boruvka [Bo26] and was later rediscovered by J. B. Kruskal [Kr56]. A related algorithm, due to V. Jarník (1931), was rediscovered by R. C. Prim (1957). (See §10.1.)

F72: Graph algorithms were developed by G. B. Dantzig and D. R. Fulkerson [DaFu54] for finding the maximum flow of a commodity between two nodes in a capacitated network, and by R. E. Gomory and T. C. Hu [GoHu61] for determining *maximum flows* in multi-terminal networks.

F73: Finding a longest path, or critical path, in an activity network dates from the 1940s and 1950s, with *PERT* (Program Evaluation and Review Technique) used by the US Navy for problems involving the building of submarines and CPM (*Critical Path Method*) developed by the Du Pont de Nemours Company to minimize the total cost of a project. (See §3.2.)

F74: There are several efficient algorithms for finding the shortest path in a given network, of which the best known is due to E. W. Dijkstra [Di59]. (See §10.1.)

F75: The Chinese postman problem, for finding the shortest route that covers each edge of a given weighted graph, was originated by Meigu Guan (Mei-Ku Kwan) [Gu60] in 1960. (See §4.3.)

F76: In matching and assignment problems one wishes to assign people as appropriately as possible to jobs for which they are qualified. This work developed from work of König and from a celebrated result on matching due to Philip Hall [Ha35], later known as the "marriage theorem" [HaVa50]. These investigations led to the subject of polyhedral combinatorics and were combined with the newly emerging study of linear programming. (See §11.3.)

F77: By the late 1960s it became clear that some problems seemed to be more difficult than others, and Edmonds [Ed65] discussed problems for which a polynomial-time algorithm exists. Cook [Co71], Karp [Ka72], and others later developed the concept of NP-completeness. The assignment, transportation, and minimum spanning-tree problems are all in the *polynomial-time class* P, while the traveling salesman and Hamiltonian cycle problems are NP-hard. It is not known whether P = NP. Further information can be found in [GaJo79].

References

[ApHa77] K. Appel and W. Haken, Every planar map is 4-colorable: Part 1, Discharging, *Illinois J. Math.* 21 (1977), 429–490.

[ApHaKo77], K. Appel, W. Haken, and J. Koch, Every planar map is 4-colorable: Part 2, Reducibility, *Illinois J. Math.* 21 (1977), 429–490.

[BiLlWi98] N. L. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory 1736-1936*, Oxford University Press, 1998.

[Bi12] G. D. Birkhoff, A determinantal formula for the number of ways of coloring a map, *Ann. of Math.* 14 (1912), 42–46.

[Bi13] G. D. Birkhoff, The reducibility of maps, *Amer. J. Math.* 35 (1913), 115–128.

[BiLe46] G. D. Birkhoff and D. C. Lewis, Chromatic polynomials, *Trans. Amer. Math. Soc.* 60 (1946), 355–451.

[Bo26] O. Boruvka, O jistém problému minimálním, *Acta Soc. Sci. Natur. Moravicae* 3 (1926), 37–58.

[Br41] R. L. Brooks, On colouring the nodes of a network, *Proc. Cambridge Philos. Soc.* 37 (1941), 194–197.

[Ca:1813] A.-L. Cauchy, Recherches sur les polyèdres-premier mémoire, *J. Ecole Polytech.* 9 (Cah. 16) (1813), 68–86.

[Ca:1857] A. Cayley, On the theory of the analytical forms called trees, *Phil. Mag.* (4) 13 (1857), 172–176.

[Ca:1874] A. Cayley, On the mathematical theory of isomers, *Phil. Mag.* (4) 47 (1874), 444–446.

[Ca:1879] A. Cayley, On the colouring of maps, *Proc. Roy. Geog. Soc.* (new Ser.) 1 (1879), 259–261.

[Ca:1889] A. Cayley, A theorem on trees, *Quart. J. Pure Appl. Math.* 23 (1889), 376–378.

[Co71] S.A. Cook, The complexity of theorem-proving procedures, 151–158 in *Proc. 3rd Annual ACM Symp. Theory of Computing*, ACM, New York, 1971.

[Cl:1844] T. Clausen, [Second postscript to] De linearum tertii ordinis proprietatibus, *Astron. Nachr.* 21 (1844), col. 209–216.

[Cr99] P. R. Cromwell, *Polyhedra*, Cambridge University Press, 1999.

[Da82] G. B. Dantzig, Reminiscences about the origins of linear programming, *Oper. Res. Lett.* 1 (1982), 43–48.

[DaFuJo54] G. B. Dantzig, D. R. Fulkerson and S. M. Johnson, Solution of a large-scale traveling-salesman problem, *Oper. Res.* 2 (1954), 393–410.

[DeHe07] M. Dehn and P. Heegaard, Analysis situs, *Encyklopädie der Mathematischen Wissenschaften* (1907), 153–120.

[DeM:1860] A. De Morgan, A review of The philosophy of discovery, chapters historical and critical, by W. Whewell, D. D., *Athenaeum* No. 1694 (1860), 501–503.

[Di59] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numer. Math.* 1 (1959), 269–271.

[Di52] G. A. Dirac, Some theorems on abstract graphs, *Proc. London Math. Soc.* (3) 2 (1952), 69–81.

[Du13] H. E. Dudeney, Perplexities, *Strand Mag.* 46 (July 1913), 110 and (August 1913), 221.

[Ed65] J. R. Edmonds, Paths, trees and flowers, *Canad. J. Math.* 17 (1965), 449–467.

[Eu:1736] L. Euler, (1736) Solutio problematis ad geometriam situs pertinentis, *Commentarii Academiae Scientiarum Imperialis Petropolitanae* 8 (1752), 128–140.

[Eu:1759] L. Euler, Solution d'une question curieuse qui ne paroit soumise à aucune analyse, *Mem. Acad. Sci. Berlin* 15 (1759), 310–337.

[FiWi77] S. Fiorini and R. J. Wilson, *Edge-Colourings of Graphs*, Pitman, 1977.

[FoFu56] L. R. Ford and D. R. Fulkerson, Maximal flow through a network, *Canad. J. Math.* 8 (1956), 399–404.

[Fr22] P. Franklin, The four color problem, *Amer. J. Math.* 44 (1922), 225–236.

[Fr34] P. Franklin, A six color problem, *J. Math. Phys.* 13 (1934), 363–369.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., 1979.

[GlHUWa79] H. H. Glover, J. P. Huneke and C. S. Wang, 103 graphs that are irreducible for the projective plane, *J. Combin. Theory* (B) 27 (1979), 332–370.

[GoHu61] R. E. Gomory and T. C. Hu, Multi-terminal network flows, *SIAM J. Appl. Math.* 9 (1961), 551–556.

[Gr74], J. L. Gross, Voltage graphs, *Discrete Math.* 9 (1974), 239–246.

[GrTu74] J. L. Gross and T. W. Tucker, Quotients of complete graphs: revisiting the Heawood problem, *Pacific J. Math.* 55 (1974), 391–402.

[Gu60] Guan Meigu, Graphic programming using odd or even points, *Acta Math. Sinica* 10 (263-266 = 1962); *Chinese Math.* 1 (1960), 273–277.

[Gu63] W. Gustin, Orientable embedding of Cayley graphs, *Bull Amer. Math. Soc.* 69 (1963), 272–275.

[Ha35] P. Hall, On representatives of subsets, *J. London Math. Soc.* 10 (1935), 26–30.

[HaVa50] P. R. Halmos and H. E. Vaughan, The marriage problem, *Amer. J. Math.* 72 (1950), 214–215.

[Ha:1856] W. R. Hamilton, Memorandum respecting a new system of roots of unity, *Phil. Mag.* (4) 12 (1856), 446.

[Ha55] F. Harary, The number of linear, directed, rooted, and connected graphs, *Trans. Amer. Math. Soc.* 78 (1955), 445–463.

[HaPa73] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, 1973.

[He:1890] P. J. Heawood, Map-colour theorem, *Quart. J. Pure Appl. Math.* 24 (1890), 332–338.

[He:1891] L. Heffter, Über das Problem der Nachbargebiete, *Math. Ann.* 38 (1891), 477–580.

[He69] H. Heesch, Untersuchungen zum Vierfarbenproblem, *B. I. Hochschulscripten*, 810/810a/810b, Bibliographisches Institut, Mannheim-Vienna-Zürich, 1969.

[Hi:1873] C. Hierholzer, Über die Möglichkeit, einen Lineanzug ohne Wiederholung und ohne Unterbrechnung zu umfahren, *Math. Ann.* 6 (1873), 30–32.

[JeTo95] T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley-Interscience, 1995.

[Ka35] I. N. Kagno, A note on the Heawood color formula, *J. Math. Phys.* 14 (1935), 228–231.

[Ka72] R. M. Karp, Reducibility among combinatorial problems, 85-103 in *Complexity of Computer Computations* (ed. R. E. Miller and J. W. Thatcher), Plenum Press, 1972.

[Ke:1879] A. B. Kempe, On the geographical problem of four colours, *Amer. J. Math.* 2 (1879), 193–200.

[Ki:1856] T. P. Kirkman, On the representation of polyedra, *Phil. Trans. Roy. Soc. London* 146 (1856), 413–418.

[Kö16] D. König, Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, *Math. Ann.* 77 (1916), 453–465.

[Kr56] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, *Proc. Amer. Math. Soc.* 7 (1956), 48–50.

[Ku30] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fund. Math.* 15 (1930), 271–283.

[LLRS85] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan, and D. B. Schmoys (eds.), *The Traveling Salesman Problem: A Guided Tour through Combinatorial Optimization*, Wiley, 1985.

[Le:1794] A. M. Legendre, *Eléments de Géométrie* (1st edn.), Firmin Didot, Paris, 1794.

[Lh:1811] S. Lhuilier, Démonstration immédiate d'un théorème fondamental d'Euler sur les polyhèdres, et exceptions dont ce théorème est susceptible, *Mém. Acad. Imp. Sci. St. Pétersb.* 4 (1811), 271–301.

[Li:1847] J. B. Listing, Vorstudien zur Topologie, *Göttingen Studien* (Abt. 1) Math. Naturwiss. Abh. 1 (1847), 811–875.

[Li:1861-2] J. B. Listing, Der Census räumliche Complexe, *Abh. K. Ges. Wiss. Göttingen Math. Cl.* 10 (1861-2), 97–182.

[LoPl86] L. Lovász and M. D. Plummer, Matching Theory, *Annals of Discrete Mathematics* 29, North-Holland, 1986.

[Lu:1882] E. Lucas, *Récréations Mathématiques*, Vol. 1, Gauthier-Villars, Paris (1882).

[LuSe29] A. C. Lunn and J. K. Senior, Isomerism and configuration, *J. Phys. Chem.* 33 (1929), 1027–1079.

[Ma69] J. Mayer, Le problème des régions voisines sur les surfaces closes orientables, *J. Combin. Theory* 6 (1969), 177–195.

[Or60] O. Ore, Note on Hamiltonian circuits, *Amer. Math. Monthly* 67 (1960), 55.

[Ot48] R. Otter, The number of trees, *Ann. of Math.* 49 (1948), 583–599.

[PaRi87] M. W. Padberg and G. Rinaldi, Optimization of a 532-city symmetric traveling salesman problem by branch and cut, *Oper. Res. Lett.* 6 (1987), 1–7.

[Pe:1891] J. Petersen, Die Theorie der regulären Graphs, *Acta Math.* 15 (1891), 193–220.

[Pe:1898] J. Petersen, Sur le théorème de Tait, *Interméd. Math.* 5 (1898), 225–227.

[Po:1809-10] L. Poinsot, Sur les polygones et les polyèdres, *J. Ecole Polytech.* 4 (1809-10) (Cah. 10), 16–48.

[Pó37] G. Pólya, Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen, *Acta Math.* 68 (1937), 145–254.

[PóRe87] G. Pólya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*, Springer, 1987.

[Pr18] H. Prüfer, Neuer Beweis eines Satzes über Permutationen, *Arch. Math. Phys.* (3) 27 (1918), 142–144.

[Re63] R. C. Read, On the number of self-complementary graphs and digraphs, *J. London Math. Soc.* 38 (1963), 99–104.

[Re27] J. H. Redfield, The theory of group-reduced distributions, *Amer. J. Math.* 49 (1927), 433–455.

[Re:1871-3] M. Reiss, Evaluation du nombre de combinaisons desquelles les 28 dés d'un jeu du domino sont susceptibles d'après la règle de ce jeu, *Ann. Mat. Pura. Appl.* (2) 5 (1871-3), 63–120.

[Ri74] G. Ringel, *Map Color Theorem*, Springer, 1974.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Nat. Acad. Sci. U.S.A.* 60 (1968), 438–445.

[RoSe85] N. Robertson and P. D. Seymour, Graph minors — a survey, in *Surveys in Combinatorics 1985* (ed. I. Anderson), London Math. Soc. Lecture Notes Series 103 (1985), Cambridge University Press, 153–171.

[RoSaSeTh97] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, The four-colour theorem, *J. Combin. Theory, Ser. B* 70 (1997), 2–44.

[Ro:1892] W. W. Rouse Ball, *Mathematical Recreations and Problems of Past and Present Times* (later entitled *Mathematical Recreations and Essays*), Macmillan, London, 1892.

[SaStWi88] H. Sachs, M. Stiebitz and R. J. Wilson, An historical note: Euler's Königsberg letters, *J. Graph Theory* 12 (1988), 133–139.

[Sh49] C. E. Shannon, A theorem on coloring the lines of a network, *J. Math. Phys.* 28 (1949), 148–151.

[Sy:1877-8] J. J. Sylvester, Chemistry and algebra, *Nature* 17 (1877-8), 284.

[Sy:1878] J. J. Sylvester, On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, *Amer. J. Math.* 1 (1878), 64–125.

[Ta:1878-80] P. G. Tait, Remarks on the colouring of maps, *Proc. Roy. Soc. Edinburgh* 10 (1878-80), 729.

[Ti10] H. Tietze, Einige Bemerkungen über das Problem des Kartenfärbens auf einseitigen Flächen, *Jahresber. Deut. Math.-Ver.* 19 (1910), 155–179.

[Tu46] W. T. Tutte, On hamiltonian circuits, *J. London Math. Soc.* 21 (1946), 98–101.

[Tu47] W. T. Tutte, The factorizations of linear graphs, *J. London Math. Soc.* 22 (1947), 107–111.

[Tu59] W. T. Tutte, Matroids and graphs, *Trans. Amer. Math. Soc.* 90 (1959), 527–552.

[Tu70] W. T. Tutte, On chromatic polynomials and the golden ratio, *J. Combin. Theory* 9 (1970), 289–296.

[Va:1771] A.-T. Vandermonde, Remarques sur les problèmes de situation, *Mém. Acad. Sci. (Paris)* (1771), 556–574.

[Ve22] O. Veblen, *Analysis Situs*, Amer. Math. Soc. Colloq. Lect. 1916, New York, 1922.

[Vi64] V.G. Vizing, On an estimate of the chromatic class of a p-graph, *Diskret. Analiz* 3 (1964), 25–30.

[Vi65] V.G. Vizing, The chromatic class of a multigraph, *Diskret. Analiz* 5 (1965), 9–17.

[We:1904] P. Wernicke, Über den kartographischen Vierfarbensatz, *Math. Ann.* 58 (1904), 413–426.

[Wh31] H. Whitney, Non-separable and planar graphs, *Proc. Nat. Acad. Sci. U.S.A.* 17 (1931), 125–127.

[Wh35] H. Whitney, On the abstract properties of linear dependence, *Amer. J. Math.* 57 (1935), 509–533.

[Wi99] R. J. Wilson, Graph Theory, Chapter 17 in *History of Topology* (ed. I. M. James), Elsevier Science, 1999.

[Wi02] R. Wilson, *Four Colors Suffice*, Princeton University Press, 2002.

GLOSSARY FOR CHAPTER 1

acyclic graph: a graph with no *cycles.*

adding a vertex u to a graph $G = (V, E)$, where $u \notin V$: an operation yielding a new graph $G \cup \{u\}$ with vertex-set $V \cup \{u\}$ and edge-set E. (The new vertex u has no neighbors.)

adding an edge d (or uv) to a graph $G = (V, E)$: an operation yielding a new graph with vertex-set V and edge-set $E \cup \{d\}$ (or $E \cup \{uv\}$).

adjacency matrix – for a simple graph G with ordered vertex-set $v_1, v_2, \ldots, v_n$: the $n \times n$ matrix A_G such that

$$A_G(i, j) = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

adjacent edges: two edges that have an endpoint in common.

adjacent vertices: vertices u and v that are endpoints of the same edge.

arc: see *directed edge.*

attribute: an enhancement of the graph model; see *vertex attribute* and *edge attribute.*

automorphism – of a graph: an isomorphism of a graph to itself.

bipartite graph: a 2-partite graph; a graph whose vertices can be partitioned into two sets in such a way that no edge joins two vertices in the same set. (For technical reasons, the graph K_1 is sometimes construed to be bipartite.)

bouquet B_n: the (general) graph with one vertex and n self-loops.

cartesian product (or *product*) – of two graphs G and H: the graph $G \times H$ such that $V(G \times H) = V(G) \times V(H)$ and $E(G \times H) = E(G) \times V(H) \cup V(G) \times E(H)$.

Cayley graph$_1$ $C(\mathcal{A}, X)$ – for a group $\mathcal{A}$ with generating set X: the digraph whose vertex-set is $\mathcal{A}$, with an edge directed from a to ax for every $a \in \mathcal{A}$ and every $x \in X$. Sometimes two oppositely directed edges corresponding to an involution x are merged into a single undirected edge.

Cayley graph$_2$: any graph isomorphic to the underlying undirected graph of a Cayley graph $C(\mathcal{A}, X)$ is commonly said to be a “Cayley graph”.

center of a graph: the *induced subgraph* on the set of all central vertices.

central vertex – of a graph: a vertex whose eccentricity equals the radius of the graph.

chromatic number – of a graph G: the smallest number c of colors such that G is c-colorable.

circulant graph Circ$(n; X)$: a Cayley graph for a cyclic group $\mathbb{Z}_n$.

color: an attribute of vertices or edges or both; see *coloring.*

c-colorable graph: a graph that has a proper vertex-coloring with c or fewer colors.

coloring: an assignment to the vertices (or sometimes to the edges) of a graph of values in a set whose elements are regarded as “colors”; commonly these “colors” are integers or letters of the alphabet.

___, **proper**: an assignment of colors to the vertices (or edges) such that no two adjacent vertices (resp., edges) are assigned the same color.

complement: short form of *edge-complement.*

complete bipartite graph: a simple bipartite graph such that each vertex in one partite set is adjacent to every vertex in the other partite set. If the two partite sets have cardinalities r and s, then this graph is denoted $K_{r,s}$.

complete digraph $\overset{\leftrightarrow}{K}_n$: the simple digraph on n vertices such that between every pair of vertices, there is an arc in both directions.

complete graph K_n: the simple graph with n vertices in which every pair of vertices is joined by an edge.

complete k**-partite graph** $K_{n_1,n_2,\ldots,n_k}$: a simple k-partite graph such that two vertices are adjacent if and only if they are in different partite sets. All such graphs are called *complete multipartite graphs.*

component – of a graph: a maximal connected subgraph.

connected graph: a graph such that between every pair of vertices there is a walk.

k**-connected graph**: a graph such that the result of removing fewer than k vertices is a connected, nontrivial graph, for all possible choices of the vertices.

connectivity of a graph G: the largest number k such that G is k-connected. It is denoted $\kappa(G)$ or $\kappa_V(G)$.

cost – of an edge: (synonym for *weight*) a real number assigned to the edge.

critically k**-chromatic graph**: a graph of chromatic number k whose chromatic number would decrease if any edge were removed. (See §5.1.)

critically k**-connected graph**: a graph of connectivity k whose connectivity would decrease if any vertex were removed. (See §4.1.)

critically k**-edge-connected graph**: a graph of edge-connectivity k whose edge-connectivity would decrease if any edge were removed. (See §4.1.)

cube graph: see *hypercube graph.*

cube: see *hypercube.*

cut-edge: an edge whose removal increases the number of components of a graph.

cutpoint: see *cut-vertex.*

cut-vertex (or *cutpoint*): a vertex whose removal increases the number of components of a graph.

cycle: a closed path of length at least one.

cycle graph C_n: the n-vertex graph with n edges, such that every edge lies on a single cycle.

cycle rank – for a graph $G = (V, E)$ with $c(G)$ components: the number $|E(G)| - |V(G)| + c(G)$.

degree (or *valence*) – of a vertex v in a graph G, denoted $deg(v)$: the number of proper edges incident on v plus twice the number of self-loops. (For simple graphs, the degree of v equals the number of neighbors.)

degree sequence – of a graph: the sequence formed by arranging the vertex degrees in non-decreasing order.

deleting a vertex u from a graph $G = (V, E)$: an operation that not only removes the vertex u, but also removes every edge of which u is an endpoint. The new graph is denoted $G - u$.

deleting an edge d from a graph $G = (V, E)$: an operation that preserves its vertex-set V, but yields edge-set $E - \{d\}$. The new graph is denoted $G - d$.

diameter – of a connected graph: the maximum *eccentricity*, taken over all vertices.

digraph (or *directed graph*): a graph in which every edge is directed.

dipole D_n: the multigraph with two vertices and n edges joining them.

directed distance from a vertex u **to** a vertex v – in a digraph: the length of the shortest directed walk from u to v.

directed edge (or *arc*): an edge e, one of whose endpoints is designated as the *tail*, and whose other endpoint is designated as the *head*. In a line drawing, the arrow points toward the head.

directed graph: see *digraph*.

direction – on an edge: the designation of *head* and *tail*.

distance – between two vertices: the length of the shortest walk between them.

eccentricity of a vertex – in a connected graph: its distance to a vertex farthest from itself.

edge: a member of the constituent set E of a *graph* $G = (V, E)$.

edge-coloring – of a graph G: a function from its edge-set E_G to a set C whose elements are called *colors*.

edge attribute: a function from the edge-set to a set of possible attribute values, such as *direction* or *weight*.

edge-complement – of a simple graph G: the graph $\overline{G}$ (alternatively denoted G^c) that has the same vertex-set as G, such that uv is an edge of $\overline{G}$ if and only if it is *not* an edge of G.

k**-edge-connected graph**: a graph such that the result of removing fewer than k edges is connected and nontrivial for all possible choices of the edges.

edge-connectivity of a graph G: the largest number k such that G is k-edge-connected. It is denoted $\kappa'(G)$ or $\kappa_E(G)$.

edge-transitive graph: a graph such that all the edges are in the same *orbit*.

empty graph $\overline{K_n}$: the graph with n vertices and no edges.

endpoint: see *graph*.

eulerian graph: a graph with a closed walk that contains every edge exactly once. (See §1.3 for the history of eulerian graphs and §4.3 for an extensive discussion of their theory.)

eulerian trail – in a graph: a trail that contains every edge of the graph.

factor – of a graph: a spanning subgraph.

k**-factor** – of a graph: a k-regular spanning subgraph.

forest: a graph all of whose components are trees.

frontier edge – relative to a tree in a graph: an edge that has one endpoint in the tree and the other endpoint not in the tree; this concept arises when growing a spanning tree within a graph.

general graph: a graph from which the possible occurrence of self-loops and multi-edges is not explicitly excluded.

genus: see *minimum genus.*

graph $G = (V, E)$: a set V called *vertices* and a set E called *edges*, such that each edge has a set of one or two vertices associated to it, which are called its *endpoints.*

hamiltonian graph: a graph that has a spanning cycle. (See §1.3 for the history of hamiltonian graphs and §4.5 for an extensive discussion.)

head: see *directed edge.*

hypercube graph Q_d: the 1-skeleton of a d-dimensional hypercube.

hypercube of dimension d: the d-dimensional polytope $\{(x_1, \ldots, x_d) \mid 0 \geq x_j)\}$.

incidence: the relationship between an edge and one of its endpoints.

indegree – of a vertex v: the number of directed edges with v at the head.

induced subgraph – on a set of vertices $W = \{w_1, \ldots, w_k\}$ of a graph G: the subgraph on vertex-set W that contains every edge of G whose endpoints are in W.

invariant – see *isomorphism invariant.*

isolated vertex – in a graph: a vertex of degree 0.

isomorphic graphs: graphs G and H between which there is an isomorphism. Notation: $G \cong H$.

isomorphism invariant – of graphs: a graph property such that two isomorphic graphs always have the same value; for instance, the radius or the chromatic number.

isomorphism of general graphs G and H: a pair of bijections $\phi_V : V_G \to V_H$ and $\phi_E : E_G \to E_H$ such that for every pair of vertices $u, v \in V_G$, the set of edges in E_G joining u and v is mapped bijectively to the set of edges in E_H joining the vertices $\phi(u)$ and $\phi(v)$.

isomorphism of simple graphs G and H: a bijection $\phi : V_G \to V_H$ such that for every pair of vertices $u, v \in V_G$, there is an edge $uv \in E_G$ if and only if there is an edge in $\phi(u)\phi(v) \in E_H$.

join$_1$ (or *suspension*): an operation on two graphs G and H that yields the graph $G+H$ such that

$$V(G+H) = V(G) \cup V(H)$$
$$E(G+H) = E(G) \cup E(H) \cup \{uv \mid u \in V(G) \text{ and } v \in V(H)\}$$

join$_2$: the graph $G + H$ is called the “join” of G and H.

join$_3$: an edge is said to “join” its endpoints.

length of a walk: the number of edges.

line graph$_1$ – of a graph G: the simple graph $L(G)$ whose vertex-set is the the edge-set of G, and in which two vertices are adjacent if the edges in G to which they correspond have a common vertex.

line graph$_2$: A graph H is said to be a *line graph* if there exists a graph G such that H is isomorphic to $L(G)$.

loop: short for *self-loop.*

loopless graph: a graph with no *self-loops.*

minimum genus (or **genus**) of a connected graph G: the smallest number g such that G can be drawn on the orientable surface S_g (see §7.1) without any edge-crossings. Notation: $\gamma_{min}(G)$ or $\gamma(G)$.

mixed graph: synonym for *partially directed graph.*

multi-arc: a set of two or more arcs having the same tail and same head.

multi-edge: a collection of two or more edges, all having the same endpoints.

multigraph: a graph from which multi-edges are not excluded, but which has no self-loops.

neighbor – of a vertex: an adjacent vertex.

nontree edge – relative to a tree T in a graph G: an edge of G that is not in T.

null graph: a graph whose vertex-set and edge-set are empty; often denoted K_0.

octahedral graph $\mathcal{O}_d$: the edge-complement of a 1-factor in K_{2d}.

orbit of a vertex u – in a graph G: the set of all vertices $v \in V_G$ such that there is an automorphism ϕ with $\phi(u) = v$.

orbit of an edge d – in a graph G: the set of all edges $e \in E_G$ such that there is an automorphism ϕ with $\phi(d) = e$.

outdegree – of a vertex v: the number of directed edges with v at the tail.

partially directed graph: a graph that has both undirected and directed edges.

p**-partite graph**: a graph whose vertex-set V has a partition into p subsets $V_1, \ldots, V_p$ (called the *partite sets*), in such a way that no edge joins two vertices in the same subset.

partite sets: see *p-partite* graph.

path: a *walk* such that no edge or internal vertex occurs more than once.

___, **closed**: a *closed walk* such that no edge or internal vertex occurs more than once.

___, **open**: an *open walk* such that no edge or vertex occurs more than once.

path graph P_n: the n-vertex graph with $n-1$ edges, such that every edge lies on a single open path. (Quite commonly elsewhere, the subscript of the notation P_n denotes the number of edges.)

Petersen graph: a 10-vertex 3-regular graph, commonly depicted as a 5-pointed star inside a pentagon, with a 1-factor joining the vertices of the pentagon to the points of the star.

planar graph: a graph of minimum genus 0, i.e., a graph that can be drawn in the sphere or plane with no edge crossings.

platonic graph: the skeleton of any of the five platonic solids.

platonic solid: any of the five regular 3-dimensional polyhedra — tetrahedron, cube, octahedron, dodecaheron, icosahedron.

product: see *cartesian product.*

proper edge: an edge that joins two distinct vertices.

proper edge-coloring: an edge-coloring such that no two adjacent edges have the same color.

proper vertex-coloring: a vertex-coloring such that no two adjacent vertices have the same color.

radius – of a connected graph: the minimum *eccentricity*, taken over all vertices.

regular graph: a graph in which every vertex is of the same degree. It is *k-regular* if every vertex is of degree k.

self-loop: an edge that joins a single endpoint to itself.

self-union, m-fold – of a graph G: is the iterated disjoint union $mG = G \cup \cdots \cup G$ of m copies of the graph G.

simple digraph: a digraph with no self-loops and no multi-arcs.

simple graph: a graph that has no self-loops or multi-edges.

simplex: the convex hull of a set S of affinely independent points in Euclidean space. It is a k-*simplex* if $|S| = k + 1$.

skeleton (or 1-*skeleton*) of a k-complex K: the graph consisting of the vertices and the edges of K.

spanning subgraph – of a graph G: a subgraph that includes every vertex of G.

spanning tree – in a graph G: a subgraph that is a tree.

standard (vertex) labeling – of a graph with n vertices: a biective labeling by the numbers $0, \ldots, n-1$ or by the numbers $1, \ldots, n$.

strongly connected digraph: a digraph such that from each vertex to every other vertex there is a directed walk.

subgraph of a graph G: a graph H such that $V_H \subset V_G$ and $E_H \subset E_G$, or any graph isomorphic to such a graph H.

suspension: see *join.*

tail: see *directed edge.*

trail: a walk such that no edge occurs more than once.

tree: a connected graph with no cycles.

tree edge – relative to a tree T in a graph G: an edge of G that is in T.

trivial graph K_1: the graph with one vertex and no edges.

trivial walk, trail, or path: a walk of length zero.

underlying graph – of a graph with vertex or edge attributes: the graph that results from eliminating all the attribute values (e.g., deleting all the edge-directions).

union operation – on two graphs G and H: produces the graph $G \cup H$ whose vertex-set and edge-set are the disjoint unions, respectively, of the vertex-sets and edge-sets of G and H.

valence: synonym for *degree.*

vertex attribute: a function from the vertex-set to some set of possible attribute values.

vertex-coloring – of a graph G: a function from its vertex-set V_G to a set C whose elements are called *colors.*

vertex: a member of the constituent set V of a *graph* $G = (V, E)$.

vertex-transitive graph: a graph such that all the vertices are in the same *orbit.*

walk – in a graph G: an alternating sequence $W = v_0, e_1, v_1, e_1, \ldots, e_n, v_n$, such that for $j = 1, \ldots, n$, vertices v_{j-1} and v_j are the endpoints of the edge e_j.

___, **closed**: a walk whose final vertex is the same as its initial vertex.

___, **open**: a walk whose final vertex is different from its initial vertex.

weakly connected digraph: a digraph whose *underlying graph* is connected.

weight – of an edge: (also called *cost*) a real number assigned to the edge.

Chapter 2

GRAPH REPRESENTATION

2.1 COMPUTER REPRESENTATIONS OF GRAPHS

Alfred V. Aho, Columbia University

2.2 THE GRAPH ISOMORPHISM PROBLEM

Mark Goldberg, Rensselaer Polytechnic Institute

2.3 THE RECONSTRUCTION PROBLEM

Josef Lauri, University of Malta, Malta

2.4 RECURSIVELY CONSTRUCTED GRAPHS

R. B. Borie, University of Alabama
R. Gary Parker, Georgia Institute of Technology
C. A. Tovey, Georgia Institute of Technology

GLOSSARY

2.1 COMPUTER REPRESENTATIONS OF GRAPHS

Alfred V. Aho, Columbia University

2.1.1 The Basic Representations for Graphs
2.1.2 Graph Traversal Algorithms
2.1.3 All-Pairs Problems
2.1.4 Applications to Pattern Matching
References

Introduction

Many problems in science and engineering can be modeled in terms of directed and undirected graphs. The data structures and algorithms used to represent graphs can have a significant impact on the size of problems that can be implemented on a computer and the speed with which they can be solved. This section presents the fundamental representations used in computer programs for graphs and illustrates the tradeoffs among the representations using key algorithms for some of the most common graph problems.

Throughout this section we use the notation $|X|$ to denote the number of elements in a set X. The graphs and digraphs in this section are assumed to be simple.

2.1.1 The Basic Representations for Graphs

The two most basic representations for a graph are the adjacency matrix and the adjacency list.

DEFINITIONS

D1: A ***directed graph*** or ***digraph*** $G = (V, E)$ consists of a finite, nonempty set of *vertices* V and a set of *edges* E. Each edge is an ordered pair (v, w) of vertices.

D2: An ***undirected graph*** $G = (V, E)$ consists of a finite, nonempty set of *vertices* V and a set of *edges* E. Each edge is a set $\{v, w\}$ of vertices.

D3: In a directed graph $G = (V, E)$, vertex w is ***adjacent*** to vertex v if (v, w) is an edge in E. The number of vertices adjacent to v is called the ***out-degree*** of v.

D4: In an undirected graph $G = (V, E)$, vertex w is ***adjacent*** to vertex v if $\{v, w\}$ is an edge in E. The number of vertices adjacent to v is called the ***degree*** of v.

D5: A ***path*** in a directed or undirected graph is a seqence of edges (v_1, v_2), (v_2, v_3),..., (v_{n-1}, v_n). This path is from vertex v_1 to vertex v_n and has length $n - 1$.

D6: A graph $G = (V, E)$ is ***dense*** when the number of edges is close to $|V|^2$.

D7: A graph $G = (V, E)$ is ***sparse*** when the number of edges is much less than $|V|^2$.

D8: An ***adjacency matrix*** representation for a simple graph or digraph $G = (V, E)$ is a $|V| \times |V|$ matrix A, where $A[i, j] = 1$ if there is an edge from vertex i to vertex j; $A[i, j] = 0$ otherwise.

D9: An ***adjacency list*** representation for a graph or digraph $G = (V, E)$ is an array L of $|V|$ lists, one for each vertex in V. For each vertex i, there is a pointer L_i to a linked list containing all vertices j adjacent to i. A linked list is terminated by a ***nil pointer***.

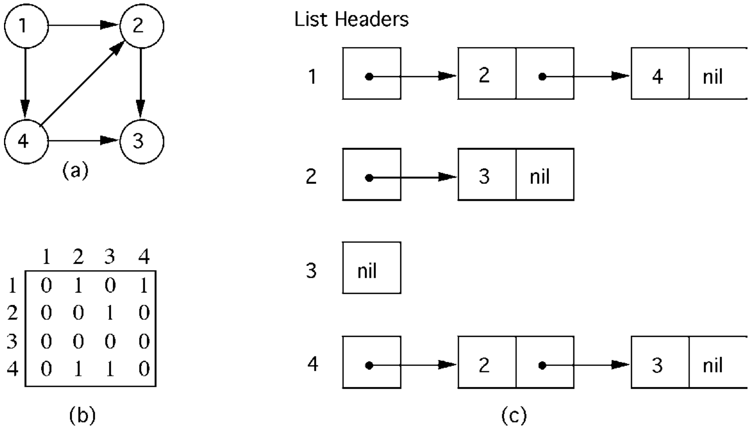
D10: An ***incidence matrix*** representation for a simple digraph $G = (V, E)$ is a $|V| \times |E|$ matrix I, where

$$I[v, e] = \begin{cases} -1 & \text{if edge } e \text{ is directed } \textit{to} \text{ vertex } v \\ 1 & \text{if edge } e \text{ is directed } \textit{from} \text{ vertex } v \\ 0 & \text{otherwise} \end{cases}$$

For an undirected graph, $I[v, e] = 1$ if e is incident on v and 0 otherwise.

EXAMPLES

E1: Figure 2.1.1 shows the adjacency matrix and adjacency list representations of a directed graph.



	1	2	3	4
1	0	1	0	1
2	0	0	1	0
3	0	0	0	0
4	0	1	1	0

Figure 2.1.1 **(a) A directed graph G. (b) Adjacency matrix for G. (c) Adjacency list representation for G.**

E2: An incidence matrix for the example digraph is shown below.

$$I_G = \begin{array}{c|ccccc} & (1,2) & (1,4) & (2,3) & (4,2) & (4,3) \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & -1 & 0 & 1 & -1 & 0 \\ 3 & 0 & 0 & -1 & 0 & -1 \\ 4 & 0 & -1 & 0 & 1 & 1 \end{array}$$

FACTS

F1: An adjacency matrix representation for a graph $G = (V, E)$ always takes $O(|V|^2)$ space.

F2: An adjacency list representation for a graph $G = (V, E)$ takes $O(|V| + |E|)$ space.

REMARKS

R1: For a more detailed discussion of graph representations, see [AhHoUl74], [AhHoUl83], [CoLeRiSt01], [Ev79], [Ta83].

R2: As a general rule, an adjacency list representation is preferred when a graph is sparse, because it takes space that is linearly proportional to the number of vertices and edges.

R3: When a graph $G = (V, E)$ is dense, both an adjacency matrix and an adjacency list representation require $O(|V|^2)$ space. However, with the adjacency matrix, we can determine whether an edge exists in constant time, whereas with the adjacency list we may need $O(|V|)$ time. For this reason, adjacency matrix representations are often used with dense graphs.

R4: Note that in an adjacency list representation of an undirected graph, an edge i, j appears on two adjacency lists: the list for vertex i and the list for vertex j.

2.1.2 Graph Traversal Algorithms

One of the most fundamental tasks in algorithms involving graphs is visiting the vertices and edges of a graph in a systematic order. Depth-first and breadth-first search are frequently used traversal techniques for both directed and undirected graphs. For both these techniques, the adjacency list representation of a graph works well.

Depth-First Search

ALGORITHM

Depth-first search systematically visits all the vertices of a graph. Initially, all vertices are marked "new". When a vertex is visited, it is marked "old". Depth-first search works by selecting a new vertex v, marking it old, and then calling itself recursively on each of the vertices adjacent to v. The algorithm below is called "depth-first search" because it searches along a path in the forward (deeper) direction looking for new vertices as long as it can.

DEFINITIONS

During the course of its traversal, depth-first search partitions the graph into a collection of ***depth-first trees*** that make up a ***depth-first forest***. The forest and its trees are determined by the edges, which are partitioned by the search into four sets:

D11: ***Tree edges*** are those edges (v, w) where w is first encountered by exploring edge (v, w).

D12: ***Back edges*** are those edges (v, w) that connect a vertex v to an ancestor w in a depth-first tree.

D13: ***Forward edges*** are those nontree edges (v, w) that connect a vertex v to a proper descendant in a depth-first tree.

D14: ***Cross edges*** are the remaining edges. They connect vertices that are neither ancestors nor descendants of one another.

Algorithm 2.1.1: Depth-First Search

Input: A graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$ and $L[v]$ is a pointer to the list of vertices adjacent to vertex v.

Output: Traversal of all vertices in V in a depth-first order.

```
procedure $DepthFirstSearch(G)\{$
  for $v := 1$ to $n$ do
    $mark[v] := new;$
  for $v := 1$ to $n$ do
    if $mark[v] = new;$ then
      $dfs(v);$
$\}$
procedure $dfs(v)\{$
  $mark[v] := old;$
  for each vertex $w$ on $L[v]$ do
    if $mark[w] = new$ then
      $dfs(w);$
$\}$
```

FACTS

F3: Depth-first search takes $O(|V| + |E|)$ time on a graph $G = (V, E)$.

F4: If we represent the first visit of a vertex v with a left parenthesis "$(v$" and its last visit by a right parenthesis "$v)$", then the sequence of first and last visits forms an expression in which the parentheses are properly nested.

F5: In a depth-first search of an undirected graph, every edge is either a tree edge or a back edge.

REMARKS

R5: Depth-first search is a fundamental graph algorithm that has been in use since the 1950s. [Ta72 and HoTa73] developed several efficient graph algorithms using depth-first search.

R6: Depth-first search forms the basis of many important graph algorithms such as determining the biconnected components of an undirected graph and finding the strongly connected components of a directed graph.

Breadth-First Search

Breadth-first search is another fundamental technique for exploring a graph G. It starts from a specified *source* vertex s from which it constructs a ***breadth-first tree*** consisting of all vertices of G reachable from s. In the process it computes a breadth-first tree rooted at s such that if a vertex v is reachable from s in G, there is a path in the tree from the root to s. The path in the tree is a shortest path from s to v in G.

ALGORITHM

Breadth-first search uses the abstract data type *queue* to hold vertices as they are being processed. The operation $enqueue(s, Q)$ places vertex s on the back of the queue Q. The operation $dequeue(Q)$ removes the element at the front of the queue Q.

Breadth-first search visits the vertices of G uniformly across the breadth of the frontier of its search, visiting all vertices distance d from s before looking for vertices at distance $d + 1$. In constrast, depth-first search plunges as deeply into the graph along a path as it can before backtracking to visit nodes closer to s.

Algorithm 2.1.2: Breadth-First Search

Input: A graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$, $L[v]$ is a pointer to the list of vertices adjacent to vertex v, and s is a specified source vertex.

Output: A breadth-first tree consisting of root s and all vertices in V reachable from s.

```
procedure $BreadthFirstSearch(G, s)$ {
  for $v := 1$ to $n$ do {
    $mark[v] := new$;
    $distance[v] := \infty$;
    $parent[v] :=$ nil;
  }
  $mark[s] := visited$;
  $distance[s] := 0$;
  initialize queue $Q$;
  $enqueue(s, Q)$;
  while $Q$ is not empty do {
    $v := dequeue(Q)$;
    for each vertex $w$ on $L[v]$ do
      if $mark[w] = visited$ then {
        $mark[w] := visited$;
        $distance[w] := distance[w] + 1$;
        $parent[w] := v$;
        $enqueue(w, Q)$;
      }
  }
}
```

DEFINITION

D15: Let BFT be the tree with root s, vertices v such that $parent[v]$ is not **nil**, and edges $\{(parent[v], v) | parent[v]$ is not **nil**$\}$. BFT is the ***breadth-first tree*** constructed by $BreadthFirstSearch(G, s)$.

FACTS

F6: Breadth-first search takes $O(|V| + |E|)$ time on a graph $G = (V, E)$.

F7: $BreadthFirstSearch(G, s)$ computes the length of the shortest path from s to v in $distance[v]$.

REMARKS

R7: Like depth-first search, breadth-first search has been used since the 1950s. Early applications of breadth-first search included maze searching and routing wires on printed circuit boards.

R8: The ideas found in breadth-first search are the building blocks of many other graph algorithms such as Dijkstra's single-source shortest-paths algorithm and Prim's algorithm for finding minimal spanning trees.

2.1.3 All-Pairs Problems

This section considers two algorithms: one for computing the shortest paths between all pairs of vertices in a directed graph and the other for computing the transitive closure of a directed graph. For both algorithms the adjacency matrix is a natural representation for the graph.

All-Pairs Shortest-Paths Algorithm

Suppose that we have a schedule that tells us the driving time between n cities at a given time of day and that we wish to compute the shortest driving time between all pairs of cities. This is an instance of the *all-pairs shortest-paths* problem. We could iterate through every pair of cities and compute the shortest path between each using a single-source shortest-path algorithm such as Dijkstra's algorithm.

ALGORITHM

An easier way is to use the Floyd-Warshall algorithm below. The natural representation for a graph in the Floyd-Warshall algorithm is an adjacency matrix. Assume that we are given a directed graph $G = (V, E)$ and that the vertices in V are numbered $1, 2, ..., n$. Further assume that we are given a matrix $C[i, j]$ that tells us the cost of edge (i, j). If there is no edge $C[i, j]$, then we assume $C[i, j]$ is set to infinity. We assume all other costs are nonnegative.

The Floyd-Warshall algorithm computes a cheapest-cost array A, where $A[i, j]$ gives the cheapest cost of any path from vertex i to vertex j. For the algorithm to work correctly, it is important that there are no negative cost cycles in the graph.

FACT

F8: The Floyd-Warshall algorithm computes the cost matrix of the cheapest paths between all pairs of vertices of a directed graph $G = (V, E)$ in $O(|V|^3)$ time and $O(|V|^2)$ space.

REMARKS

R9: For additional discussion of the Floyd-Warshall algorithm and its variants see [AhHoUl74] and [CoLeRiSt01].

Algorithm 2.1.3: Floyd-Warshall

Input: A directed graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$ and a cost matrix $C[i, j]$.
Output: Cost matrix $A[1..n, 1..n]$ where $A[i, j]$ is the cost of the cheapest path from i to j.

```
procedure $FloydWarshall(G)$ {
  for $i := 1$ to $n$ do
    for $j := 1$ to $n$ do
      $A[i, j] := C[i, j]$;
  for $i := 1$ to $n$ do
    $A[i, i] := 0$;
  for $k := 1$ to $n$ do
    for $i := 1$ to $n$ do
      for $j := 1$ to $n$ do
        if $A[i, k] + A[k, j] < A[i, j]$ then
          $A[i, j] := A[i, k] + A[k, j]$;
}
```

R10: Let $A^k[i, j]$ be the cost of the cheapest path from vertex i to vertex j that does not pass through a vertex numbered higher than k, except possibly for the endpoints. We can prove by induction on k that $A^k[i, j] = min(A^{k-1}, A^{k-1}[i, k] + A^{k-1}[k, j])$. In the next section we see that the Floyd-Warshall algorithm is a special case of Kleene's algorithm.

Transitive Closure

In some problems we may just want to know whether there exists a path from vertex i to vertex j of length one or more in a graph $G = (V, E)$. We call this the problem of computing the ***transitive closure*** of G. Given a directed graph $G = (V, E)$ with adjacency matrix A, we want to compute a Boolean matrix T such that $T[i, j]$ is 1 if there is a path from i to j of length 1 or more, and 0 otherwise. We call T the *transitive closure* of the adjacency matrix.

The transitive-closure algorithm below is similar to the Floyd-Warshall algorithm except that it uses the Boolean operation **and** to conclude that if there is a path from i to k and one from k to j, then there is a path from i to j.

FACT

F9: The algorithm $TransitiveClosure(G)$ computes the transitive closure of G in $O(|V|^3)$ time and $O(|V|^2)$ space.

REMARKS

R11: The transitive closure algorithm is due to S. Warshall [Wa62].

Algorithm 2.1.4: Transitive Closure

Input: A directed graph $G = (V, E)$, with $V = \{1, 2, ..., n\}$ and adjacency matrix $A[i, j]$.
Output: Boolean transitive-closure matrix $T[1..n, 1..n]$ where $T[i, j]$ is 1 if there is a path from i to j if length 1 or more, 0 otherwise.

```
procedure $TransitiveClosure(G)$ {
  for $i := 1$ to $n$ do
    for $j := 1$ to $n$ do
      $T[i, j] := A[i, j]$;
  for $k := 1$ to $n$ do
    for $i := 1$ to $n$ do
      for $j := 1$ to $n$ do
        if $A[i, j] =$ false then
          $A[i, j] := A[i, k]$or$A[k, j]$;
}
```

R12: Let $T^k[i, j] = 1$ if there is a path of length one or more from vertex i to vertex j that does not pass through an intermediate vertex numbered higher than k, except for the endpoints. We can prove by induction on k that

$$C^k[i, j] = C^{k-1}[i, j] \textbf{ or } C^{k-1}[i, k] \textbf{ and } C^{k-1}[k, j]$$

where **and** and **or** are the Boolean *and* and *or* operators. In the next subsection we will see the transitive closure algorithm is a special case of Kleene's algorithm.

2.1.4 Applications to Pattern Matching

Graphs play a major role in problems arising in the specification and translation of programming languages. A special kind of graph called a ***finite automaton*** is used in language theory to specify and recognize sets of strings called ***regular expressions***. Regular expressions are used to specify the lexical structure of many programming language constructs. They are also widely used in many string-pattern-matching applications.

This section presents an algorithm due to S. C. Kleene to construct representations called ***regular expressions*** for all paths between the vertices of a directed graph.

DEFINITIONS

D16: A ***nondeterministic finite automaton*** (NFA) is a labeled, directed graph $G = (V, E)$ in which

1. one vertex is distinguished as the *start* vertex
2. a set of vertices are distinguished as *final* vertices
3. each edge is labeled by a symbol from a set $\Sigma \cup \{\epsilon\}$ where

   Σ is a finite set of *alphabet symbols*, and

   ϵ is a special symbol denoting the empty string

D17: An NFA G ***accepts*** a string x if there is a path in G from the start vertex to a final vertex whose edge labels spell out x.

D18: The set of strings accepted by an NFA G is called the ***language*** defined by G.

D19: If R and S are sets of strings, then their ***concatenation*** $R \cdot S$ is the set of strings $\{xy|x$ is in r and y is in $S\}$.

D20: Let S be a set of strings. Define $S^0 = \{\epsilon\}$ and $S^i = S \cdot S^{i-1}$ for $i \geq 1$. The ***Kleene closure*** of S, denoted S^*, is defined to be $\cup_{i=0}^{\infty} S^i$.

D21: Let Σ be a finite set of alphabet symbols. The ***regular expressions*** over Σ and the languages they denote are defined recursively as follows:

1. ϕ is a regular expression that denotes the empty set.
2. ϵ is a regular expression that denotes $\{\epsilon\}$
3. For each a in Σ, a is a regular expression that denotes $\{a\}$
4. If r and s are regular expressions denoting the languages R and S, then

   $(r + s)$ is a regular expression denoting the language $R \cup S$,

   rs is a regular expression denoting $R \cdot S$, and

   (r^*) is a regular expression denoting R^*.

We can avoid writing many parentheses in a regular expression by adopting the convention that the Kleene closure operator * has higher precedence than concatenation or $+$, and that concatenation has higher precedence than $+$. For example, $((a(b^*)) + c)$ may be written $ab^* + c$. This regular expression denotes the set of strings $\{ab^i | i \geq 0\} \cup \{c\}$.

Kleene's Algorithm

S. C. Kleene presented an algorithm for constructing a regular expression from a nondeterministic finite automaton. This algorithm, shown below, includes the Floyd-Warshall algorithm and the transitive closure algorithm as special cases.

ALGORITHM

Let $G = (V, E)$ be an NFA in which the vertices are numbered $1, 2, ..., n$. Kleene's algorithm (see below) works by constructing a sequence of matrices C^k in which the entry $C^k[i, j]$ is a regular expression for all paths from vertex i to vertex j with no intermediate vertex on the path (except possibly for the endpoints) that is numbered higher than k.

REMARKS

R13: Kleene's algorithm appeared in [Kl56].

R14: To prove the correctness of Kleene's algorithm, we can prove by induction on k that $C^k[i, j]$ is the set of path labels of all paths from vertex i to vertex j with no intermediate vertex numbered higher than k, excluding the endpoints. The term $C^{k-1}[i, k]$ in the inner loop represents the labels of all paths from vertex i to vertex k that do not have an intermediate vertex numbered higher than $k - 1$. The term $(C^{k-1}[k, k])^*$ represents the labels of all paths that go from vertex k to vertex k zero or more times without passing through an intermediate vertex numbered higher than $k - 1$. The term $C^{k-1}[k, j]$ represents the labels of all paths from vertex k to vertex

Algorithm 2.1.5: Kleene's Algorithm

Input: A directed graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$, and a label matrix $L[i, j]$.
Output: Matrix $C[1..n, 1..n]$ where $C[i, j]$ is a regular expression describing all paths from i to j.

```
procedure $Kleene(G)$ {
  for $i := 1$ to $n$ do
    for $j := 1$ to $n$ do
      $C^0[i, j] := L[i, j]$;
  for $i := 1$ to $n$ do
    $C^0[i, i] := \epsilon + C^0[i, i]$;
  for $k := 1$ to $n$ do
    for $i := 1$ to $n$ do
      for $j := 1$ to $n$ do
        $C^k[i, j] := C^{k-1}[i, j] + C^{k-1}[i, k] \cdot (C^{k-1}[k, k])^* \cdot C^{k-1}[k, j]$;
  for $i := 1$ to $n$ do
    for $j := 1$ to $n$ do
      $C[i, j] := C^n[i, j]$;
}
```

j that do not have an intermediate vertex numbered higher than $k - 1$. Thus, the term $C^{k-1}[i, j] \cdot (C^{k-1}[k, k])^* \cdot C^{k-1}[k, j])$ represents the path labels of all paths with the segments: from i to k, from k to k zero or more times, and from k to j with no intermediate vertex numbered higher than $k - 1$ on any of the segments.

R15: The Floyd-Warshall algorithm is a special case of Kleene's algorithm with the inner loop replaced by $C^k[i, j] := min(C^{k-1}[i, j], C^{k-1}[i, k] + C^{k-1}[k, j]$. In the Floyd-Warshall algorithm we don't need to consider paths from k to k since we assume the edge costs are nonnegative. Also, in the Floyd-Warshall algorithm the operator representing concatenation $(\cdot)$ is arithmetic addition.

R16: The transitive closure algorithm is a special case of Kleene's algorithm with the inner loop replaced by $C^k[i, j] := C^{k-1}[i, j] + C^{k-1}[i, k] \cdot C^{k-1}[k, j]$ where $+$ represents Boolean **or** and $\cdot$ represents Boolean **and**.

R17: Aho, Hopcroft, and Ullman present Kleene's algorithm in the general setting of a closed semiring [AhHoUl74].

R18: One of the key results of formal language theory is that the set of languages defined by NFAs is exactly the same as the set of languages defined by regular expressions. These languages are called ***regular sets***.

R19: For applications of finite automata and regular expressions to string pattern matching and compiling see [Ah90, AhSeUl86].

References

[Ah90] A. V. Aho, Algorithms for finding patterns in strings, pp. 255–300 in *Handbook of Theoretical Computer Science A, Algorithms and Complexity*, Ed. J. Van Leeuwen, MIT Press, 1990.

[AhHoUl74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley, 1974.

[AhHoUl83] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms,* Addison-Wesley, 1983.

[AhSeUl86] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques, and Tools,* Addison-Wesley, 1986.

[CoLeRiSt01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition,* MIT Press, 2001.

[Ev79] S. Even, *Graph Algorithms,* Computer Science Press, 1979.

[Fl62] R. W. Floyd, Algorithm 97 (Shortest Path), *Comm. ACM* 5(6) (1962), 345.

[HoTa73] J. E. Hopcroft and R. E. Tarjan, Efficient algorithms for graph manipulation, *Comm. ACM* 16(6) (1973), 372–378.

[Jo77] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, *J. ACM* 24(1) (1977), 1–13.

[Kl56] S. C. Kleene, Representation of events in nerve nets and finite automata, pp. 3–40 in *Automata Studies*, Eds. C. E. Shannon and J. McCarthy, Princeton Univ. Press, 1985.

[Ta72] R. E. Tarjan, Depth first search and linear graph algorithms, *SIAM J. Comput.* 1(2) (1972), 146–160.

[Ta83] R. E. Tarjan, *Data Structures and Network Algorithms,* Soc. Industrial and Applied Mathematics, 1983.

[Wa62] S. Warshall, A theorem on Boolean matrices, *J. ACM* 9(1) (1962), 11–12.

2.2 THE GRAPH ISOMORPHISM PROBLEM

Mark Goldberg, Rensselaer Polytechnic Institute

2.2.1 Variations of the Problem
2.2.2 Refinement Technique
2.2.3 Practical Graph Isomorphism
2.2.4 Group-Theoretic Approach
2.2.5 Complexity
References

Introduction

The ***graph isomorphism problem***, abbreviated ***ISO***, is to construct an efficient algorithm for testing whether two given graphs are isomorphic. In the context of *ISO*, *efficiency* is interpreted as subexponentiality. It is expected (folklore) that testing for isomorphism can be done in $O(\exp(\log^2 n))$, or maybe even in $O(\exp(\log n \log\log n))$ time. The currently fastest algorithm for *ISO* runs in $O(\exp(n^{1/2+o(1)})$ time. *ISO* is fundamental for graph theory as well as for computational complexity theory.

Throughout this section, graphs are assumed to be simple, unless specifically mentioned otherwise.

NOTATION: $\mathcal{G}$ denotes the class of all graphs.

2.2.1 Variations of the Problem

DEFINITIONS

D1: Two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are ***isomorphic***, denoted $G_1 \approx G_2$, if there is a one-to-one, onto mapping $\phi : V_1 \to V_2$, such that for any two vertices $x, y \in V_1$, there is an edge xy in G_1 if and only if there is an edge $\phi(x)\phi(y)$ in G_2. Such an adjacency-preserving bijection ϕ is called an ***isomorphism***.

D2: A ***labeled graph*** is a graph whose vertices and/or edges are labeled, possibly with repetitions, using symbols from a finite alphabet. If some vertices and/or edges have no labels, then they can be regarded as having a special label different from the rest. Thus, we may always assume that all vertices and all edges are labeled.

D3: Two labeled graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are ***isomorphic as labeled graphs*** if there is an isomorphism $\phi : G_1 \to G_2$, such that for each $v \in V_1$, the vertices v and $\phi(v)$ have the same label.

D4: A hypergraph G is a pair (V, E), where V is a set of elements called ***vertices*** and E is a collection of non-empty subsets of V, called ***hyperedges***. Two hypergraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are ***isomorphic***, denoted $G_1 \approx G_2$, if there is a one-to-one, onto mapping $\phi : V_1 \to V_2$, such that for every member s of E_1, $\phi(s) \in E_2$; and for every hyperedge t of E_2, there is a hyperedge $s \in E_1$, such that $t = \phi(s)$.

D5: A ***certificate*** for isomorphism is a graph invariant ι such that for any two graphs $G_1, G_2 \in \mathcal{G}$, $\iota(G_1) = \iota(G_2)$ if and only if $G_1 \approx G_2$.

D6: A ***canonical numbering algorithm*** for graphs is an algorithm $\mathcal{N}$ that outputs a permuted sequence

$$\mathcal{N}(G) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}\}$$

of the vertices of its input graph, such that two graphs $G = (\{v_1, \ldots, v_n\}, E)$ and $H = (\{u_1, \ldots, u_n\}, F)$ are isomorphic if and only if the bijection $\{v_{i_j} \mapsto u_{i_j} : j = 1, \ldots, n\}$ is a graph isomorphism. The vertex ordering produced by a canonical numbering algorithm for a given graph G is called a ***canonical ordering*** of the vertices of G.

FACTS

F1: Two graphs G_1 and G_2 with respective adjacency matrices A_1 and A_2 are isomorphic if and only if there is a permutation matrix P, such that

$$A_2 = P \times A_1 \times P^{-1}$$

This holds also when the definition of isomorphism is extended to directed graphs and to graphs with multiple adjacencies and/or self-loops.

F2: The problem of testing labeled graphs for isomorphism is polynomially equivalent to *ISO*.

F3: The problem of testing hypergraphs for isomorphism is polynomially equivalent to *ISO*.

EXAMPLES

E1: Examples of labeled graphs are *chemical graphs* representing *molecules*; the labels of the vertices of such a graph are the names of the atoms, and the labels of edges are integers indicating the multiplicity of the links; usually, the links are not more than tripled. Sometimes, instead of showing labels on the edges, the edges are replicated (see [Kl95]).

E2: Given an ordering $\tau = \langle v_1, \ldots, v_n \rangle$ of vertices of a graph G, let $S_\tau(G)$ denote the $0,1$-vector obtained by concatenating the rows of the adjacency matrix $A_\tau(G)$ in their natural order. We define $S(G)$ to be the lexicographically smallest such bit-vector. Then the invariant $G \mapsto S(G)$ is a certificate for graphs. The corresponding vertex ordering is a canonical numbering of the vertices of the graph.

2.2.2 Refinement Technique

The technique of *refinement* is often used to establish whether two given graphs are isomorphic. Sometimes it is helpful in constructing an isomorphism between the two graphs. This technique, started in [CoGo70], [We76], [We79] is simple, intuitive and powerful. Most efficient heuristics and randomized algorithms for *ISO* are based on some implementation of refinement (see [AHU74], [BaErSe80], [Ba97], [CoBo81], [CoGo84], [DeDaLo77], [FiMa80], [FüScSp83], [GHLSW87], [Go83], [HoWo74],[Kl95], [Ma79], [Mc81]). The software package **nauty** (`http://cs.anu.edu.au/~bdm/nauty`) — a widely used program for *ISO* — is based on this technique.

DEFINITIONS

D7: A ***coloring of a graph*** G is a mapping $\sigma : V_G \to C$ from its vertex set to a set C (often a set of integers).

D8: A ***trivial coloring*** assigns the same single color to every vertex.

D9: A ***color class*** for a graph with a coloring is the set of all vertices that are assigned the same color. Alternatively, a coloring $\sigma : V_G \to C$ may be regarded as a partition $\sigma = [C_1, \ldots, C_m]$ of V into the color classes.

D10: The ***neighborhood*** of a vertex v of a graph is the set of all vertices adjacent to v. It is denoted by $N(v)$.

D11: The ***degree vector*** of a graph coloring $\sigma = [C_1, \ldots, C_m]$ assigns to each vertex v the vector

$$\vec{deg}_\sigma(v) = [|N(v) \cap C_1|, \ldots, |N(v) \cap C_m|]$$

D12: The ***refinement of a graph coloring*** $\sigma = [C_1, \ldots, C_m]$ is the new coloring $R(\sigma)$ of G, derived by subpartitioning each color class C_j into subsets with the same degree vector $\vec{deg}_\sigma(v)$. For every $i \in [1, m]$, color the vertices of C_i using new colors $i_1, \ldots, i_{m_i}$ so that two vertices $x, y \in C_i$ have the same new color iff $\vec{deg}_\sigma(x) = \vec{deg}_\sigma(y)$.

D13: A coloring σ is ***stable***, if it has the same color classes as its refinement.

D14: For a stable coloring σ, define the ***color-class size vector***

$$s(\sigma) = [|C_1|, |C_2|, \ldots, |C_m|]$$

and the ***color-class adjacency matrix***

$$M(\sigma) = [m_{i,j}] = |N(x) \cap C_j|, \text{ where } x \in C_i$$

(For a stable coloring, these coefficients are independent of the choice of x.)

D15: The ***stabilization of a coloring*** σ is the coloring that results from iterating the refinement process until a stable coloring is obtained. It is denoted σ^*.

D16: Given two colored graphs G_1 and G_2, a mapping $\phi : V(G_1) \to V(G_2)$ is called ***color-preserving*** if every color class of G_1 is mapped to a single color class of G_2.

FACTS

F4: Let σ_1 and σ_2 be colorings of graphs G_1 and G_2, respectively, and let ϕ be a color-preserving isomorphism between G_1 and G_2. Then ϕ is a color-preserving mapping for the refinements $R(\sigma_1)$ and $R(\sigma_2)$. Furthermore,

$$\forall x \in V(G_1), \quad \vec{deg}_{\sigma_1}(x) = \vec{deg}_{\sigma_2}(\phi(x))$$

F5: Let σ_1 and σ_2 be colorings of graphs G_1 and G_2, respectively. If there is a color-preserving isomorphism between G_1 and G_2, then $s(\sigma_1^*) = s(\sigma_2^*)$ and $M(\sigma_1^*) = M(\sigma_2^*)$.

F6: Let τ_1 and τ_2 be the trivial colorings of graphs G_1 and G_2, respectively. Then G_1 and G_2 are isomorphic if and only if there is a color-preserving isomorphism with respect to the colorings τ_1^* and τ_2^*.

REMARK

R1: A generalization of the degree sequence of a graph has found applications in organic chemistry and drug design (see [Ran90]). For a given n-vertex graph G, the ***path degree sequence matrix*** or ***path layer matrix*** is the matrix

$$\lambda(G) = [\lambda_{ij}],\ i \in [1, n];\ j \in [1, n-1]$$

where λ_{ij} denotes the the number of paths of length j that start at the vertex v_i of G. In spite of initial hopes, the matrix does not identify graphs up to isomorphism. As Dobrynin and Mel'nikov [DoMe02] notice, "mathematical investigations of this matrix deal with finding a pair of non-isomorphic graphs having some specified properties and such that both graphs have the same path layer matrix."

Backtracking

The refinement technique is essential in applying the backtracking algorithmic strategy.

DEFINITIONS

D17: A ***node*** of a backtracking tree is labeled with quadruples $\langle G_1, \pi_1; G_2, \pi_2\rangle$, where π_1 and π_2 are stable colorings of the respective graphs G_1 and G_2, for which a color-preserving isomorphism is suspected.

D18: The ***root*** of the tree is labeled with $\langle G_1, \pi_1^0; G_2, \pi_2^0\rangle$, where π_1^0 and π_2^0 are stabilizations of the trivial colorings of the respective graphs.

D19: For a ***non-leaf node*** $\langle G_1, \pi_1; G_2, \pi_2\rangle$, a ***child-node*** is constructed by selecting a vertex, called ***a pivot-vertex***, in a non-singleton color class of π_1, and a vertex in the corresponding color class of π_2, then making these vertices corresponding singleton classes (*fixing the vertices*) of the new colorings. The stabilization of the new colorings completes the generation of a child-node. If m is the size of the selected color-class, then m children-nodes are created.

D20: Let C be a non-singleton color class of a coloring π, let $x \in C$, and let ρ be the stabilization of the coloring obtained from π by replacing C with two color classes: $\{x\}$ and $C - x$. We say that fixing x ***shatters*** C, if every color class C' of ρ which is a subset of C is either a singleton, or $1 < |C'| < |C|/2$.

FACTS

A remarkable feature of *ISO* is that backtracking together with refining is often efficient in establishing non-isomorphism or in constructing an isomorphism between two graphs. A beautiful example of the efficiency of the backtracking algorithm fused with refining is given by Miller.

F7: ([Mi78], [Mi79]) Isomorphism testing of two projective planes can be done in $O(n^{\log\log n})$ time.

The cornerstone of Miller's algorithm is a classical theorem of Bruck.

F8: ([Br55]) Let Φ be a proper sub-plane of a finite projective plane Ψ, and let the orders of Φ and Ψ be m and n, respectively. Then either $m^2 \leq n$, or $m^2 + m \leq n$.

The efficiency of the reduction step depends on how fixing a pivot-vertex and stabilizing the resulting coloring splits the color classes of the current coloring.

F9: If for every non-leaf node of the backtracking tree, fixing the pivot-vertex shatters every non-singleton color class, then the height of the backtracking tree is $O(\log n)$, where n is the vertex number of the graphs in question, and the running time of the algorithm is $O(n^{\log n})$.

The example of disconnected graphs show that for some graphs, fixing no pivot-vertex shatters all color classes. A more complicated series of examples is based on the notion of a *section* of a stable coloring (see [CoGo84], [Go83]).

2.2.3 Practical Graph Isomorphism

FACT

F10: Each of the following classes of graphs admits a polynomial algorithm for isomorphism testing:

1. rooted trees (by Edmonds – see [AHU74]);
2. planar graphs (by Hopcroft and Wong [HoWo74]; and by Fonten [Fo76]);
3. interval graphs (by Lueker and Booth [LuBo79]);
4. circular graph (by Hsu [Hs95]);
5. graphs with bounded genus (by Miller [Mi80], by Filotti and Mayer [FiMa80], and by Grohe [Gr00]);
6. graphs with bounded vertex degree (by Luks [Lu82]);
7. graphs whose adjacency matrices have bounded eigenvalue multiplicity (by Babai, Grigoryev, and Mount [BaGrMo82]); and
8. graphs of bounded tree-width (by Bodlaender [Bo88]).

REMARKS

R2: Efficient implementation of the Weisfeiler-Leman algorithm (a generalization of refining) can be found in [Ba97].

R3: Practical graph isomorphism is successfully provided by the program **nauty**, developed by Brendan D. McKay. Using **nauty**, one can compute sets of generators for automorphism groups of directed and undirected graphs; **nauty** can also be used for producing a canonical labeling of a graph.

R4: The program **nauty** is written in a portable subset of C and runs on a variety of Unix/Linux systems.

R5: The web-site of the program **nauty** with a complete documentation can be found at `http://cs.anu.edu.au/~bdm/nauty`.

R6: The background methodology is described in [Mc81], although the algorithm and the implementation were significantly improved subsequently. The program **nauty** implements the backtracking algorithmic strategy; the nodes of the backtracking tree virtually built by the program (except for very simple cases, for which the tree is explicitly generated) are stable colorings π of a given graph G, with the associated sets of generators of the automorphism groups of the colored graph.

R7: The size of the graphs that can be considered by the program **nauty** is $2^{15} - 1 = 32765$, unless *BIGNAUTY* is defined, in which case the absolute size of the graph order is $2^{24} - 1 = 16777213$. Well-designed data structures, efficient implementations, and a number of additional options fine-tune the performance of **nauty**, making this remarkable software system a highly useful tool for research and applications.

2.2.4 Group-Theoretic Approach

The group-theoretic approach (see [WeLe68], [We79], [Ba79], [FuHoLu80], [Lu82], [BaLu83]) reduces *ISO* to the color automorphism problem, *CAP*.

DEFINITIONS

D21: The ***color automorphism problem*** (abbr. *CAP*) is to find a set of generators for the subgroup of color-preserving permutations of a permutation group $\mathcal{P}$ acting on a colored set X.

FACTS

By itself the group-theoretic reformulation of *ISO* in Fact 11 does not yield efficient isomorphism testing. A fundamental discovery of [Ba79] and [Lu82] was that for many classes of graphs, $\mathcal{A}ut(G)$ is contained in a direct product of small groups. This led to a polynomial-time probabilistic algorithm for computing the automorphism group for colored graphs with bounded color classes, and to the remarkable polynomial-time isomorphism test by Luks [Lu82] for graphs with bounded degree. To give the flavor of Luks's technique, we outline his algorithm for 3-regular graphs.

F11: (Mathon [Ma79]) Given an n-vertex graph G, define X to be the set of all vertex pairs (x, y) in G, each colored red or blue depending on whether (x, y) is an edge. Then, *ISO* is equivalent to the problem of finding a set of generators of $\mathcal{A}ut(G)$, as a subgroup of the complete symmetric group Σ_n acting on the set X.

NOTATION: Let $\mathcal{A}ut_e(G)$ denote the subgroup of automorphisms of a graph G that fix edge e.

F12: (Luks[82]) If there is a polynomial algorithm that outputs a set of generators for the subgroup $\mathcal{A}ut_e(G)$ corresponding to an arbitrary edge of a connected 3-regular graph, then there is a polynomial algorithm that determines if there is an isomorphism between any two 3-regular graphs G_1 and G_2.

NOTATION: For a connected 3-regular graph G and for $e \in E(G)$, let G_r (for each $r > 0$) denote the *induced subgraph* on the vertices in the union of the set of paths of length r that contain e. (Then for each $r > 0$, $Aut_e(G_r)$ is the group of automorphisms of G_r that fix e.)

F13: There exists a number $r > 0$ such that $Aut_e(G_{r+1}) = Aut_e(G_r)$ and such that $\mathcal{A}ut_e(G_r) = \mathcal{A}ut_e(G)$.

NOTATION: We observe that for every number $i > 1$, the restriction of an automorphism $\sigma \in \mathcal{A}ut_e(G_{i+1})$ to the vertices of G_r is an automorphism in $\mathcal{A}ut_e(G_r)$. The restriction operator $\mathcal{A}ut_e(G_{r+1}) \to \mathcal{A}ut_e(G_r)$ is denoted by π_r.

F14: There is a polynomial algorithm that finds a set of generators for the kernel of the homomorphism π_r, i.e., the group of automorphisms in $Aut_e(G_{r+1})$ that fix the vertices of the graph G_r.

F15: There is a polynomial algorithm that finds a set of generators for the subgroup $\pi_r(\mathcal{A}ut_e(G_{r+1})$ of automorphisms in $\mathcal{A}ut_e(G_r)$ that can be extended to automorphisms in $\mathcal{A}ut_e(G_{r+1})$.

F16: [GHLSW87] There is an $O(n^3 \log n)$ time algorithm for *ISO* on degree-three graphs.

Interestingly, it was possible to extend this technique to develop fast, yet not subexponential, algorithms for the canonical numbering of graphs.

REMARKS

R8: The proofs of Fact 13 and Fact 14 are at the heart of Luks's technique.

R9: The isomorphism testing algorithm for 3-regular graphs given in Fact 16 is the fastest known at this time.

R10: A paper presented at STOC'83 ([BaLu83]) contains a brief description of an algorithm that constructs a canonical numbering in $O(\exp(n^{1/2+o(1)})$ time.

2.2.5 Complexity

Testing two graphs for isomorphism is not known to be NP-complete, nor to be polynomial. It follows from Ladner's theorem [La75] that if $P \neq NP$, then the class $NPI = NP - P$ of problems with intermediate complexity is not empty. *ISO* is a prime candidate for an intermediate computational status (see [GaJo79], [Kö93]).

DEFINITION

D22: An ***isomorphism-complete problem*** is a problem that is polynomially equivalent to *ISO*.

FACTS

The uncertain computational status of *ISO* prompted the search for isomorphism-complete problems (see [AHU74, Bs94, Bo78, CoCo78, Ma79, HaPlSt82, Lu81]).

F17: The list of isomorphism-complete problems includes the following:

1. bipartite graph isomorphism;
2. regular graph isomorphisms;
3. complement graph isomorphism (given two graphs, determine if one is isomorphism to the complement to the other);
4. computing the number of isomorphic mappings between two given graphs;

5. automorphism with restriction (given a graph G and a vertex v), determine if there is $\phi \in \mathcal{A}ut(G)$, for which $\phi(v) \neq v$);
6. automorphism orbits (determine the orbits of $\mathcal{A}ut(G)$);
7. automorphism generators (determine a set of generators for $\mathcal{A}ut(G)$);
8. determine if a set of n graphs of order $n-1$ represent a collection of $n-1$-subgraphs of a graph;
9. the rooted directed acyclic graphs isomorphism;
10. the term equality problem (given two terms with functions that are commutative, associative, or both, and commutative variables, determine if the two terms are equal).

F18: *ISO* $\in$ *NP*. (A deterministic polynomial procedure checks whether a given one-to-one mapping $\phi : V_1 \to V_2$ is an isomorphism.)

F19: (Klivans and van Melkebeek [KlMe02]) There is a sub-exponential proof for testing membership in *co-ISO*, unless the *polynomial-time hierarchy* collapses.

REMARKS

R11: One of the reasons that *ISO* is expected to be of intermediate complexity is that the structure of *co-ISO*, the Graph Non-isomorphism Problem, is different from that of the complements of problems known to be *NP*-complete. Fact 19 is regarded as important evidence for a special computational status of *ISO*.

R12: It is widely believed that the *polynomial-time hierarchy*, introduced by A. Meyer and L. Stockmeyer ([MeSt72], [St76]), does not collapse.

References

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[Ba79] L. Babai, “Monte-Carlo algorithms in graph isomorphism testing”, *manuscript*, 1979.

[BaErSe80] L. Babai, P. Erdös, and S. M. Selkov, Random graph isomorphism, *SIAM J. of Comp.* 9 (1980), 628–635.

[BaGrMo82] L. Babai, D. Y. Grigoryev, and D. M. Mount, Isomorphism of graphs with bounded multiplicity, *Proc. 14th Ann. ACM Symp. Theory of Comp.* (1982), 310–324.

[BaLu83] L. Babai and E. M. Luks, Canonical labeling of graphs, *Proc. 15th Ann. ACM Symp. Theory of Comp.* (1983), 171–183.

[Ba97] L. Babel, I. V. Chuvaeva, M. Klin, D. V. Pasechnik, Algebraic Combinatorics in Mathematical Chemistry. Methods and Algorithms. II. Program Implementation of the Weisfeiler-Leman Algorithm, TUM-M9701, Januar 97, Technische Universität München.

[Bs94] D. Basin, A term equality problem is equivalent to graph isomorphism, *Information Processing Letters* 51 (1994), 61–66.

[Bo88] H. L. Bodlaender, Polynomial algorithms for chromatic index and graph isomorphism on partial k-trees, *J. Algorithms* 11 (1990), 631–643.

[Bo78] K. S. Booth, Isomorphism testing for graphs, semi-groups and finite automata are polynomially equivalent, *SIAM J. Comp.* 7 (1978), 273–279.

[Br55] R.H. Bruck, Difference sets in a finite group, *Trans. of the Amer. Math. Soc.* 78 (1955), 464–481.

[CoCo78] M. J. Colbourn and C. J. Colbourn, Graph isomorphism and self-complementary graphs, *SIGACT News* 10 (1978), 25–29.

[CoBo81] M. J. Colbourn and K. S. Booth, Linear time automorphism algorithm for trees, interval graphs, and planar graphs, *SIAM J. Comp.* 10 (1981), 203–225.

[CoGo70] D. G. Corneil and C. C. Gotlieb, An efficient algorithm for graph isomorphism, *J. ACM* 17 (1970), 51–64.

[CoGo84] D. Corneil and M. Goldberg, A non-factorial algorithm for canonical numbering of a graph, *J. Algorithms* 5 (1984), 345–362.

[DeDaLo77] N. Deo, J. M. Davis, and R. E. Lord, A new algorithm for digraph isomorphism, *BIT* 17 (1977), 16–30.

[DoMe02] A. A. Dobrynin and L. S. Mel'nikov, Recent results in the theory of the path layer matrix, *Graph Theory Notes of New York* XLIII (2002), New York Acad. Sci., 48–56.

[FiMa80] I. S. Filotti and J. N. Mayer, A polynomial algorithm for determining the isomorphism of graphs of fixed genus, *Proc. 12th Ann. ACM Symp. Theory of Comp.* 17 (1980), 236–243.

[Fo76] M. Fontet, Linear algorithms for testing isomorphism of planar graphs, pp. 411–423 in *Proc. 3rd Coll. Automata, Languages and Programming* (S. Michaelson and R. Milner, Eds.), Edinburgh Univ. Press, 1976.

[FüScSp83] M. Fürer, W. Schnyder, and E. Specker, Normal forms for trivalent graphs and graphs of bounded valence, *Proc. 15th Ann. ACM Symp. Theory of Comp.* (1983), 161–170.

[FuHoLu80a] M. Furst, J. Hopcroft, and E. Luks, A sub-exponential algorithm for trivalent graph isomorphism, Proc. 11th Southeastern Conf. on Combinatorics, Graph Theory, and Comp. (1980), *Congressum Numerantium* 33 (1980), 36–41.

[FuHoLu80b] M. Furst, J. Hopcroft, and E. Luks, Polynomial time algorithms for permutation groups, *Proc. 21th IEEE Symp. on Foundations Comp. Sci.* (1980), 36–41.

[GHLSW87] Z. Galil, C. M. Hoffman, E. M. Luks, C. P. Schnorr, A. Weber, An $O(n^3 \log n)$ deterministic and $O(n^3)$ probabilistic isomorphism test for trivalent graphs, *J. ACM* 34 (1987), 513–531.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.

[Go83] M. Goldberg, A non-factorial algorithm for the graph isomorphism problem, *Discrete Applied Math.* 6 (1983), 229–236.

[Gr00] M. Grohe, Isomorphism testing for embeddable graphs through definability, *Proc. 32nd ACM Symp. Theory of Comp.* (2000), 229–236.

[GrYe98] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Ha82] F. Harary, M. Plantholt, and R. Statman, The graph isomorphism problem is polynomially equivalent to the legitimate deck problem for regular graphs, *Caribbean J. Math.* 1 (1982), 15–23.

[Ho74] J. E. Hopcroft and C. K. Wong, Linear-time algorithms for isomorphism of planar graphs, *Proc. 6th Ann. ACM Symp. Theory of Comp.* (1974), 172–184.

[Ho82] C. M. Hoffman, Group-Theoretic Algorithms and Graph Isomorphism, *Lectures Notes in Computer Science* 136, Springer, 1982.

[Hs95] W. Hsu, $\mathcal{O}$(MN) algorithms for the recognition and isomorphism problem on circular graphs, *SIAM J. Comp.* 24 (1995), 411–439.

[Kl95] M. Klin, Ch. Rücker, G. Rücker, G. Tinhofer, Algebraic Combinatorics in Mathematical Chemistry. Methods and Algorithms. I. Permutation Groups and Coherent (Cellular) Algebras. TUM-M9510, Dezember 95, Technische Universität München.

[KlMe02] A. Klivans and D. van Melkebeek, Graph nonisomorphism has subexponential size proof unless the polynomial-time hierarchy collapses, *SIAM J. Computing* 31 (2002), 1501–1526.

[Kö93] J. Köbler, U. Schöning, and J. Torán, *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser, 1993.

[Kr94] D. Kratsch and L. A. Hemaspaandra, On the complexity of graph reconstruction, *Math. Systems Theory* 27 (1994), 257–273.

[Kr02] I. Krasikov, A. Lev, and B. D. Thatte, Upper bounds on the automorphism group of a graph, *Discrete Math.* 256 (2002), 489–493.

[La75] R. E. Ladner, On the structure of polynomial time reducibility, *J. Assoc. Comput. Mach.* 22 (1975), 155–171.

[Le99] J. van Leeuwen, Graph Algorithms, Chapter 10 (pp. 525–631) in *Handbook of Theoretical Computer Science*, Elsevier, 1994.

[Lu81] A. Lubiw, Some NP-complete problems similar to graph isomorphism, *SIAM J. Computing* 10 (1981), 11–24.

[Lu79] G. S. Lueker and K. S. Booth, A linear time algorithm for deciding interval graph isomorphism, *J. ACM* 26 (1979), 183–195.

[Lu82] E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comput. System Sci.* 25 (1982), 42–65.

[Ma82] A. Mansfield, The relationship between the computational complexities of the legitimate deck and isomorphism problems, *Quart. J. Math. Oxford (2)* 33 (1982), 345–347.

[Ma79] R. Mathon, A note on the graph isomorphism counting problem, *Information Processing Letters* 8 (1979), 131–132.

[Mc81] B. D. McKay, Practical graph isomorphism, *Congressus Numerantium* 30 (1981), 45–87.

[MS72] A. Meyer and L. Stockmeyer, The equivalence problem for regular expressions with squaring requires exponential space, *Proc. of the 13th IEEE Symposium on Switching and Automata Theory* (1972), 125–129.

[Mi78], G. Miller, On the $n^{\log n}$ isomorphism technique: a preliminary report, *Proc. 10th Ann. ACM Symp. Theory of Computing* (1978), 51–58.

[Mi79], G. Miller, Graph isomorphism, general remarks, *J. Computer and System Sci.* (1979), 128–142.

[Mi80], G. Miller, Isomorphism testing for graphs of bounded genus, *Proc. 12th Ann. ACM Symp. Theory of Computing* 17 (1980), 225–235.

[Ran90], M. Randiíc, Design of molecules with desired properties, pp. 77–145 in *Concepts and Applications of Molecular Similarity* (M. A. Johnson and G. M. Maggiora, Eds.), Wiley, 1990.

[St76] L. Stockmeyer, The polynomial-time hierarchy, *Theoretical Computer Science* 3 (1976), 1–22.

[We68], B. Weisfeiler and A. A. Leman, Reduction of a graph to a canonical form and an algebra arising during this reduction, *Nauchno-Technicheskaya Informatsia (Ser. 2)* 9 (1968), 12–16.

[We79], B. Weisfeiler, (ed.) On construction and identification of graphs, *Lecture Notes in Mathematics* 558, Springer, 1976.

[Ul76] J. R. Ullmann, An algorithm for subgraph isomorphism, *J. ACM* 23 (1976), 31–42.

2.3 THE RECONSTRUCTION PROBLEM

Josef Lauri, University of Malta, Malta

2.3.1 Two Reconstruction Conjectures
2.3.2 Some Reconstructible Parameters and Classes
2.3.3 Reconstructing from a Partial Deck
2.3.4 Tutte's and Kocay's Results
2.3.5 Lovász's Method; Nash-Williams's Lemma
2.3.6 Digraphs
2.3.7 Illegitimate Decks
References

Introduction

In the first volume of the *Journal of Graph Theory* (1977), the journal editors characterized the Reconstruction Conjecture as "the foremost currently unsolved problem in Graph Theory". Although 25 years have passed, we are still nowhere near a complete solution of this problem. We hope here to provide a panoramic view of the present state of knowledge on the Reconstruction Problem.

Throughout this section, graphs are assumed to be simple.

2.3.1 Two Reconstruction Conjectures

Some classical problems in mathematics are of the following type: if the invariant S' is associated with the given structure S, does S' determine S uniquely? In graph theory we ask what knowledge, short of full incidence relations, is sufficient to determine the graph completely. We know, for instance, that neither the chromatic polynomial nor the spectrum are enough. The best known open problem of this type in graph theory is the Reconstruction Problem.

Decks and Edge-Decks

DEFINITIONS

D1: Let G be a graph on n vertices. For any vertex v of G, the ***vertex-deleted subgraph*** $G - v$ is obtained by removing v and all edges incident to v. For any edge e, the ***edge-deleted subgraph*** $G - e$, is obtained by deleting edge e.

D2: The ***deck*** of a graph G, denoted $\mathcal{D}(G)$, is the collection of all vertex-deleted subgraphs. The ***edge-deck***, denoted $\mathcal{ED}(G)$, is the collection of all edge-deleted subgraphs of G.

The graphs in the deck are unlabelled. If G contains isomorphic vertex-deleted subgraphs, then such subgraphs are repeated in $\mathcal{D}(G)$. The same holds for the edge-deck.

Therefore, the deck and the edge-deck are multisets, rather than sets, of isomorphism types of graphs.

NOTATION: If A is a subset of vertices of a graph G, then $G - A$ denotes the graph obtained from G by deleting all vertices in A and every edge incident to at least one of them. If B is a subset of edges of G, then $G - B$ denotes the subgraph of G obtained by deleting all the edges in B.

TERMINOLOGY NOTE: In some sections of the *Handbook*, the adjectives used to describe the subgraphs in Definition 1 are *vertex-deletion* and *edge-deletion*.

EXAMPLE

E1: Figure 2.3.1 shows an example of a graph and its deck.

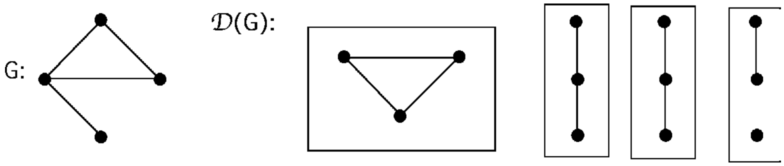


Figure 2.3.1 **A graph and its deck.**

Reconstructibility

Two important surveys [BoHe77], [Na78] give the early state of knowledge on the Reconstruction Problem, together with a complete bibliographic list. Since then, a number of survey or expository articles have been published [El88], [Ma88], [La87], [Bo91]. The monograph [LaSc03] contains four chapters on the reconstruction problem.

DEFINITIONS

D3: Any graph H with the same deck as G is called a ***reconstruction*** of G. If every reconstruction of G is isomorphic to G then G is said to be ***reconstructible***.

D4: An ***edge-reconstruction*** of G is a graph which has the same edge-deck as G, and the graph G is said to be ***edge-reconstructible*** if every edge-reconstruction of G is isomorphic to it.

CONJECTURES

Suppose that G and H are graphs such that $\mathcal{D}(H) = \mathcal{D}(G)$. Our concern is whether H must be isomorphic to G. We have a similar concern for the edge-deck invariant.

The Reconstruction Conjecture [Ke57, Ul60]: *Every graph with at least three vertices is reconstructible.*

The Edge-Reconstruction Conjecture [Ha64]: *Every graph on at least four edges is edge-reconstructible.*

In principle, given a deck $\mathcal{D}(G)$ of order n, one can consider all graphs on n vertices to check which graphs have the given deck. The reconstructibility question is one of uniqueness, that is, whether this search will find only one graph with the given deck.

EXAMPLES

E2: The complete graph K_2 not reconstructible, because the graph $2K_1$ consisting of two isolated vertices is a reconstruction of G, yet not isomorphic to G. According to the Reconstruction Conjecture, K_2 and $2K_1$ are the only nonreconstructible graphs.

E3: The graph $G = K_3 \cup kK_1$ is not edge-reconstructible because, if the graph $K_{1,3} \cup (k-1)K_1$ is an edge-reconstruction of $\mathcal{D}(G)$ that is not isomorphic to it. Also, $2K_2$ is not edge-reconstructible because if $H = P_3 \cup K_1$ (where P_3 is the path on three vertices), then $\mathcal{ED}(G) = \mathcal{ED}(H)$, but $G \not\cong H$. According to the Edge-Reconstruction Conjecture, these are the only graphs that are not edge-reconstructible.

Relationship between Reconstruction and Edge-Reconstruction

Intuition seems to suggest that it is easier to reconstruct a graph from its edge-deck than from its deck; there are generally more graphs in the edge-deck, and edge-deleted subgraphs are generally more nearly like the original graph than vertex-deleted subgraphs. Indeed, Fact 1 implies that if the Reconstruction Conjecture is true for graphs without isolated vertices, then so is the Edge-Reconstruction Conjecture. Fact 2 shows that the problem of edge-reconstruction is a special case of reconstruction.

FACTS

F1: (Greenwell's Theorem) [Gr71] Let G be a graph without isolated vertices. The deck of G is edge-reconstructible, that is, $\mathcal{D}(G)$ is uniquely determined from $\mathcal{ED}(G)$. Therefore, if G is reconstructible, then it is also edge-reconstructible.

F2: (Hemminger's Theorem) [He69] A graph is edge-reconstructible if and only if its line graph is reconstructible and is not K_3.

REMARKS

R1: We tacitly assume henceforth, unless otherwise stated, that any graph to be reconstructed has no isolated vertices.

R2: The Edge-Reconstruction Problem holds its own considerable independent interest, because several results known in edge-reconstruction have not yet been achieved for vertex-reconstruction, and some elegant proof techniques have been developed for edge-reconstruction. We consider these techniques in some detail in a later section.

Reconstruction and Graph Symmetries

At the heart of the difficulty of reconstructing a graph G from its deck is the symmetry of G and of the subgraphs in its deck. Suppose, for illustration, that the vertices of G are labelled $1, \ldots, n$ and that these labels are preserved on every vertex-deleted subgraph. Then, clearly, the graph G can be uniquely reconstructed from any three subgraphs. The labelings remove all symmetries of G and its subgraphs, and therefore all ambiguities of how these subgraphs are embedded inside G. A later subsection brings out more clearly the role of the automorphism of G in the edge-reconstruction of G.

DEFINITION

D5: Let k be a fixed but arbitrary integer. Then the graph G is said to have ***property*** A_k if, whenever A and B are distinct k-sets of vertices of G, the graphs $G-A$ and $G-B$

are not isomorphic. In other words, if G has n vertices, then any two subgraphs of G induced by different sets of $n-k$ vertices are not isomorphic.

FACTS

F3: If G has property A_{k+1}, then it has property A_k, and if it has property A_1, then its automorphism group is trivial.

F4: [Ko71, Mü76, Bo90] For a fixed k, almost every graph has property A_k, meaning that the proportion of labeled graphs on n vertices that have this property tends to 1 as n goes to ∞.

F5: [My88, Bo90] If a graph G has property A_3, then it can be reconstructed uniquely from any three subgraphs in its deck.

REMARK

R3: Fact 5 provides insight into the Reconstruction Problem. Let $G-u, G-v, G-w$ be any three subgraphs from the deck of G. Since G also has property A_2, we can identify v in $G-u$ and u in $G-v$ as the only vertices which give $G-u-v \simeq G-v-u$. Also, by property A_3, there is a unique isomorphism from $G-u-v$ to $G-v-u$. This isomorphism labels the two graphs uniquely, and we have the situation of a labeled graph which we described above. By comparing the two graphs $G-u, G-v$ we can then clearly put the vertex u back in $G-u$ and join it to its neighbors in G. The only uncertainty is whether u is adjacent to v. But this can be resolved by repeating the above with $G-u$ and $G-w$ instead of $G-v$.

2.3.2 Reconstructible Parameters and Classes

Reconstructible Parameters

DEFINITION

D6: A ***reconstructible (or edge-reconstructible) graph parameter*** is a parameter $\mathcal{P}$ such that, for any graph G with $\mathcal{P}=p$, any reconstruction (or edge-reconstruction) of G also has parameter value p.

NOTATION: Let H and G be two graphs. Then $\binom{G}{H}$ denotes the number of subgraphs of G that are isomorphic to H.

FACTS

Perhaps the single most useful result in reconstruction has proven to be *Kelly's Lemma*, Fact 11.

F6: The number of vertices and the number of edges are both reconstructible and edge-reconstructible (see [LaSc03]).

F7: The degree sequence is reconstructible and edge-reconstructible. (See [Bo91, LaSc03].)

F8: Given a graph $G-v$ from the deck of G, the degrees in G of the missing vertex v and of its neighbors are reconstructible (see [LaSc03]).

F9: For graphs without isolated vertices, a parameter is edge-reconstructible if it is reconstructible, by Greenwell's Theorem.

F10: Given a graph $G - e$ in the edge-deck of G, the degrees in G of the vertices with which the missing edge e is incident is edge-reconstructible.

F11: (Kelly's Lemma) [Ke57] Let G and H be graphs with G having more vertices than H. Then $\binom{G}{H}$ is reconstructible from $\mathcal{D}(G)$. Similarly, if G has at least as many vertices as H and strictly more edges than H, then $\binom{G}{H}$ is reconstructible from $\mathcal{ED}(G)$.

Reconstructible Classes

When we say that a class $\mathcal{C}$ of graphs is "reconstructible", one is only given the deck or the edge-deck of a graph, and not the information that the graph to be reconstructed is in $\mathcal{C}$. It must first be first determined from the deck alone that the graph is in $\mathcal{C}$, and then proved, by using this derived piece of information, that G is reconstructible. The following is a more exact definition of these two stages.

DEFINITIONS

D7: A class $\mathcal{C}$ of graphs is said to be ***recognizable*** (or ***edge-recognizable***) if for any graph $G \in \mathcal{C}$, any reconstruction (or edge-reconstruction) of G is also in $\mathcal{C}$. Equivalently, $\mathcal{C}$ is recognizable (or edge-recognizable) if it can be determined from $\mathcal{D}(G)$ (from $\mathcal{ED}(G)$) whether G is in $\mathcal{C}$.

D8: A graph $G \in \mathcal{C}$ is said to be ***weakly reconstructible*** (or ***weakly edge-reconstructible***) if any reconstruction (or edge-reconstruction) of G that is also in the class $\mathcal{C}$ is isomorphic to G. Equivalently, G is weakly reconstructible (or weakly edge-reconstructible) if, with the extra information that G is in $\mathcal{C}$, it can be determined uniquely from the deck (or edge-deck).

FACTS

The two-step process was essential in the proofs of reconstructibility of nearly all of the following classes. (Recall also that reconstructible graphs without isolated vertices are reconstructible are edge-reconstructible, by Greenwell's Theorem.)

F12: Regular graphs are reconstructible (see [LaSc03]).

F13: Disconnected graphs are reconstructible (see [LaSc03]).

F14: [Ke57] Trees are reconstructible.

F15: [Bo69b] Separable graphs (that is, graphs with connectivity 1) without vertices of degree 1 are reconstructible.

F16: [Ya88] The reconstruction conjecture is true if all 2-connected graphs are reconstructible.

F17: [Mc77] A computer search has shown that all graphs on nine or fewer vertices are reconstructible.

F18: [FiMa78, FiLa81, La81] Maximal planar graphs are reconstructible.

F19: [Gi76] Outerplanar graphs are reconstructible.

F20: [GoMc81] If all but at most one eigenvalue of G is simple and the corresponding eigenvectors are not orthogonal to the all-1's vector, then G is reconstructible. In particular, if G and its complement share no eigenvalue, then G is reconstructible.

F21: [Yu82] If there exists a subgraph $G - v$ of G none of whose eigenvectors is orthogonal to the all-1's vector, then G is reconstructible.

F22: [Fa94] Planar graphs with minimum degree at least 3 are edge-reconstructible.

F23: [Zh98a, Zh98b] Any graph of minimum degree 4 that triangulates a surface is edge-reconstructible. Any graph that triangulates a surface of characteristic at least 0 is edge-reconstructible. A graph G that triangulates a surface Σ of characteristic $\chi(\Sigma)$ is edge-reconstructible if $|V(G)| \geq -43\chi(\Sigma)$.

F24: [FaWuWa01] Series parallel networks (that is, 2-connected graphs without a subdivision of K_4) are edge-reconstructible.

F25: [Ch71] If a graph has property A_2 then it is reconstructible.

F26: [[ElPyXi88] Claw-free graphs are edge-reconstructible.

F27: [MyElHo87] Bidegreed graphs are edge-reconstructible.

REMARKS

R4: A ***claw-free*** graph is one that has no induced subgraph isomorphic to $K_{1,3}$. This result made essential use of Nash-Williams' Lemma, which we shall discuss below.

R5: ***Bidegreed graphs*** are graphs whose vertices can have only one of two possible degrees. (The degrees have to be consecutive numbers, else edge-reconstruction is trivial.) The next step would be edge-reconstruction of tridegreed graphs. (Again, if the three degrees are not consecutive, then edge-reconstruction is easy.) However, even for degrees 1, 2, and 3, this seems extremely difficult to tackle [Sc84].

R6: [BoHe77] suggested the problems of reconstructing and in edge-reconstructing bipartite graphs. No progress has been achieved to date.

R7: The proofs of several of the above results involved long arguments very specific to the class of graphs under consideration, although in some cases common techniques began to emerge. Their proofs use special properties of such classes, and often new properties have to be unearthed. §2.3.4 and §2.3.5 present more general results, that are less restricted to particular classes of graphs.

2.3.3 Reconstructing from a Partial Deck

The proofs of most of the results above use much less information than the full deck or edge-deck. In particular, trees have been shown to be reconstructible by deleting only their endvertices (vertices of degree 1) [HaPa66], or only their peripheral vertices (vertices at maximum distance from the center of the tree) [Bo69a].

Endvertex-Reconstruction

DEFINITION

D9: The ***endvertex-deck*** of a graph G is the collection of graphs $G - v$ for all vertices v with degree 1 in G. A graph G is endvertex-reconstructible if it is uniquely determined by its endvertex-deck.

FACTS

F28: [HaPa66] Trees are endvertex-reconstructible.

F29: [Br71] For every positive integer k, there exists a graph with k endvertices that is not endvertex-reconstructible.

A result in §2.3.5 indicates that the proportion of endvertices in a graph is what determines its endvertex-reconstructibility.

Reconstruction Numbers

Again noting that not all graphs in the deck are usually needed for reconstruction, Harary and Plantholt [HaPa85] introduced the definition of reconstruction numbers.

DEFINITIONS

D10: The ***reconstruction number*** of a graph G, denoted by $\mathrm{rn}(G)$, is the least number of subgraphs in the deck of G that guarantees that G is uniquely determined. The ***edge-reconstruction number***, denoted by $\mathrm{ern}(G)$, is analogously defined.

D11: Let $\mathcal{C}$ be a class of graphs. The ***class reconstruction number*** of a graph G in $\mathcal{C}$, denoted by $\mathcal{C}\mathrm{rn}(G)$, is the minimum number of subgraphs in the deck of G that, together with the information that G is in $\mathcal{C}$, guarantees that G is uniquely determined. The ***class edge-reconstruction number***, denoted by $\mathcal{C}\mathrm{ern}(G)$, is analogously defined.

FACTS

Facts 31 and 32 imply that there is no nonconnected graph with c vertices in each component and reconstruction number equal to $c + 1$. They also raise the natural question of investigating the gap between 3 and $c + 1$ for the reconstruction number of nonconnected graphs.

F30: [My88, Bo90] Almost every graph has reconstruction number equal to 3.

F31: [My90] A nonconnected graph with components not all isomorphic has reconstruction number 3. If all components are isomorphic and have c vertices each, then the reconstruction number can be equal to $c + 2$.

F32: [AsLa02] If the reconstruction number of a nonconnected graph is at least $c + 1$ then G must consist of copies of K_c.

F33: [BaBaHo87] If $\mathcal{C}$ is the class of total graphs and G is in $\mathcal{C}$, then $\mathcal{C}\mathrm{rn}(G)$ equals 1.

F34: [My90] The reconstruction number of trees is 3.

F35: [HaLa87] If $\mathcal{C}$ is the class of maximal planar graphs and G is maximal planar then $\mathcal{C}\mathrm{rn}(G)$ is at most 2. Those maximal planar graphs with class reconstruction number equal to 1 are characterized.

F36: Almost every graph has edge-reconstruction number equal to 2 (see [LaSc03]).

F37: [Mo95] Let G be a nonconnected graph. If G contains a pair of nontrivial, nonisomorphic components, then $\mathrm{ern}(G)$ is at most 3. If, furthermore, G is not a forest and contains a component other than K_3 and $K_{1,3}$, then $\mathrm{ern}(G)$ is at most 2. If the components of G are all isomorphic and contain k edges, then the edge-reconstruction number can be as high as $k+2$.

F38: [Mo93] Every tree T with at least 4 edges has $\mathrm{ern}(T) \leq 3$.

CONJECTURE

Harary and Lauri [HaLa88]: If $\mathcal{C}$ is the class of trees and T is a tree, then $\mathcal{C}\mathrm{rn}(T)$ is at most 2.

FURTHER REMARKS

R8: Although some results indicate that edge-reconstruction is implied by reconstruction, no such relationship seems to exist between these $\mathrm{rn}(G)$ and $\mathrm{ern}(G)$. In fact, the edge-reconstruction number for a graph could be greater than its reconstruction number.

R9: We have seen that lack of symmetry favors reconstruction, but highly symmetric graphs are regular, and thus trivially reconstructible. Reconstruction numbers seem to put this in a better perspective because, while graphs with property A_3 have reconstruction number 3, it seems [My88] that regular graphs are the candidates for being the graphs with the largest reconstruction number. The reconstruction number of regular graphs is not yet known.

R10: Myrvold [My88] calls the reconstruction number the *ally reconstruction number*, regarding it as the smallest number of graphs from the deck which a player **A** can give to an ally **B** such that the ally can determine the graph uniquely. She also defines the *adversary reconstruction number* to be one more than the largest number of subgraphs which **A** can give an adversary **B** such that **B** cannot determine the graph uniquely. Only partial results have been obtained on the adversary reconstruction number, which seems even more difficult to tackle than the (ally) reconstruction number.

Set Reconstruction

Harary [Ha64] suggested another way of reconstructing, by the Set Reconstructure Conjecture.

CONJECTURE

Set Reconstruction Conjecture: *Any graph G with $n \geq 4$ vertices can be reconstructed uniquely from its set of nonisomorphic subgraphs $G-v$.*

In other words, one is now only given *one* graph from each isomorphism class in the deck, and one does not know how many times each given graph appears in the deck.

DEFINITION

D12: A graph or a parameter that can be determined from the respective set of non-isomorphic subgraphs is said to be ***set reconstructible***.

FACTS

F39: [Ma76] The number of edges and the set of degrees of a graph is set reconstructible.

F40: [Ma76] For every graph in which no vertex of minimum degree lies on a triangle, the degree sequence is set reconstructible.

F41: [Ma76] The degree sequence of any graph with minimum degree at most 3 is set reconstructible.

F42: [Ma76] The connectivity of any graph is set reconstructible.

F43: [Ma76] Nonconnected graphs are set reconstructible.

F44: Separable graphs (that is, graphs with connectivity 1) without vertices of degree 1 are set reconstructible.

F45: [Ma70] Trees are set reconstructible.

F46: [Gi76] Outerplanar graphs are set reconstructible.

F47: [ArCo74] Unicyclic graphs (that is, graphs having only one cycle) are set reconstructible.

Set Edge-Reconstructibility

The idea of set reconstruction can also be applied to edge-reconstruction, that is, only one copy of each isomorphism type in the edge-deck is given. When a parameter or a class of graph is so reconstructible we say that it is ***set edge-reconstructible***. We highlight a few results in set edge-reconstructibility.

FACTS

F48: [Ma76] The degree sequence of a graph is set edge-reconstructible.

F49: [DeFaRa02] The degree sequence of a graph with at least four edges is uniquely determined by the set of degree sequences of its edge-deleted subgraphs with one well-described class of exceptions. Moreover, the multiset of the degree sequences of the edge-deleted subgraphs determines the degree sequence of the graph.

F50: [AnDiVe96] If a graph G with at least four edges has at most two non-isomorphic edge-deleted subgraphs, then G is set edge-reconstructible.

Reconstruction from the Characteristic Polynomial Deck

Schwenk [Sc79] proposed the problem of reconstruction from the characteristic polynomial of each subgraph in the deck, which we call the ***polynomial deck***. He suggested a weakening of the problem so that what is required is the reconstruction of the characteristic polynomial of G from its polynomial deck. This problem is still open, and we here limit ourselves to presenting four results from the few that have been obtained.

FACTS

F51: [Sc79] The characteristic polynomial of any graph is reconstructible up to a constant from the polynomial deck.

F52: If a subgraph in the deck of G has a characteristic polynomial with repeated roots, then the characteristic polynomial of G is reconstructible from its polynomial deck (see [LaSc03]).

F53: [CvLe98] The characteristic polynomial of a tree is reconstructible from its polynomial deck.

F54: [Sc] If a graph of order n has at least $n/3$ vertices of degree 1 then its characteristic polynomial is reconstructible from its polynomial deck.

Reconstructing from k-Vertex-Deleted Subgraphs

A k-vertex-deleted subgraph of G is a subgraph obtained from G by deleting k of its vertices and all edges incident to them. We shall have more to say about this mode of reconstruction in Section 5 when we consider k-edge-deleted subgraphs. Here we limit ourselves to one result.

FACT

F55: [Ta89] Let $k \geq 3$ be an integer. Then the degree sequences of all sufficiently large graphs are determined by their k-vertex-deleted subgraphs. In particular, this result is true for all graphs on at least $f(k)$ vertices, where $f(k)$ is a certain function that is asymptotic to ke.

2.3.4 Tutte's and Kocay's Results

If Kelly's Lemma held for all spanning subgraphs of G, then this would solve the reconstruction problem. Tutte [Tu79] showed how Kelly's Lemma can be extended to certain classes of spanning subgraphs, which has had very important consequences. In [Ko81], Kocay rederived Tutte's results with easier proofs. Kocay's method receives fuller exposition in [Bo91] and [LaSc03].

Kocay's Parameter

DEFINITION

D13: Let G be a graph and $\mathcal{F} = (F_1, F_2, \ldots, F_k)$ a sequence of graphs (different F_i could be isomorphic). A ***cover*** of G by $\mathcal{F}$ is a sequence $\mathcal{G} = (G_1, G_2, \ldots, G_k)$ of subgraphs of G (not necessarily distinct) such that: (i) $G_i \simeq F_i$, $i = 1, \ldots, k$ and (ii) $G = \cup_i G_i$. The number of covers of G by $\mathcal{F}$ is denoted by $c(\mathcal{F}, G)$.

FACTS [Tu79], [Ko81] (see also [Bo91], [LaSc02])

F56: Let G be a graph and let $\mathcal{F} = (F_1, F_2, \ldots, F_k)$ be a sequence of graphs with each $|V(F_i| < |V(G)|$. Let $\kappa(\mathcal{F}, G)$ be the parameter defined by

$$\sum_X c(\mathcal{F}, X)\binom{G}{X}$$

where the summation is taken over all isomorphism types X of graphs such that $|V(X)| = |V(G)|$. Then $\kappa(\mathcal{F}, G)$ is reconstructible.

The following results are then obtained by defining a suitable choice for the F_i.

F57: The number of 1-factors of G is reconstructible.

F58: The number of spanning trees of G is reconstructible.

F59: The number of Hamiltonian cycles of G is reconstructible.

F60: The number of 2-connected spanning subgraphs of G with a specified number of edges is reconstructible.

The Characteristic and the Chromatic Polynomials

DEFINITION

D14: An ***elementary graph*** is a graph in which every component is either an edge or a cycle.

NOTATION: For any graph X, $c(X)$ denotes the number of components of X and $s(X)$ the number of cycles.

FACTS

F61: [Sa64] (see also [Bi93], p.49) Let the characteristic polynomial of G be

$$\lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \ldots + a_n$$

Then each coefficient a_i is given by

$$a_i = \sum_X (-1)^{c(X)} 2^{s(X)} \binom{G}{X}$$

where the summation extends over all isomorphism types X of elementary graphs on i vertices.

F62: [Wh32] (see also [Bi93], p.77) Let the chromatic polynomial of G be

$$b_1 x + b_2 x^2 + \ldots + b_n x^n$$

Then each coefficient b_i is given by

$$b_i = \sum_X (-1)^{|E(X)|} \binom{G}{X}$$

where the summation extends over all isomorphism types X of graphs on n vertices and i components.

F63: [Tu79, Ko81] The characteristic polynomial is reconstructible. (See also [Bo91, LaSc02].)

F64: [Tu79, Ko81] The chromatic polynomial is reconstructible. (See [Bo91, LaSc02].)

2.3.5 Lovász's Method; Nash-Williams's Lemma

It is quite arguable that the deepest and most general results obtained in reconstruction are those described in this section. Lovász [Lo72] showed that if a graph has at least half the largest possible number of edges, then it is edge-reconstructible, making a surprising and elementary use of the inclusion-exclusion principle. Using the same method as Lovász, Müller [Mü77] obtained a stronger conclusion. In a subsequent survey paper (still applying the method introduced by Lovász), Nash-Williams [Na78] proved a lemma from which Lovász's and Müller's results follow. A more extended exposition can be found in [El88] or [Bo91] or [LaSc03].

The Nash-Williams Lemma

DEFINITIONS

D15: For simple graphs G and H, a ***monomorphism*** from G to H can be defined as a bijection of their vertex sets, such that the endpoints of every edge of G are mapped to the endpoints of an edge of H.

NOTATION: The number of such monomorphisms $G \to H$ is denoted by $[H]_G$.

D16: A ***monomorphism with forbidden edge-set*** X from G to H, where $X \subseteq E(G)$, is a bijection of their vertex sets such that the endpoints of every edge in $E(G)-X$ are mapped to the endpoints of an edge of H, but the endpoints of every edge in X are mapped to a nonadjacent pair of vertices in H.

NOTATION: The number of such monomorphisms $G \to H$ with forbidden edge-set X is denoted by $[H]_{G\backslash X}$.

REMARK

R11: Notice that $[H]_{G-X} \geq [H]_{G\backslash X}$, even though both count monomorphisms from $G - X$ to H. To be counted in $[H]_{G\backslash X}$, a monomorphism must map every edge of X to a non-edge of H. However, to be counted in $[H]_{G-X}$, it can map edges in X either to edges or non-edges of H.

FACTS

F65: [Lo72] Let G and H be graphs and $X \subseteq E(G)$. Then

$$[H]_{G\backslash X} = \sum_{Y \subset X} (-1)^{|Y|}[H]_{G-X+Y}$$

F66: *Nash-Williams's Lemma* [Na78] Let G and H be graphs and $X \subseteq E(G)$, and suppose that G and H have the same edge-deck. Then

$$[H]_G = |\mathrm{Aut}(G)| + (-1)^{|X|}([H]_{G\backslash X} - [G]_{G\backslash X})$$

(See [Bo91], [LaSc03].)

F67: (Corollary to Nash-Williams's Lemma) (See [Bo91], [LaSc03].) Suppose that G, H, and X are as in Nash-Williams's Lemma, and that $G \not\cong H$. Then,

(i) if $|X|$ is odd, then $[H]_{G\backslash X} > 0$;

(ii) if $|X|$ is even, then $[G]_{G\backslash X} > 0$;

F68: [Lo72] Let G be a graph such that $|E(G)| > \binom{n}{2}/2$. Then G is edge-reconstructible.

F69: [Mü77] Let G be a graph such that $2^{|E(G)|-1} > n!$. Then G is edge-reconstructible.

F70: [Py90] A Hamiltonian graph with a sufficiently large number of vertices is edge-reconstructible.

This last result is perhaps the most striking obtained by these methods.

Structures Other Than Graphs

To see the full generality of the methods above, we define a ***structure*** to be a triple (D, Γ, E) where D is a finite set, Γ is a group of permutations acting on D, and E is a subset of D. By edge-reconstruction of a structure, we mean that the subsets $E - x$ are given, up to 'translation' by the group Γ. The question is whether E can be reconstructed uniquely, again up to action by the group Γ.

In edge-reconstruction for graphs, D would be the set of all possible $\binom{n}{2}$ edges on n vertices, E the edges of the n-vertex graph to be reconstructed, and Γ the full symmetric group with its induced action on the unordered, distinct pairs of vertices.

DEFINITION

D17: Instead of removing one edge at a time, suppose that k edges at a time are removed. Let us call the resulting reconstruction problem ***k-edge-reconstruction***.

FACTS

We now present some results obtained by viewing edge-reconstruction in this more general setting. The first result, although a straightforward application of the Nash-Williams Lemma to structures, leads to asking what is the minimum proportion of endvertices required to guarantee endvertex-reconstructibility. An extended treatment of edge-reconstruction seen in this light appears in [Bo91, LaSc03].

F71: [LaSc03] Let H be a graph with minimum degree 2, and let G be obtained from H by adding k endvertices such that no two have a common neighbor. Then G is endvertex-reconstructible if either $k > V|H|/2$ or $2^{k-1} > \mathrm{Aut}(H)$.

F72: [AlCaKrRo89] Let (D, Γ, E) be a structure such that $2^{|E|-k} > |\Gamma|$. Then the structure is k-edge-reconstructible.

Radcliffe and Scott [RaSc98] consider the reconstruction of a subset of the cyclic group Z_n or of the reals, R, up to translation from the collection of its subsets of a given size, also up to translation.

F73: [RaSc98] Suppose that p is prime. Then every subset of Z_p is reconstructible from the collection of its 3-subsets.

F74: [RaSc98] For arbitrary n, almost all subsets of Z_n are reconstructible from the collections of their 3-subsets.

F75: [RaSc98] For any n, every subset of Z_n is reconstructible from its $9\alpha(n)$-subsets, where $\alpha(n)$ is the number of distinct prime factors of n.

F76: [Rasc99] A locally finite subset of R (that is, a subset which contains only finitely many translates of any given finite set of size at least 2) is reconstructible from its 3-subsets.

F77: [Ra02] Consider two such subsets of R^2 to be isomorphic if one can be transformed into the other by a translation or a rotation by a multiple of 90 degrees. Then every finite subset A of the plane R^2 is uniquely determined by at most 5 of its subsets of cardinality $|A|-1$, given up to isomorphism; that is, in the terminology of graph reconstruction, the subset A has reconstruction number 5.

The paper [AlCaKrRo89] should be studied carefully by anyone interested in extending the reconstruction of structures in the direction of k-edge-reconstruction.

The Reconstruction Index of Groups

Looking at reconstruction of structures (D, Γ, E) has led some authors to focus attention more directly on the permutation group Γ.

DEFINITION

D18: The ***reconstruction index*** $\rho(\Gamma, D)$ of the permutation group Γ acting on D is the smallest number t such that for any subset $E \subset D$ with $|E| \geq t$, the structure (D, Γ, E) is edge-reconstructible.

FACTS

F78: The Edge-Reconstruction Conjecture can be rephrased: if $Y = \{1, 2, \ldots, n\}$, if D is the set of unordered, distinct pairs of Y, and if $S_n^{(2)}$ is the symmetric group of Y acting on these pairs, then $\rho(S_n^{(2)}, D) = 4$.

F79: [Mn98] The reconstruction index of an abelian group is 4 and the reconstruction index of hamiltonian groups is 5.

For additional results on the reconstruction index of groups, see [Ca96, Ma96, Mn87, Mn92, Mn95].

2.3.6 Digraphs

The reconstruction conjecture for digraphs is false. Ramachandran has noted that all non-reconstructible digraphs discovered thus far would be reconstructible if, with every $D - v$, one is also given the in-degree and the out-degree in digraph D of the missing vertex v.

DEFINITION

D19: [Ra97] A digraph D is said to be ***N-reconstructible*** if it is uniquely determined by the triples $(D - v_i, \deg_{\text{in}}(v_i), \deg_{\text{out}}(v_i))$, for all vertices v_i of D.

FACTS

F80: [St77, Ko85] There exists an infinite family of tournaments that are not reconstructible.

F81: [HaPa67] Tournaments on at least five vertices that are not strongly connected are reconstructible. (See also [BoHe77].)

Some additional positive results on the reconstructibility of tournaments can be found in [DeGu90, Gu96, Vi99].

CONJECTURE

The N-Reconstruction Conjecture for Digraphs (Ramachandran): *Every digraph is N-reconstructible.*

2.3.7 Illegitimate Decks

DEFINITIONS

D20: A collection of graphs $G_1, G_2, \ldots, G_n$ each on $n - 1$ vertices is said to be an ***illegitimate deck*** if there is no graph G having the given collection as deck.

D21: The ***illegitimate deck problem*** is to determine whether such a given collection of graphs is indeed the deck of some graph.

FACTS

F82: [Ma82] Determining whether a given collection of graphs is an illegitimate deck is at least as hard as the isomorphism problem.

F83: [HaPlSt82] The graph isomorphism problem is polynomially equivalent to the illegitimate deck problem for regular graphs.

More information about the relationship between the computational complexities of the illegitimate deck problem and the graph isomorphism problem can be found in [KöScTo93, KrHe94].

References

[AlCaKrRo89] N. Alon, Y. Caro, I. Krasikov, and Y Roditty, Combinatorial reconstruction problems, *J. Combin. Theory (Ser. B)* 47 (1989), 153–161.

[AnDiVe96] L. D. Andersen, S. Ding, and P. D. Vestergaard, On the set edge-reconstruction conjecture, *J. Combin. Math. Combin. Comput.* 20 (1996), 3–9.

[ArCo74] E. Arjomandi and D.G. Corneil, Unicyclic graphs satisfy Harary's Conjecture, *Canad. Math. Bull.* 17 (1974), 593–596.

[AsLa02] K. J. Asciak and J. Lauri, On disconnected graphs with large reconstruction number, *Ars Combin.* 62 (2002), 173–181.

[BaBaHo87] D. W. Bange, A. E. Barkauskas, and L. H. Host, Class-reconstruction of total graphs, *J. Graph Theory* 11 (1987), 221–230.

[Bi93] N. L. Biggs, *Algebraic Graph Theory*, Cambridge University Press, 1993.

[Bo90] B. Bollabás, Almost every graph has reconstruction number 3, *J. Graph Theory* 14 (1990), 1–4.

[Bo69a] J. A. Bondy, On Kelly's congruence theorem for trees. *Proc. Camb. Phil. Soc.* 65 (1969), 387–397.

[Bo69b] J. A. Bondy, On Ulam's conjecture for separable graphs, *Pacific J. Math.* 31 (1969), 281–288.

[Bo91] J. A. Bondy, A graph reconstructor's manual, pages 221–252 in A. D. Keedwell (Ed.), *Surveys in Combinatorics*, Cambridge University Press, 1991.

[BoHe77] J. A. Bondy and R. L. Hemminger, Graph reconstruction — a survey, *J. Graph Theory* 1 (1977), 227–268.

[Br71] R. M. Bryant, On a conjecture concerning the reconstruction of graphs, *J. Combin. Theory* 11 (1971), 139–141.

[Ca96] P. J. Cameron, Stories from the age of reconstruction. *Congr. Num.* 113 (1996), 31–41.

[Ch71] P. Z. Chinn, A graph with p points and enough distinct $p-2$-order subgraphs is reconstructible, pages 71–73 in M. Capobianco, J. B. Frechen and M. Krolik (Eds.), *Recent Trends in Graph Theory*, Springer-Verlag, 1971.

[CvLe98] D. M. Cvetković and M. Lepović, Seeking counterexamples to the reconstruction conjecture for the characteristic polynomial of graphs and a positive result, *Bull. Cl. Sci. Math. Nat. Sci. Math.* 23 (1998), 91–100.

[DeFaRa02] C. Delorme, O. Favaron, and D. Rautenbach, On the reconstruction of the degree sequence, *Discrete Math.* 259 (2002), 293–300.

[DeGu90] D. C. Demaria and C. Guido, On the reconstruction of normal tournaments, *J. Combin. Inform. System Sci.* 15 (1990), 301–323.

[El88] M. N. Ellingham, Recent progress in edge reconstruction, *Congr. Numer.* 62 (1988), 3–20.

[ElPyXi88] M. N. Ellingham, L. Pyber and Y. Xingxing, Claw-free graphs are edge reconstructible, *J. Graph Theory* 12 (1988), 445–451.

[Fa94] H. Fan, Edge reconstruction of planar graphs with minimum degree at least three—IV, *Systems Sci. Math. Sci.* 7 (1994), 218–222.

[FaWuWa01] H. Fan, Y-L. Wu, and C. K. Wang, On fixed edges and the edge reconstruction of series parallel networks, *Graphs Combin.* 17(2) (2001), 213–225.

[FiMa78] S. Fiorini and B. Manvel, A theorem on planar graphs with an application to the reconstruction problem, II, *J. Combin. Inf. Sys. Sci.* 3(3) (1978), 200–216.

[FiLa81] S. Fiorini and J. Lauri, The reconstruction of maximal planar graphs, I: Recognition, *J. Combin. Theory (Ser. B)* 30 (1981), 188–195.

[Gi76] W. B. Giles, Point deletions of outerplanar blocks, *J. Combin. Theory (Ser. B)* 20 (1976), 103–116.

[GoMc81] C. D. Godsil and B. D. McKay, Spectral conditions for reconstructibility of a graph, *J. Combin. Theory (Ser. B)* 30 (1981), 285–289.

[Gr71] D. L. Greenwell, Reconstructing graphs, *Proc. Amer. Math. Soc.* 30 (1971), 431–433.

[Gu96] C. Guido, A larger class of reconstructible tournaments, *Discrete Math.* 152 (1996), 171–184.

[Ha64] F. Harary, On the reconstruction of a graph from a collection of subgraphs, pages 47–52 in M. Fiedler (Ed.), *Theory of Graphs and its Applications*, Proc. Symposium Smolenice 1963, Academic Press, 1964.

[HaLa87] F. Harary and J. Lauri, The class-reconstruction number of a maximal planar graph, *Graphs and Combinatorics* 3 (1987), 45–53.

[HaLa88] F. Harary and J. Lauri, On the class-reconstruction number of trees, *Quart. J. Math. Oxford (2)* 39 (1988), 47–60.

[HaPa66] F. Harary and E. M. Palmer, The reconstruction of a tree from its maximal subtrees, *Canad. J. Math.* 18 (1966), 803–810.

[HaPa67] F. Harary and E. M. Palmer, On the problem of reconstructing a tournament from subtournaments, *Monatsh. Math.* 71 (1967), 14–23.

[HaPl85] F. Harary and M. Plantholt, The graph reconstruction number, *J. Graph Theory* 9 (1985), 451–454.

[HaPlSt82] F. Harary, M. Plantholt, and R. Statman, The graph isomorphism problem is polynomially equivalent to the legitimate deck problem for regular graphs, *Caribbean J. Math.* 1(1) (1982), 15–23.

[He69] R. L. Hemminger, On reconstructing a graph, *Proc. Amer. Math. Soc.* 20 (1969), 185–187.

[Ke57] P. J. Kelly, A congruence theorem for trees, *Pacific J. Math.* 7 (1957), 961–968.

[KöScTo93] J. Köbler, U. Schöning, and J. Torán, *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser, 1993.

[Ko81] W. L. Kocay, On reconstructing spanning subgraphs, *Ars Combinatoria* 11 (1981), 301–313.

Ko85] W. L. Kocay, On Stockmeyer's non-reconstructible tournaments, *J. Graph Theory* 9 (1985), 473–476.

[Ko71] A. D. Korshunov, Number of nonisomorphic graphs in an n-point graph, *Math. Notes of the Acad. of Sciences of the USSR* 9 (1971), 155–160.

[KrLeTh02] I. Krasikov, A. Lev, and B.D. Thatte, Upper bounds on the automorphism group of a graph, *Discrete Math.* 256 (2002), 489–493.

[KrHe94] D. Kratsch and L. A. Hemaspaandra, On the complexity of graph reconstruction, *Math. Systems Theory* 27(3) (1994), 257–273.

[La81] J. Lauri, The reconstruction of maximal planar graphs II: Reconstruction, *J. Combin. Theory (Ser. B)* 30 (1981), 196–214.

[La87] J. Lauri, Graph reconstruction—some new techniques and new problems, *Ars Combinatoria (Ser B)* 24 (1987), 35–61.

[LaSc03] J. Lauri and R. Scapellato, *Topics in Graph Automophisms and Reconstruction*, Cambridge University Press, 2003.

[Lo72] L. Lovász, A note on the line reconstruction problem. *J. Combin. Theory (Ser. B)* 13 (1972), 309–310.

[Ma82] A. Mansfield, The relationship between the computational complexities of the legitimate deck and isomorphism problems, *Quart. J. Math. Oxford (2)* 33 (1982), 345–347.

[Ma70] B. Manvel, Reconstruction of trees, *Canad. J. Math.* 22 (1970), 55–60.

[Ma76] B. Manvel, On reconstructing graphs from their sets of subgraphs, *J. Combin. Theory (Ser. B)* 21 (1976), 156–165.

[Ma88] B. Manvel, Reconstruction of graphs: progress and prospects, *Congr. Numer.* 63 (1988), 177–187.

[Ma96] P. Maynard, *On Orbit Reconstruction Problems*, PhD thesis, University of East Anglia, Norwich, 1996.

[Mc77] B. D. McKay, Computer reconstruction of small graphs. *J. Graph Theory* 1 (1977), 281–283.

[Mn87] V. B. Mnukhin, Reconstruction of k-orbits of a permutation group, *Math. Notes* 42 (1987), 975–980.

[Mn92] V. B. Mnukhin, The k-orbit reconstruction and the orbit algebra, *Acta. Applic. Math.* 29 (1992), 83–117.

[Mn95] V. B. Mnukhin, The reconstruction of oriented necklaces, *J. Combin., Inf. & Sys. Sciences* 20 (1995), 261–272.

[Mn98] V. B. Mnukhin, The k-orbit reconstruction for abelian and hamiltonian groups, *Acta. Applic. Math.* 52 (1998), 149–162.

[Mo93] R. Molina, The edge reconstruction number of a tree, *Vishwa Int. J. Graph Theory* 2(2) (1993), 117–130.

[Mo95] R. Molina, The edge reconstruction number of a disconnected graph, *J. Graph Theory* 19(3) (1995), 375–384.

[Mü76] V. Müller, Probabilistic reconstruction from subgraphs, *Comment. Math. Univ. Carolinae* 17 (1976), 709–719.

[Mü77] V. Müller, The edge reconstruction hypothesis is true for graphs with more than $n \log_2 nedges$, *J. Combin. Theory (Ser. B)* 22 (1977), 281–283.

[My88] W. J. Myrvold, *Ally and Adversary Reconstruction Problems*, PhD thesis, University of Waterloo, 1988.

[My89] W. J. Myrvold, The ally-reconstruction number of a disconnected graph, *Ars Combinatoria* 28 (1989), 123–127.

[My90] W. J. Myrvold, The ally-reconstruction number of a tree with five or more vertices is three, *J. Graph Theory* 14 (1990), 149–166.

[MyElHo87] W. J. Myrvold, M. N. Ellingham and D. G. Hoffman, Bidegreed graphs are edge reconstructible, *J. Graph Theory* 11(3) (1987), 281–302.

[Na78] C. St. J. A. Nash-Williams, The reconstruction problem, Chapter 8 in L. W. Beineke and R. J. Wilson (Eds.), *Selected Topics in Graph Theory*, Academic Press, London, 1978.

[Py90] L. Pyber, The edge-reconstruction of hamiltonian graphs, *J. Graph Theory* 14 (1990), 173–179.

[RaSc98] A. J. Radcliff and A. D. Scott, Reconstructing subsets of Z_n, *J. Combin. Theory (Ser. A)* 83(2) (1998), 169–187.

[RaSc99] A. J. Radcliff and A. D. Scott, Reconstructing subsets of reals, Research Paper 20 in *Electronic J. Combin.* 1 (1999), 7pp.

[Ra97] S. Ramachandran, N-reconstructibility of nonreconstructible tournaments. *Graph Theory Notes of N.Y.* 32 (1997), 23–29.

[Ra02] D. Rautenbach, On a reconstruction problem of Harary and Manvel, *J. Combin. Theory (Ser. A)* 99 (2002), 32–39.

[Sa64] H. Sachs, Beziehungen zwischen den in einem Graphen enthaltenen Kreisen und seinem charakteristischen Polynom, *Publ. Math. Debrecen* 11 (1964), 119-134.

[Sc85] J. Schönheim, personal communication.

[Sc79] A. J. Schwenk, Spectral reconstruction problems, *Topics in Graph Theory*, Annals New York Academy of Science 328 (1979), 183–189.

[Sc] I. Sciriha, Polynomial reconstruction and terminal vertices, *Linear Algebra and its Applications*, 356 (2002) 145–156.

[St77] P. K. Stockmeyer, The falsity of the reconstruction conjecture for tournaments, *J. Graph Theory* 1 (1977), 19–25.

[Ta89] R. Taylor, Reconstructing degree sequences from k-vertex-deleted subgraphs, *Discrete Maths.* 79 (1989/90), 207–213.

[Tu79] W. T. Tutte, All the king's horses — a guide to reconstruction in J. A. Bondy and U. S. R. Murty (Eds.), *Graph Theory and Related Topics*, Academic Press, 1979.

[Ul60] S. M. Ulam, *A Collection of Mathematical Problems*, Wiley (Interscience), New York, 1960.

[Vi99] P. Vitolo, The reconstruction of simply disconnected tournaments, *J. Combin. Inform. System Sci.* 24 (1999), 65–77.

[Wh32] H. Whitney, A logical expansion in mathematics, *Bull. Amer. Math. Soc.* 38 (1932), 572–579.

[Yu82] H. Yuan, An eigenvector condition for reconstructibility, *J. Combin. Theory (Ser. B)* 32 (1982), 245–256.

[Zh98a] Y. Zhao, On the edge reconstruction of graphs embedded in surfaces. II, *J. London Math. Soc.* 57 (1998), 268–274.

[Zh98b] Y. Zhao, On the edge reconstruction of graphs embedded in surfaces. III, *J. Combin. Theory (Ser. B)* 74 (1998), 302–310.

[Zh88] Yang Yong Zhi, The reconstruction conjecture is true if all 2-connected graphs are isomorphic, *J. Graph Theory* 12 (1988), 237–243.

2.4 RECURSIVELY CONSTRUCTED GRAPHS

R. B. Borie, University of Alabama
R. Gary Parker, Georgia Institute of Technology
C. A. Tovey, Georgia Institute of Technology

2.4.1 Some Parameterized Families of Graph Classes
2.4.2 Equivalences and Characterizations
2.4.3 Recognition
References

Introduction

The core idea of recursively constructed graphs is captured in Definition 1, but the substantial literature on the subject has motivated a considerable breadth and variety of notational distinctions.

NOTATION: All graphs in this section are simple, and an edge with endpoints x and y is denoted (x, y).

DEFINITIONS

D1: A ***recursively constructed graph class*** is defined by a set (usually finite) of primitive or ***base graphs***, in addition to one or more operations that compose larger graphs from smaller subgraphs. Each operation involves either fusing specific vertices from each subgraph or adding new edges between specific vertices from each subgraph.

D2: Each graph in a recursive class has a corresponding ***decomposition tree*** that shows how to build it from base graphs.

REMARK

R1: Graphs in these classes possess a modular structure, so fast algorithms can often be designed to solve hard problems restricted to these classes. The algorithms typically proceed by solving the desired problem on the base graphs, then employ dynamic programming to combine solutions for small subgraphs into a solution for a larger graph. The construction of these algorithms is the subject of Section 10.4.

2.4.1 Some Parameterized Families of Graph Classes

Trees

DEFINITION

D3: The graph with a single vertex r (and no edges) is a ***tree*** with root r (the sole base graph). Let (G, r) denote a tree with root r. Then $(G_1, r_1) \oplus (G_2, r_2)$ is a tree formed by taking the disjoint union of G_1 and G_2 and adding an edge (r_1, r_2). The root of this new tree is $r = r_1$.

TERMINOLOGY NOTE: Technically, the pairs (G, r) in Definition 3 denote *rooted trees*. However, the specification of distinguished vertices r_1 and r_2 (and hence r) is relevant here only as a vehicle in the recursive construction.

EXAMPLE

E1: Figure 2.4.1 illustrates the recursive construction of trees.

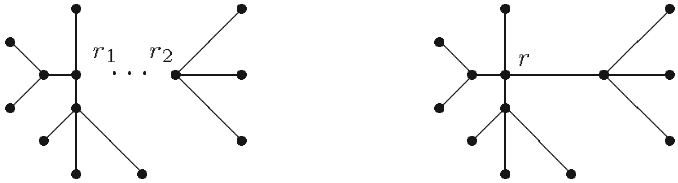


Figure 2.4.1 **Recursive construction of a tree.**

Series-Parallel Graphs

From a non-recursive perspective, a graph is series-parallel if it has no subgraph homeomorphic to K_4 [Du65]. The graph on the left of Figure 2.4.2 is not series-parallel; the offending subgraph is identified by bold edges. Removal of two edges, as indicated, yields the graph to the right which is series-parallel.

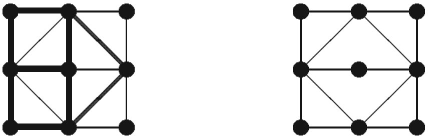


Figure 2.4.2 **Non-series-parallel and series-parallel graphs.**

Following, we give a recursive definition of this class.

DEFINITION

D4: A ***series-parallel graph*** with distinguished ***terminals*** l and r is denoted (G, l, r) and is defined recursively as follows:

- The graph consisting of a single edge (v_1, v_2) is a series-parallel graph (G, l, r) with $l = v_1$ and $r = v_2$.
- A ***series operation*** $(G_1, l_1, r_1) \odot_s (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2. The terminals of the new graph are l_1 and r_2.
- A ***parallel operation*** $(G_1, l_1, r_1) \odot_p (G_2, l_2, r_2)$ forms a series-parallel graph by identifying l_1 with l_2 and r_1 with r_2. The terminals of the new graph are l_1 and r_1.
- A ***jackknife operation*** $(G_1, l_1, r_1) \odot_j (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2; the new terminals are l_1 and r_1.

COMPUTATIONAL NOTE: The jackknife operation can also be specified where the new terminals, after composition, are defined to be l_1 and l_2.

EXAMPLE

E2: The three operations defining series-parallel graphs are demonstrated in Figure 2.4.3. The pair-specific composition is on the left; the result is shown to the right. Terminal vertices are circled and labeled.

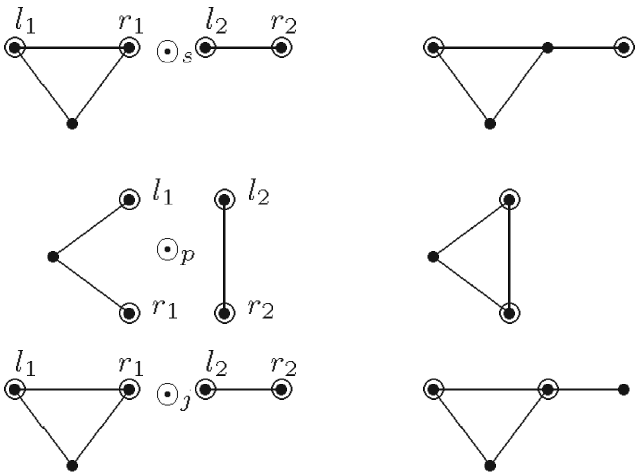


Figure 2.4.3 Composition operations for series-parallel graphs.

k-Trees and Partial k-Trees

DEFINITIONS

D5: The k-vertex complete graph, K_k, is a ***k*-tree**. A k-tree with $n+1$ vertices ($n \geq k$) is constructed from a k-tree on n vertices by adding a vertex adjacent to all vertices of one of its K_k subgraphs, and only to those vertices.

D6: A ***partial k-tree*** is a subgraph of a k-tree.

TERMINOLOGY: In a given construction of a k-tree, the original K_k subgraph is referred to as its ***basis***.

D7: A graph is ***chordal*** (or ***triangulated***) if it contains no induced cycles of length greater than 3.

D8: A graph is ***perfect*** if every induced subgraph has chromatic number equal to the size of its maximum clique.

FACTS

F1: Trees are 1-trees, and forests are partial 1-trees.

F2: Series-parallel graphs are partial 2-trees.

F3: Any K_k subgraph of a k-tree can act as its basis.

F4: All k-trees are chordal graphs and, hence, perfect (because every chordal graph is perfect).

EXAMPLES

E3: A 3-tree is shown on the left in Figure 2.4.4, and a partial 3-tree is shown to the right.

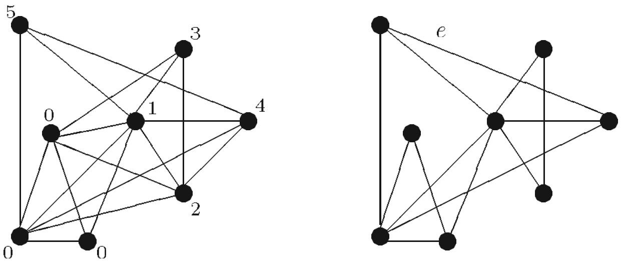


Figure 2.4.4 **Construction of a 3-tree and a partial 3-tree.**

Demonstrated by the graph to the left in Figure 2.4.4 above is the "creation" of a 3-tree following a small number of composition operations starting from the basis given by an initial K_3 identified by vertex labels of 0. At each step, a new (consecutively labeled) vertex is added. Observe that if edge e is eliminated from the graph on the right in Figure 2.4.4, a partial 2-tree is created.

E4: The graph on the left in Figure 2.4.5 below is series-parallel; it is a subgraph (and hence a partial 2-tree) of the 2-tree on the right. The dotted edges complete the 2-tree where the construction is verified by the labels on the vertices that are interpreted just as in the first note after Example 3.

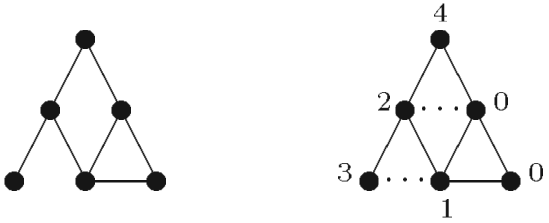


Figure 2.4.5 **A series-parallel graph and a 2-tree.**

Halin Graphs

DEFINITION

D9: A ***Halin graph*** is a planar graph having the property that its edge set E can be partitioned as $E = \langle T, C\rangle$, where T is a tree with no vertex of degree 2 and C is a cycle including only and all leaves of T.

FACTS

F5: Halin graphs are contained in the class of partial-3 trees.

F6: The set of Halin graphs is not closed under the taking of subgraphs, i.e., some subgraphs of Halin graphs are not Halin graphs.

EXAMPLES

E5: A Halin graph is given in Figure 2.4.6 below, with the cycle edges drawn on the outer face; their removal leaves a tree satisfying the stated degree stipulation.

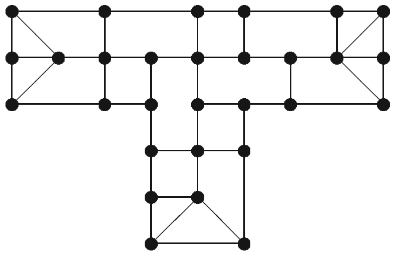


Figure 2.4.6 A Halin graph.

E6: The graph G in Figure 2.4.7 below is a 3-tree; vertex labels guide the construction as before. The subgraph given in bold is a tree of an underlying Halin graph; the cycle edges from G can be easily traced through the leaves of the specified tree.

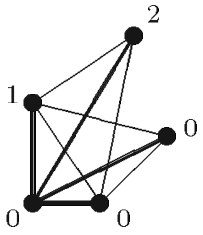


Figure 2.4.7 Another Halin graph.

Bandwidth-k Graphs

DEFINITION

D10: A graph $G(V, E)$ is a ***bandwidth-k graph*** if there exists a vertex labeling $h : V \to \{1, 2, \ldots |V|\}$ such that $\{u, v\} \in E \Rightarrow |h(u) - h(v)| \leq k$. (Bandwidth is discussed in §9.4.)

EXAMPLE

E7: A bandwidth-3 graph is shown to the left in Figure 2.4.8; displayed to the right is a bandwidth-2 graph.

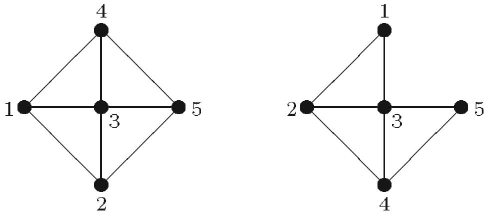


Figure 2.4.8 Bandwidth-3 and bandwidth-2 graphs.

Treewidth-k Graphs

The seminal works by Robertson and Seymour (*cf.*, [RoSe86-a], [RoSe86-b], [RoSe91-a]) are commonly identified as being responsible for motivating the creation of the graph classes in this section. Most notable is the concept of treewidth, which played a key role in the authors' work on graph minors culminating, ultimately, in the proof of Wagner's conjecture, a topic addressed in §2.4.2.

DEFINITIONS

D11: A ***tree-decomposition*** of a graph $G = (V, E)$ is a pair $(\{X_i \mid i \in I\}, T)$, where $\{X_i \mid i \in I\}$ is a family of subsets of V and T is a tree with vertex set I such that

- $\bigcup_{i \in I} X_i = V$
- for all edges $(x, y) \in E$ there is an element $i \in I$ with $x, y \in X_i$
- for all triples $i, j, k \in I$ if j is on the path from i to k in T, then $X_i \bigcap X_k \subseteq X_j$.

D12: The ***width*** of a given tree-decomposition is measured as $\max_{i \in I}\{|X_i| - 1\}$.

D13: The ***treewidth*** of a graph G is the minimum width taken over all tree-decompositions of G.

D14: A graph G is a ***treewidth-k graph*** if it has treewidth no greater than k.

REMARK

R2: Trivially, every graph, G, has a tree-decomposition that is defined by a single vertex (representing G itself). On the other hand, we are interested in tree-decompositions and, hence, their graphs, in which the X_i are small (i.e., graphs with small treewidth).

EXAMPLE

E8: A sample tree-decomposition is shown in Figure 2.4.9. For the stated graph, G, one family of suitable vertex sets can be given by: $X_1 = \{v_1, v_2, v_3\}, X_2 = \{v_2, v_7, v_8\}$, $X_3 = \{v_2, v_3, v_7\}, X_4 = \{v_3, v_5, v_7\}, X_5 = \{v_3, v_4, v_5\}$, and $X_6 = \{v_5, v_6, v_7\}$. An appropriate tree T is shown next and then on the right side of Figure 2.4.9, the relevant subgraphs of G induced by the stated pair $(\{X_i\}, T)$ are displayed. Moreover, the graph G has treewidth 2; in fact, the graph is series-parallel.

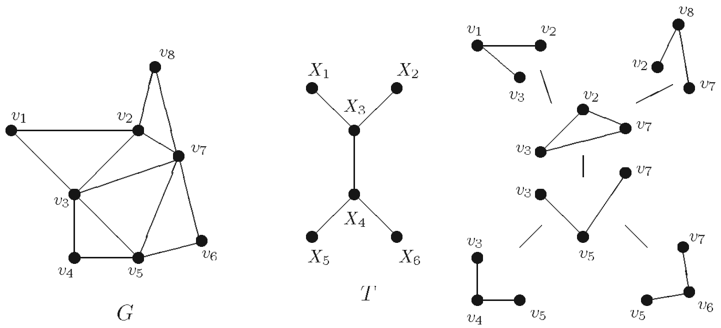


Figure 2.4.9 **A sample tree-decomposition.**

Pathwidth-k Graphs

DEFINITIONS

D15: A ***path-decomposition*** is a tree-decomposition whose tree is a path.

NOTATION: A path-decomposition is often denoted simply by a sequence of vertex subsets of V, say $\{X_1, X_2, \ldots, X_t\}$, listed in order defined by their position on the path.

D16: The ***width*** of a path-decomposition is $\max_{1 \le i \le t}\{|X_i| - 1\}$.

D17: The ***pathwidth*** of a graph G is the smallest width taken over all path-decompositions of G.

D18: A ***pathwidth-k graph*** is a graph that has pathwidth no greater than k.

EXAMPLE

E9: A sample path-decomposition is shown in Figure 2.4.10. The vertex-sets X_i and the first edge occurrences are displayed below the corresponding vertices of T.

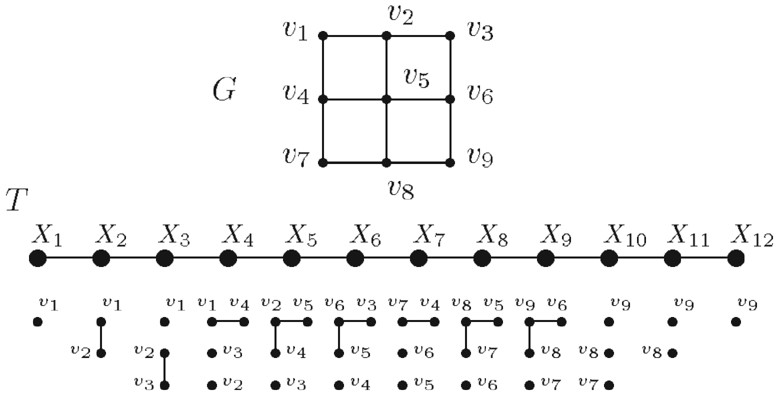


Figure 2.4.10 **A path-decomposition.**

Branchwidth-k Graphs

DEFINITIONS

D19: A ***branch-decomposition*** of a graph $G = (V, E)$ is a pair (T, f), where T is a tree in which every non-leaf vertex has exactly three neighbors and f is a bijection from the leaves of T to E.

D20: If the degree of every non-leaf vertex in T is *at least* 3, the pair (T, f) is called a ***partial branch-decomposition***.

D21: Let (T, f) be a branch decomposition of a graph $G = (V, E)$. The ***order*** of an edge e of T is the number of vertices v in V such that there exist leaves l_1 and l_2 of T residing in different components of $T - e$, where $f(l_1)$ and $f(l_2)$ are both incident on v.

D22: The ***width*** of a branch decomposition (T, f) is the maximum order of the edges of T.

D23: The ***branchwidth*** of G is the minimum width taken over all branch-decompositions of G.

D24: A graph G is a ***branchwidth-k graph*** if it has branchwidth no greater than k.

FACTS

F7: [RoSe91-a] A graph G is branchwidth-0 if and only if every component of G has at most one edge.

F8: [RoSe91-a] A graph G is branchwidth-1 if and only if every component of G has no more than one vertex with degree greater than or equal to 2.

F9: [RoSe91-a] A graph G is branchwidth-2 if and only if G has treewidth no greater than 2.

EXAMPLE

E10: A branchwidth-2 graph is shown on the left in Figure 2.4.11 (edges are numbered); its branch-decomposition is given to the right.

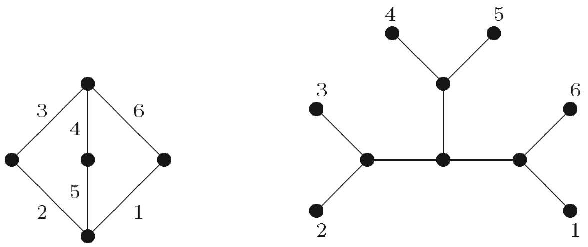


Figure 2.4.11 **Branchwidth-2 graph and its branch-decomposition.**

k-Terminal Graphs

DEFINITIONS

D25: A ***k-terminal graph*** $G = (V, T, E)$ has a vertex set V, an edge set E, and a set of *distinguished terminals* $T = \{t_1, t_2, \ldots, t_{|T|}\} \subseteq V$, where $|T| \leq k$.

D26: A ***k-terminal recursively structured class*** $C(B, R)$ is specified by a set B of base graphs and a finite rule set $R = \{f_1, f_2, \ldots, f_n\}$, where each f_i is a *recursive composition operation.*

EXAMPLE

E11: A construction for a 2-terminal graph is shown in Figure 2.4.12. Vertices are labeled in order to clarify how constituent subgraphs compose; terminals are denoted by doubly circled vertices.

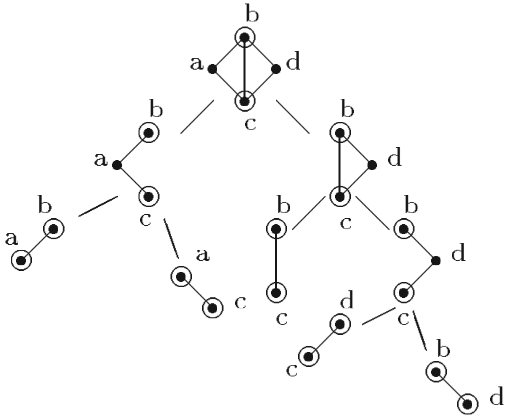


Figure 2.4.12 **Recursive construction of a 2-terminal graph.**

REMARKS

R3: Typically, for some k, B is the set of connected k-terminal graphs (V, T, E) with $V = T$. But each such base graph is trivially composed of individual edges, so it is reasonable and hence convenient to simply use $C(R)$ to denote $C(B, R)$, where B is a singleton consisting only of edges (*i.e.*, K_2).

R4: The notion of ***composition*** typically permitted in the context of k-terminal graphs, can be described in a more formal way. For $1 \leq i \leq m$, let $G_i = (V_i, T_i, E_i)$, such that $V_1, \ldots, V_m$ are mutually disjoint vertex sets. Let $G = (V, T, E)$ as well. Then a ***valid vertex mapping*** is a function $f : \cup_{1 \leq i \leq m} V_i \to V$ such that

- vertices from the same G_i remain distinct:
$$v_1 \in V_i, v_2 \in V_i, f(v_1) = f(v_2) \Rightarrow v_1 = v_2$$
- only (not necessarily all) terminals map to terminals:
$$v \in V_i, f(v) \in T \Rightarrow v \in T_i$$
- only terminals can merge:
$$v_1 \in V_{i_1}, v_2 \in V_{i_2}, i_1 \neq i_2, f(v_1) = f(v_2) \Rightarrow v_1 \in T_{i_1}, v_2 \in T_{i_2}$$
- edges are preserved:
$$(\exists i)(\{v_1, v_2\} \in E_i) \Leftrightarrow \{f(v_1), f(v_2)\} \in E$$

NOTATION: If f is a valid vertex mapping, the corresponding m-ary composition operation (denoted by f) is generally written $f(G_1, \ldots, G_m) = G$.

Cographs

DEFINITION

D27: A ***cograph*** is defined recursively as follows:

- A graph with a single vertex is a cograph.
- If G_1 and G_2 are cographs, then the disjoint union $G_1 \cup G_2$ is a cograph.
- If G_1 and G_2 are cographs, then the cross-product $G_1 \times G_2$ is a cograph, which is formed by taking the union of G_1 and G_2 and adding all edges (v_1, v_2) where v_1 is in G_1 and v_2 is in G_2.

TERMINOLOGY: Cographs are also referred to as ***complement reducible graphs***.

EXAMPLE

E12: A cograph construction is demonstrated in Figure 2.4.13. The relevant operations are signified at each node of the decomposition tree (left) for the graph G shown on the right.

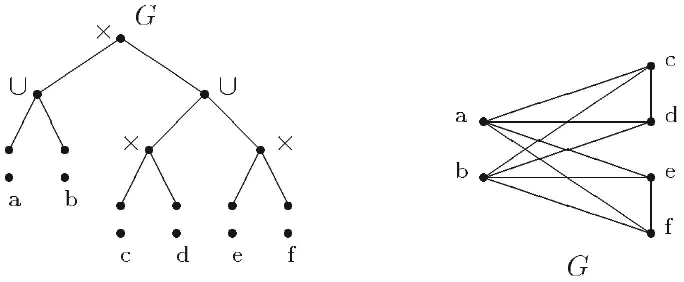


Figure 2.4.13 **Cograph construction.**

FACTS

F10: [CoLeBu81] The complement of any cograph is also a cograph.

F11: [CoLeBu81] All cographs are perfect.

Cliquewidth-k Graphs

The graph parameter *cliquewidth* was introduced in [CoEnRo93] and formed a seminal concept in linking research in graph theory and logic.

DEFINITION

D28: Let [k] denote the set of integers $\{1, 2, \ldots, k\}$. A ***cliquewidth-k graph*** is defined recursively as follows:

- Any graph G with $V(G) = \{v\}$ and $l(v) \in [k]$ is a cliquewidth-k graph.
- If G_1 and G_2 are cliquewidth-k graphs and $i, j \in [k]$, then
  (1) the disjoint union $G_1 \cup G_2$ is a cliquewidth-k graph.
  (2) the graph $(G_1)_{i \times j}$ is a cliquewidth-k graph, where $(G_1)_{i \times j}$ is formed from G_1 by adding all edges (v_1, v_2) such that $l(v_1) = i$ and $l(v_2) = j$.
  (3) the graph $(G_1)_{i \to j}$ is a cliquewidth-k graph, where $(G_1)_{i \to j}$ is formed from G_1 by switching all vertices with label i to label j.

REMARK

R5: Definition 28 defines the *class* of cliquewidth-k graphs. The cliquewidth of a graph G is the smallest value of k such that G is a cliquewidth-k graph. A cliquewidth decomposition for a graph is a rooted tree such that the root corresponds to G, each leaf corresponds to a labeled, one-vertex graph, and each non-leaf node of the tree is obtained by applying one of the operations $\cup$, $i \times j$, or $i \to j$ to its child or children.

TERMINOLOGY NOTE: In this section, the term *clique* refers to *any* complete subgraph of the graph. In some other sections of this handbook, clique is defined to be a *maximal* subset of pairwise adjacent vertices of the graph.

TERMINOLOGY: Every tree is a treewidth-1 graph, so treewidth is a measure of how much a graph varies from a tree. Similarly, every clique is a cliquewidth-2 graph, so cliquewidth is a measure of how much a graph varies from a clique. This analogy forms the basis for coining the term *cliquewidth* (*cf.* [CoOl00]).

EXAMPLE

E13: A cliquewidth-3 construction is given in Figure 2.4.14. As in Example 12, the relevant operations are identified at each node of the decomposition tree (left) for the graph G shown on the right.

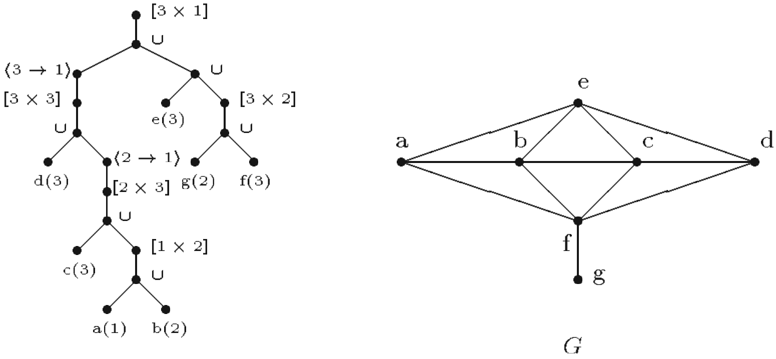


Figure 2.4.14 **A cliquewidth-3 construction.**

k-NLC Graphs

DEFINITION

D29: Let [k] denote the set of integers $\{1, 2, \ldots, k\}$ and let B denote a bipartite graph on [k] × [k]. A ***k-NLC (node-label-controlled) graph*** is defined recursively as follows:

- Any graph G with $V(G) = \{v\}$ and $l(v) \in [k]$ is a k-NLC graph.
- If G_1 and G_2 are k-NLC graphs and $i, j \in [k]$, then the join $G_1 \times_B G_2$ is a k-NLC graph, where $G_1 \times_B G_2$ is formed from $G_1 \cup G_2$ by adding all edges (v_1, v_2) where $v_1 \in V_1$, $l(v_1) = i$; $v_2 \in V_2$, $l(v_2) = j$ and (i, j) is an edge in E_B.
- The graph $(G_1)_{i \to j}$ is a k-NLC graph, which is formed from G_1 by switching all vertices with label i to label j.

EXAMPLE

E14: The graph in Figure 2.4.14 is a 2-NLC graph. In Figure 2.4.15, the decomposition tree has leaves corresponding to the vertices a, b, c, d, e, f, and g with starting labels drawn from the set k = $\{1, 2\}$ as shown. Relevant operations are identified with the internal nodes of the tree, *i.e.*, (i → j) for label switching and (i, j) indicating the specific edge from E_B inducing the stated composition.

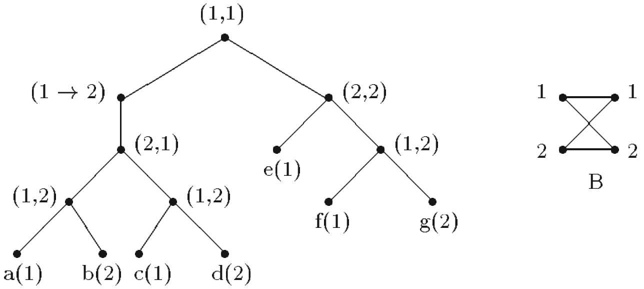


Figure 2.4.15 A 2-NLC graph construction.

k-HB Graphs

Denoting "homogeneous balanced," these graphs produce a modular decomposition upon application of a certain decomposition algorithm.

DEFINITION

D30: ***k-HB (homogeneous balanced) graphs*** are graphs for which there is a particular $O(n^{k+2})$-time top-down decomposition algorithm that constructs a pseudo-cliquewidth-$(k+2^k)$ balanced decomposition.

REMARKS

R6: Top-down decomposition refers to a recognition algorithm that places a candidate graph at the root of a tree, and then decomposes this graph into smaller subgraphs that become its children in the tree, and so on recursively until reaching the leaves of the tree.

R7: A pseudo-cliquewidth decomposition is simliar to a k-NLC decomposition, except that the vertex labels used at one node in the tree are not enforced at other nodes.

R8: A balanced decomposition of an n-vertex graph is a decomposition tree that has height $O(\lg n)$.

R9: The requirement that the decomposition must be balanced is more restrictive, while simultaneously the pseudo-cliquewidth condition is less restrictive. This trade-off yields the class of k-HB graphs. For more details on these matters, see [BoJoRaSp02].

R10: However, k-HB graphs are an ambiguously defined class due to the nondeterministic nature of this decomposition algorithm. On the other hand, the decomposition is guaranteed to succeed for every cliquewidth-k graph despite this nondeterminism, so every cliquewidth-k graph is a k-HB graph.

2.4.2 Equivalences and Characterizations

Relationships between Recursive Classes

A number of equivalences serve to relate many of the recursive graph classes defined in the previous subsection. Several of these are listed below. Unless a specific source is cited, a good general and fairly comprehensive reference for Facts 12 through 19 (and others) is [BrLeSp99].

FACTS

F12: A graph has treewidth at most k if and only if it is a partial k-tree.

F13: Every bandwidth-k graph is a pathwidth-k and thus a treewidth-k graph.

F14: The class of partial k-trees can be defined as a $(k+1)$-terminal recursive graph class (*cf.* [WiHe86], [Wi87]).

F15: 1-trees are trees in the usual sense and have treewidth 1.

F16: Trees are series-parallel graphs where only the jackknife operation is used.

F17: Series-parallel graphs in which only the series and parallel operations are used are precisely the 2-terminal series-parallel graphs.

F18: Series-parallel and outerplanar graphs are partial 2-trees and have treewidth 2.

F19: Halin graphs are contained in the class of partial 3-trees; they are also defined as a class of 3-terminal graphs by an appropriate choice of composition operations.

F20: [CoEnRo93] Cographs are precisely the cliquewidth-2 graphs.

F21: [CoOl00] Every treewidth-k graph is a cliquewidth-$(2^{k+1}+1)$ graph.

F22: [RoSe91-a] Every graph of branchwidth at most k has treewidth at most $3k/2$.

F23: [RoSe91-a] Every graph of treewidth at most k has branchwidth at most $k+1$.

F24: [Wa94] Cographs are exactly the 1-NLC graphs.

F25: [Wa94] Every treewidth-k graph is a $(2^{k+1}-1)$-NLC graph.

F26: [Wa94] Every cliquewidth-k graph is a k-NLC graph.

F27: [Wa94] Every k-NLC graph is a cliquewidth-$2k$ graph.

F28: [BoJoRaSp02] Every cliquewidth-k graph is a k-HB graph.

Characterizations

Structural characterizations of recursive graph classes are generally stated in terms of forbidden subgraph minors.

DEFINITIONS

D31: An ***edge-extraction*** operation on a graph $G=(V,E)$ removes an edge e leaving a graph, $G-e$, with $V(G-e)=V$ and $E(G-e)=E-\{e\}$.

D32: The operation of ***edge-contraction*** produces a graph with edge-set $E-\{e\}$ but with a vertex set obtained by replacing ("merging") the vertices defining e in G, thus creating a new single vertex where the latter inherits all of the adjacencies of the pair of replaced vertices, without introducing loops or multiple edges.

D33: A graph H is a ***minor*** of a graph G if and only if it can be obtained from G by a finite sequence of edge-extraction and edge-contraction operations.

REMARKS

R11: A result apparently first conjectured (but unpublished) by K. Wagner asserts the following: Suppose $\mathcal{F}$ is a graph class with the property that if G is in $\mathcal{F}$ and H is contained as a minor in G, then H is in $\mathcal{F}$, *i.e.,* the class $\mathcal{F}$ is closed under minors. Then there exists a finite set $\{H_1, H_2, \ldots, H_k\}$ of graphs, the ***forbidden minors*** such that G is in $\mathcal{F}$ if and only if it contains no minor isomorphic to any member H_i for $1 \leq i \leq k$.

R12: Robertson and Seymour ([RoSe88-b]) confirmed Wagner's conjecture and with their proof, established that any graph class $\mathcal{F}$ closed under minors can be recognized in polynomial time. Unfortunately, this outcome, although deep, is an existential one; we do not know the number of forbidden minors or their sizes in an arbitrary case.

R13: The class of partial k-trees is closed under minors and thus, by the Robertson-Seymour results is completely characterized by a finite set of forbidden minors.

R14: The forbidden minors for partial 3-trees are known (see Fact 33 below), but complete lists of explicit minors for partial k-trees are not known for values of $k \geq 4$.

FACTS

F29: [CoLeBu81] Cographs have no induced paths P_4.

F30: Trees are graphs having no K_3 minor.

F31: The set of forbidden minors of partial 2-trees is a singleton consisting of the complete graph, K_4.

F32: The forbidden minors of outerplanar graphs are K_4 and $K_{2,3}$.

F33: The class of partial 3-trees has four forbidden minors: K_5 and the three graphs shown in Figure 2.4.16.

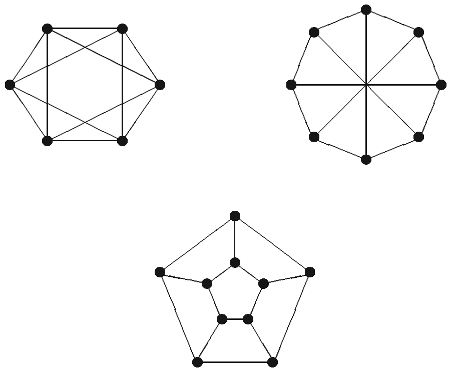


Figure 2.4.16 **Forbidden minors of partial 3-trees.**

2.4.3 Recognition

In order to solve graph problems on recursive classes and particularly, to do so efficiently, it is necessary that membership in the classes be ***quickly recognized***.

REMARKS

R15: Some recognition cases are direct and essentially *ad hoc*. For example, Halin graphs can be recognized by first testing for 3-connectivity. Then simply embed the candidate structure in the plane (since easy to test, one can assume planarity), select any cycle of edges defining a face on the plane embedding, remove the edges and test if the graph remaining is a tree of the stated form (see [CoNaPu83]).

R16: Partial 2-trees or series-parallel graphs are recognizable, unambiguously, by successive application of the following ***reduction operations*** (*cf.* [Du65]): replacement of any vertex of degree 2 say v_j and its incident edges (v_i, v_j) and (v_j, v_k) by a new edge (v_i, v_k); replacement of any pair of multiple edges by a single edge; and elimination of any edges incident to a vertex of degree 1 unless only one edge remains. Then a single edge remains, upon an admissible application of these reduction operations, if and only if the original graph is a partial 2-tree; otherwise, the process will stop with either K_4 or a graph with a K_4 minor.

R17: Similar reduction operations have also been described in the case of partial 3-trees (*cf.* [ArPr86]) as well as for partial 4-trees ([Sa96]).

EXAMPLE

E15: An illustration of a successful reduction sequence is shown in Figure 2.4.17.

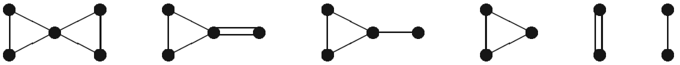


Figure 2.4.17 **Reduction operations for a partial 2-tree.**

Recognition of Recursive Classes

FACTS

F34: Trees can be recognized and their decomposition constructed in linear time.

F35: Series-parallel graphs can be recognized and their decomposition constructed in linear time.

F36: Treewidth-k, pathwidth-k, branchwidth-k, and bandwidth-k graphs can be recognized and their decompositions constructed in $O(n^{k+2})$ time.

COMPUTATIONAL NOTE: For fixed k the polynomial-time algorithms of Fact 36 are practical.

F37: The graph classes of Fact 36 can be recognized in *linear-time* for fixed k.

COMPUTATIONAL NOTE: The corresponding algorithms referred to by Fact 37 are not practical because their running times possess enormous hidden constants.

F38: When $k \leq 4$, more practical linear-time recognition algorithms have been found for the graph classes in Fact 36 (*cf.* [MaTh91] for $k = 3$; [Sa96] when $k = 4$).

F39: When k is part of the problem instance, the recognition problems associated with the graphs of Fact 36 are $\mathcal{NP}$-complete.

F40: Branchwidth can be determined in polynomial time for planar graphs [SeTh94].

F41: Since partial k-trees are characterizable by a finite set of forbidden minors, they are polynomially recognizable (*cf.,* [RoSe88-b]).

COMPUTATIONAL NOTE: Fact 41 was established in the graph minors results of Robertson and Seymour. However, the result is existential rather than constructive and so the actual exhibition of the implied algorithms remains elusive.

F42: [Wi87] Every k-terminal graph is a treewidth-k' graph for some k' that depends upon k and the particular set of recursive composition operations. For example, if m denotes the maximum *arity* of any operation, then $k' \leq km$.

TERMINOLOGY: The term "arity" refers to the number of operands. For example, a binary operation has arity 2.

F43: [CoPeSt85] Cographs can be recognized and their decomposition constructed in linear time.

F44: The complexity status of recognizing cliquewidth-k graphs is open.

F45: The complexity status of recognizing k-NLC graphs is open.

F46: [BoJoRaSp02] In the case of k-HB graphs, algorithms for problems defined on same are robust with respect to cliquewidth-k graphs. That is, such an algorithm either determines the correct answer or reports that the decomposition was unsuccessful and hence the input graph is not a cliquewidth-k graph. The $O(n^{k+2})$-time decomposition algorithm for k-HB graphs is guaranteed to succeed for all cliquewidth-k graphs as well as some others.

References

[ArCoPr87] S. Arnborg, D. G. Corneil, and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM J. Algebraic and Discrete Methods* **8** (1987), 277–284.

[ArCoPrSe93] S. Arnborg, B. Courcelle, A. Proskurowski, and D. Sesse, An algebraic theory of graph reductions, *J. ACM* **40** (1993), 1134–1164.

[ArHePr94] S. Arnborg, S. Hedetniemi, and A. Proskurowski, (eds.), Efficient algorithms and partial k-trees, special issue of *Discrete Applied Math.* **54** (1994).

[ArPr85] S. Arnborg and A. Proskurowski, Characterization and recognition of partial k-trees, *Congressus Numerantium* **47** (1985), 69–75.

[ArPr86] S. Arnborg and A. Proskurowski, Characterization and recognition of partial 3-Trees, *SIAM J. Algebraic and Discrete Methods* **7** (1986) 305–314.

[ArPrCo90] S. Arnborg, A. Proskurowski, and D. G. Corneil, Forbidden minors characterization of partial 3-trees, *Discrete Math.* **80** (1990), 1–19.

[BaOl98] L. Babel and S. Olariu, On the structure of graphs with few P_4s, *Discrete Applied Math.* **84** (1998), 1–13.

[BePi71] L. W. Beineke and R. E. Pippert, Properties and characterizations of k-trees, *Mathematika* **18** (1971), 141–151.

[BiRoSeTh91] D. Bienstock, N. Robertson, P. D. Seymour, and R. Thomas, Quickly excluding a forest, *J. Combin. Theory Series B* **52** (1991), 274–283.

[Bo88] R. B. Borie, Recursively constructed graph families: membership and linear algorithms, Ph.D. Dissertation, School of Information and Computer Science, Georgia Institute of Technology, 1988.

[Bo90] H. L. Bodlaender, Classes of graphs with bounded treewidth, *Tech. Rep. RUU-CS-86-22*, Department of Computer Science, Utrecht University, The Netherlands, 1990.

[Bo93] H. L. Bodlaender, A tourist guide through treewidth, *Acta Cybernetica* **11** (1993), 1–23.

[Bo96] H. L. Bodlaender, A linear-time algorithm for finding tree decompositions of small treewidth, *SIAM J. Computing* **25** (1996), 1305–1317.

[BoGiHaKl95] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, Approximating treewidth, pathwidth, and minimum elimination tree height, *J. Algorithms* **18** (1995), 238–255.

[BoJoRaSp02] R. B. Borie, J. L. Johnson, V. Raghavan, and J. Spinrad, Robust polynomial time algorithms on cliquewidth-k graphs, manuscript (2002).

[BoKl96] H. L. Bodlaender and T. Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs, *J. Algorithms* **21** (1996), 358–402.

[BoMö93] H. L. Bodlaender and R. H. Möhring, The pathwidth and treewidth of cographs, *SIAM J. Discrete Math.* **6** (1993), 181–186.

[BoPaTo91] R. B. Borie, R. G. Parker, and Tovey, C.A., Deterministic decomposition of recursive graph classes, *SIAM J. Discrete Math.* **4** (1991), 481–501.

[BrLeSp99] A. Brandstädt, V. B. Le, and J. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Math. and Applications, SIAM, Philadelphia (1999).

[Co90] B. Courcelle, The monadic second-order logic of graphs I: Recognizable sets of finite graphs, *Inform. and Comput.* **85** (1990), 12–75.

[Co92] B. Courcelle, The monadic second-order logic of graphs III: Tree-decompositions, minors, and complexity issues, *Inform. Théorique Appl.* **26** (1992), 257–286.

[Co95] B. Courcelle, The monadic second-order logic of graphs VIII: Orientations, *Annals of Pure and Applied Logic* **72** (1995), 103–143.

[Co96] B. Courcelle, The monadic second-order logic of graphs X: Linear orderings, *Theoretical Computer Science* **160** (1996), 87–143.

[CoEnRo93] B. Courcelle, J. Engelfriet, and G. Rozenberg, Handle-rewriting hypergraph grammars, *J. of Computer and Systems Sciences* **46** (1993), 218–270.

[CoKi83] D. G. Corneil and D. G. Kirkpatrick, Families of recursively defined perfect graphs, *Congressus Numerantium* **39** (1983), 237–246.

[CoLeBu81] D. G. Corneil, H. Lerchs, and L. S. Burlington, Complement reducible graphs, *Discrete Applied Math.* **3** (1981), 163–174.

[CoMo93] B. Courcelle and M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, *Theoretical Computer Science* **109** (1993), 49–82.

[CoNaPu83] G. Cornuejols, D. Naddef, and W. R. Pulleyblank, Halin graphs and the travelling salesman problem, *Math Programming* **26** (1983), 287–294.

[CoOl00] B. Courcelle and S. Olariu, Upper bounds to the clique-width of graphs, *Discrete Applied Math.* **101** (2000), 77–114.

[CoPeSt84] D. G. Corneil, Y. Perl, and L. K. Stewart, Cographs: recognition, applications and algorithms, *Congressus Numerantium* **43** (1984), 249–258.

[CoPeSt85] D. G. Corneil, Y. Perl, and L. Stewart, A linear recognition algorithm for cographs, *SIAM J. Computing* **14** (1985), 926–934.

[Du65] R. J. Duffin, Topology of series-parallel graphs, *J. Math. Anal. Appl.* **10** (1965), 303–318.

[GoRo99] M. C. Golumbic and U. Rotics, On the clique-width of perfect graph classes, *Lecture Notes in Computer Science* 1665 (1999), 135–147.

[GrSk91] D. Granot and D. Skorin-Kapov, NC Algorithms for recognizing partial 2-trees and 3-trees, *SIAM J. Algebraic and Discrete Methods* **4** (1991), 342–354.

[HeYe87] X. He and Y. Yesha, Parallel recognition and decomposition of two-terminal series-parallel graphs, *Information and Computing* **75** (1987), 15–38.

[KaIsUe85] Y. Kajitani, A. Ishizuka, and S. A. Ueno, Characterization of the partial k-tree in terms of certain structures, pp. 1179–1182 in *Proc. ISCAS '85*, 1985.

[Kl94] T. Kloks, Treewidth-computations and approximations, *Lecture Notes in Computer Science* **842**, Springer-Verlag, 1994.

[KlKr95] T. Kloks and D. Kratsch, Treewidth of chordal bipartite graphs, *J. Algorithms* **19** (1995), 266–281.

[MaTh91] J. Matoušek and R. Thomas, Algorithms for finding tree-decompositions of graphs, *J. Algorithms* **12** (1991), 1–22.

[Pr93] A. Proskurowski, Graph reduction and techniques for finding minimal forbidden minors, pp. 591–600 in *Proc. AMS Workshop on Graph Minors*, Seattle 1991, Contemporary Math. AMS **147**, Graph Structure Theory, 1993.

[Re92] B. Reed, Finding approximate separators and computing treewidth quickly, pp. 221–228 in *Proc. 9th Symposium on Theoretical Aspects of Computer Science*, 1992.

[Re93] B. Reed, Treewidth and Tangles: A New Connectivity Measure and Some Applications, pp. 87–162 in *Surveys in Combinatorics, 1997*, ed. R. A. Bailey, Cambridge University Press, 1997.

[Ro74] D. J. Rose, On simple characterization of k-trees, *Discrete Math.* **7** (1974), 317–322.

[RoSe83] N. Robertson and P. D. Seymour, Graph minors. I. Excluding a forest, *J. Combin. Theory Series B* **35** (1983), 39–61.

[RoSe84] N. Robertson and P. D. Seymour, Graph minors. III. Planar treewidth, *J. Combin. Theory Series B* **36** (1984), 49–64.

[RoSe86-a] N. Robertson and P. D. Seymour, Graph minors. II. Algorithmic aspects of treewidth, *J. Algorithms* **7** (1986), 309–322.

[RoSe86-b] N. Robertson and P. D. Seymour, Graph minors. V. Excluding a planar graph, *J. Combin. Theory Series B* **41** (1986), 92–114.

[RoSe86-c] N. Robertson and P. D. Seymour, Graph minors. VI. Disjoint paths across a disc, *J. Combin. Theory Series B* **41** (1986), 115–138.

[RoSe88-a] N. Robertson and P. D. Seymour, Graph minors. VII. Disjoint paths on a surface, *J. Combin. Theory Series B* **45** (1988), 212–254.

[RoSe88-b] N. Robertson and P. D. Seymour, Graph minors. XX. Wagner's conjecture, manuscript, 1988.

[RoSe90–a] N. Robertson and P. D. Seymour, Graph minors. IV. Treewidth and well-quasi-ordering, *J. Combin. Theory Series B* **48** (1990), 227–254.

[RoSe90-b] N. Robertson and P. D. Seymour, Graph minors. IX. Disjoint crossed paths, *J. Combin. Theory Series B* **49** (1990), 40–77.

[RoSe90-c] N. Robertson and P. D. Seymour, Graph minors. VIII. A Kuratowski theorem for general surfaces, *J. Combin. Theory Series B* **48** (1990), 255–288.

[RoSe91-a] N. Robertson and P. D. Seymour, Graph minors. X. Obstructions to tree-decompositions, *J. Combin. Theory Series B* **52** (1991), 153–190.

[RoSe91-b] N. Robertson and P. D. Seymour, Graph minors. XVI. Excluding a non-planar graph, manuscript, 1991.

[RoSe92] N. Robertson and P. D. Seymour, Graph minors. XXII. Irrelevant vertices in linkage problems, manuscript, 1992.

[RoSe94] N. Robertson and P. D. Seymour, Graph minors. XI. Distance on a surface, *J. Combin. Theory Series B* **60** (1994), 72–106.

[RoSe95] N. Robertson and P. D. Seymour, Graph minors. XIII. The disjoint paths problem, *J. Combin. Theory Series B* **63** (1995), 65–110.

[RoSeTh94] N. Robertson, P. D. Seymour, and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Series B* **62** (1994), 323–348.

[Sa96] D. P. Sanders, On linear recognition of treewidth at most four, *SIAM J. Discrete Math.* **9** (1996), 101–117.

[SaTu90] A. Satyanarayana and L. Tung, A characterization of partial 3-trees, *Networks* **20** (1990), 299–322.

[Sc88] P. Scheffler, What graphs have bounded treewidth?, in *Proc. Fischland Colloquium on Discrete Math. and Applications*, Rostock Math. Kolloq. (1988).

[SeTh94] P. D. Seymour and R. Thomas, Call routing and the ratcatcher, *Combinatorica* **14** (1994), 217–241.

[Wa94] E. Wanke, k-NLC graphs and polynomial algorithms, *Discrete Applied Math.* **54** (1994), 251–266.

[Wi87] T. V. Wimer, Linear algorithms on k-terminal graphs, Ph.D. Dissertation, Department of Computer Science, Clemson University, 1987.

GLOSSARY FOR CHAPTER 2

adjacency list representation – for a graph or digraph $G = (V, E)$: an array L of $|V|$ lists, one for each vertex in V; for each vertex i, there is a pointer L_i to a linked list containing all vertices j adjacent to i.

adjacency matrix representation – of a simple graph or digraph $G = (V, E)$: a $|V| \times |V|$ matrix A, where $A[i, j] = 1$ if there is an edge from vertex i to vertex j, and $A[i, j] = 0$ otherwise.

adversary reconstruction number – of a graph G: the minimum number k such that every choice of k subgraphs from the deck of G determines G uniquely.

all-pairs shortest-paths problem: determining the shortest path between every pair of vertices in a graph.

ally reconstruction number – of a graph G: same as the reconstruction number.

back edge – for a spanning tree in a directed graph: a nontree edge that joins a vertex to a proper ancestor.

bandwidth-k graph: a graph for which there exists a vertex labeling $h : V \to \{1, 2, \ldots |V|\}$ such that $\{u, v\} \in E \Rightarrow |h(u) - h(v)| \leq k$.

bidegreed graph: a graph whose vertices have only two possible degrees.

branch-decomposition – of a graph $G = (V, E)$: a pair (T, f), where T is a tree in which every non-leaf vertex has exactly three neighbors and f is a bijection from the leaves of T to E.

___, **partial**: a branch-decomposition in which the degree of every non-leaf vertex in T is *at least* 3.

branchwidth-k graph: a graph whose branchwidth is no greater than k.

branchwidth – of a graph G: the minimum width taken over all branch-decompositions of G.

breadth-first search: a systematic method for finding all vertices of a graph that are reachable from a given start vertex, by beginning at the start vertex and then visiting the unvisited vertices in a shortest-distance-from-the-start-vertex order.

breadth-first tree: tree of all vertices reachable from a given start vertex of a graph during a breadth-first search.

canonical numbering algorithm for graphs: an algorithm $\mathcal{N}$ that outputs a permuted sequence $\mathcal{N}(G) = \{v_{i_1}, v_{i_2}, \ldots, v_{i_n}\}$ of the vertices of its input graph, such that two graphs $G = (\{v_1, \ldots, v_n\}, E)$ and $H = (\{u_1, \ldots, u_n\}, F)$ are isomorphic iff the bijection $\{v_{i_j} \mapsto u_{i_j} : j = 1, \ldots, n\}$ is a graph isomorphism.

canonical numbering – of a graph: a numbering of the vertex-set produced by a canonical numbering algorithm.

CAP: see *color automorphism problem.*

certificate for isomorphism: a graph invariant ι such that for any two graphs $G_1, G_2 \in \mathcal{G}$, $\iota(G_1) = \iota(G_2)$ iff $G_1 \approx G_2$.

chordal graph: a graph that contains no induced cycles of length greater than 3.

class edge-reconstruction number – of a graph G in a class $\mathcal{C}$: the least number of subgraphs in the edge-deck of G which, together with the information that G is in the class $\mathcal{C}$, guarantees that G is uniquely determined.

class reconstruction number – of a graph G in a class $\mathcal{C}$: the least number of subgraphs in the deck of G which, together with the information that G is in the class $\mathcal{C}$, guarantees that G is uniquely determined.

claw-free graph: a graph that has no induced subgraph isomorphic to $K_{1,3}$.

cliquewidth – of a graph: the minimum number of labels that are sufficient to construct a graph from isolated vertices, while using only the union, module join, and relabeling operations.

cliquewidth-k graph: defined recursively as follows ([k] denotes the set of integers $\{1, 2, \ldots, k\}$):

- Any graph G with $V(G) = \{v\}$ and $l(v) \in [k]$ is a cliquewidth-k graph.
- If G_1 and G_2 are cliquewidth-k graphs and $i, j \in [k]$, then
  (1) the disjoint union $G_1 \cup G_2$ is a cliquewidth-k graph.
  (2) the graph $(G_1)_{i\times j}$ is a cliquewidth-k graph, where $(G_1)_{i\times j}$ is formed from G_1 by adding all edges (v_1, v_2) such that $l(v_1) = i$ and $l(v_2) = j$.
  (3) the graph $(G_1)_{i\to j}$ is a cliquewidth-k graph, where $(G_1)_{i\to j}$ is formed from G_1 by switching all vertices with label i to label j.

cograph: defined recursively as

- A graph with a single vertex is a cograph.
- If G_1 and G_2 are cographs, then the disjoint union $G_1 \cup G_2$ is a cograph.
- If G_1 and G_2 are cographs, then the cross-product $G_1 \times G_2$ is a cograph, which is formed by taking the union of G_1 and G_2 and adding all edges (v_1, v_2) where v_1 is in G_1 and v_2 is in G_2.

color automorphism problem (CAP): the problem of finding a set of generators for the subgroup of color-preserving permutations, within a given permutation group acting on a given colored set.

color class – for a graph: the set of all vertices that are assigned the same color.

coloring – of a graph G: a mapping $\sigma : V_G \to C$ from its vertex set to a set C (often a set of integers); alternatively, a partition $\sigma = [C_1, \ldots, C_m]$ of the vertex set into *color classes*.

___, **trivial** – for a graph: a coloring that assigns the same color to every vertex.

color-preserving mapping: a graph mapping such that any two like-colored vertices of the domain are mapped to like-colored vertices in the codomain.

cover of a graph G by $\mathcal{F}$ – for a sequence $\mathcal{F} = (F_1, F_2, \ldots, F_k)$ of graphs (in which different F_i could be isomorphic): a sequence $\mathcal{G} = (G_1, G_2, \ldots, G_k)$ of subgraphs of G (not necessarily distinct) such that (i) $G_i \simeq F_i$, $i = 1, \ldots, k$ and (ii) $G = \cup_i G_i$.; the number of covers of G by $\mathcal{F}$ is denoted by $c(\mathcal{F}, G)$.

cross edge – for a spanning forest in a directed graph: a nontree edge that joins two vertices that are neither ancestors nor descendants of each other.

deck – of a graph G: the collection $\mathcal{D}(G)$ of all vertex-deleted subgraphs of the graph G.

degree of a vertex v: the number of vertices adjacent to v.

degree sequence – of a graph G: the sequence of degrees of the vertices of G, written in non-descending order.

degree vector – of a graph coloring $\sigma = [C_1, \ldots, C_m]$: the vector assignment

$$v \mapsto \vec{deg}_\sigma(v) = [|N(v) \cap C_1|, \ldots, |N(v) \cap C_m|]$$

dense graph $G = (V, E)$: one in which the order of magnitude of $|E|$ is close to $|V|^2$.

depth-first forest: set of depth-first trees formed in a depth-first search of a graph.

depth-first search: a systematic method for visiting all vertices of a graph by beginning at a vertex, picking an unvisited adjacent vertex, and recursively continuing the search from that vertex.

depth-first tree: tree formed by tree edges discovered in a depth-first search of a graph.

edge-contraction – of an edge e in a graph $G = (V, E)$: an operation that results in a graph with edge-set $E - \{e\}$ but with a vertex-set obtained by replacing ("merging") the endpoints of e in G, thus creating a new single vertex where the latter inherits all of the adjacencies of the pair of replaced vertices, without introducing loops or multiple edges.

edge-deck – of a graph G: the collection $\mathcal{ED}(G)$ of all edge-deleted subgraphs of G.

edge-deleted subgraph – of a graph G: a graph $G - e$ obtained from G by deleting an edge e; also called *edge-deletion subgraph.*

k**-edge-deleted subgraph** – of a graph G: a subgraph obtained from G by deleting k of its edges.

edge-extraction – on a graph $G = (V, E)$: an operation that removes an edge e leaving the edge-deletion graph $G - e$.

edge-recognizable class: a class $\mathcal{C}$ of graphs such that, for any graph $G \in \mathcal{C}$, every edge-reconstruction of G is also in $\mathcal{C}$.

edge-reconstructible graph: a graph G whose every edge-reconstruction is isomorphic to G.

edge-reconstructible parameter: a graph parameter $\mathcal{P}$ such that, for any graph G with parameter value p, every edge-reconstruction of G also has value p for that parameter.

Edge-Reconstruction Conjecture: the conjecture that every graph on at least four edges is edge-reconstructible.

edge-reconstruction number – of a graph G: the least number of subgraphs in the edge-deck of G which guarantees that G is uniquely determined.

edge-reconstruction of a graph G: a graph H with the same edge-deck as G.

edge-reconstruction problem for a structure (D, Γ, E) – where all the subsets $E - x$ are given, up to action by the group Γ: the question of whether E can be reconstructed from these subsets uniquely, again up to action by the group Γ.

k**-edge-reconstruction problem**: the problem of determining uniquely, up to isomorphism, a graph or a structure from its k-edge-deleted subgraphs or sub-structures.

elementary graph: a graph in which any component is either an edge or a cycle.

endvertex – of a graph G: a vertex whose degree is 1.

endvertex-deck – of a graph G: the collection of graphs $G - v$ for all endvertices v of G.

endvertex-reconstructible graph: a graph that is uniquely determined by its endvertex deck.

Floyd-Warshall algorithm: an algorithm to compute the shortest length path (or least cost) between vertex i and vertex j, for all vertices i and j.

forward edge – for a spanning tree in a directed graph: a nontree edge that joins a vertex to a proper descendant.

graph isomorphsm problem (ISO): the problem of constructing an efficient algorithm to test whether two given graphs are isomorphic.

p**-group** – for a prime p: a group whose order is a power of the prime p.

Halin graph: planar graph whose edge set can be partitioned into a spanning tree, with no vertices of degree 2, and a cycle through the leaves of this tree.

k**-HB graph**: graph that yields a balanced modular decomposition when a certain decomposition algorithm is applied; see Definition 30 in §2.4.

illegitimate deck: a collection of graphs $G_1, G_2, \ldots, G_n$, each on $n-1$ vertices such that there is no graph G having the given collection as its deck.

illegitimate deck problem: the problem to determine whether or not a given collection of graphs is indeed the deck of some graph.

incidence matrix$_1$ representation – of a simple graph $G = (V, E)$ is a $|V| \times |E|$ matrix I, where $I[v, e] = 1$ if e is incident on v and 0 otherwise.

incidence matrix$_2$ representation – of a simple digraph $G = (V, E)$ is a $|V| \times |E|$ matrix I, where

$$I[v,e] = \begin{cases} -1 & \text{if edge } e \text{ is directed } \textit{to} \text{ vertex } v \\ 1 & \text{if edge } e \text{ is directed } \textit{from} \text{ vertex } v \\ 0 & \text{otherwise} \end{cases}$$

ISO: see *graph isomorphism problem*.

isomorphic graphs: two graphs G and H, such that there is an isomorphism $G \to H$.

isomorphism of labeled graphs G and H: an isomorphism $\phi : G \to H$, such that for each $v \in V_G$, the vertices v and $\phi(v)$ have the same label.

isomorphism of simple graphs: a vertex bijection that preserves adjacency relationships.

isomorphism-complete problem: a problem that is polynomially equivalent to *ISO*.

Kleene closure of a set of strings S: the set $S^* = \cup_{i=0}^{\infty} S^i$.

Kleene's algorithm: an algorithm for constructing a regular expression that describes all paths between every pair of vertices in a labeled graph.

labeled graph: a graph whose vertices and/or edges are labeled, possibly with repetitions, using symbols from a finite alphabet.

linear-time algorithm: algorithm that runs in $O(V + E)$ time for input graph $G = (V, E)$.

minor – of a graph G: a graph that can be obtained from G by a finite sequence of *edge-extraction* and *edge-contraction* operations.

module: with respect to a subgraph, a set of vertices that share exactly the same neighbors outside this subgraph.

monomorphism with forbidden X – of simple graphs G and H, where X is a subset of the edges of G: a bijection of V such that if $\{u, v\}$ is an edge in $E(G) - X$ then

$\{f(u), f(v)\}$ is also an edge in H, but if $\{u, v\}$ is an edge in X then $\{f(u), f(v)\}$ is not an edge in H. The number of monomorphisms from G to H with forbidden X is denoted by $[H]_{G \backslash X}$.

monomorphism – of simple graphs G and H: a one-to-one function $f : V_G \rightarrow V_H$ such that if $\{u, v\}$ is an edge of G, then $\{f(u), f(v)\}$ is an edge of H. The number of monomorphisms from G to H is denoted by $[H]_G$.

nauty: the name of a practical computer program for use in graph isomorphism testing. (The name is a quasi-acronym for "no automorphisms, yes".)

neighborhood – of a vertex v of a graph: the set of all vertices adjacent to v. It is denoted by $N(v)$.

k-NLC (node-label-controlled) graph: defined recursively as follows ($[k]$ denotes the set of integers $\{1, 2, \ldots, k\}$, and B denotes a bipartite graph on $[k] \times [k]$):

- Any graph G with $V(G) = \{v\}$ and $l(v) \in [k]$ is a k-NLC graph.
- If G_1 and G_2 are k-NLC graphs and $i, j \in [k]$, then the join $G_1 \times_B G_2$ is a k-NLC graph, where $G_1 \times_B G_2$ is formed from $G_1 \cup G_2$ by adding all edges (v_1, v_2) where $v_1 \in V_1$, $l(v_1) = i$; $v_2 \in V_2$, $l(v_2) = j$ and (i, j) is an edge in E_B.
- The graph $(G_1)_{i \rightarrow j}$ is a k-NLC graph, which is formed from G_1 by switching all vertices with label i to label j.

nondeterministic finite automaton: a directed graph (possibly with multiple edges) between the same pair of vertices, having a distinguished start state, a set of final states, and labels on the edges.

N-reconstructible digraph: a digraph D such that the set of triples $(D - v_i, \deg_{\text{in}}(v_i), \deg_{\text{out}}(v_i))$, for all vertices v_i of D, is sufficient information to determine D uniquely.

order – of an edge e in T in a branch-decomposition (T, f) of a graph $G = (V, E)$: the number of vertices $v \in V$ such that there exist leaves l_1 and l_2 of T residing in different components of $T - e$, where $f(l_1)$ and $f(l_2)$ are both incident on v.

partial k-tree: subgraph of a k-tree.

path in a graph: a sequence of edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$.

path-decomposition: a tree-decomposition whose tree is a path.

pathwidth-k graph: a graph that has pathwidth no greater than k.

pathwidth – of a graph G: the smallest width taken over all path-decompositions of G; measures how closely the graph resembles a path.

perfect: a graph in which every induced subgraph has chromatic number equal to the size of its maximum clique.

peripheral vertex of a tree: a vertex that has maximum distance from the center of the tree.

polynomial deck: the collection (multi-set) of the characteristic polynomials of all subgraphs in the deck.

polynomial-time algorithm: an algorithm that runs in $O((V + E)^k)$ time for input graph $G = (V, E)$ for some constant k.

property A_k – of a graph G: the property that whenever A and B are distinct k-sets of vertices of G, the graphs $G - A$ and $G - B$ are not isomorphic.

recognizable class of graphs: a class $\mathcal{C}$ of graphs such that, for any $G \in \mathcal{C}$, every reconstruction of G is also in C.

reconstructible graph: graph G whose every reconstruction is isomorphic to G.

reconstructible parameter: a graph parameter $\mathcal{P}$ such that, for any graph G with the value p for that parameter, every reconstruction of G also has paraemeter value p.

Reconstruction Conjecture: the conjecture that every graph with at least three vertices is reconstructible.

reconstruction index – of a group Γ: the smallest number t such that for any $E \subset D$ with $|E| \geq t$, the structure (D, Γ, E) is edge-reconstructible.

reconstruction number – of a graph G: the least number of subgraphs in the deck of G which guarantees that G is uniquely determined.

reconstruction of a graph G: a graph H with the same deck as G.

recursively constructed graph class: defined by a set (usually finite) of primitive or *base graphs*, in addition to one or more operations that compose larger graphs from smaller subgraphs; each operation involves either fusing specific vertices from each subgraph or adding new edges between specific vertices from each subgraph.

refinement of a graph coloring – an operation that yields a new coloring of the graph: two vertices with the same old color get the same new color if and only if they have the same numbers of neighbors of every old color.

regular expression: a notation for describing a regular set by using the operators union, concatenation, and Kleene closure.

series-parallel graph with distinguished *terminals* l and r, denoted (G, l, r) – defined recursively:

- The graph consisting of a single edge (v_1, v_2) is a series-parallel graph (G, l, r) with $l = v_1$ and $r = v_2$.
- A ***series operation*** $(G_1, l_1, r_1) \odot_s (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2. The terminals of the new graph are l_1 and r_2.
- A ***parallel operation*** $(G_1, l_1, r_1) \odot_p (G_2, l_2, r_2)$ forms a series-parallel graph by identifying l_1 with l_2 and r_1 with r_2. The terminals of the new graph are l_1 and r_1.
- A ***jackknife operation*** $(G_1, l_1, r_1) \odot_j (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2; the new terminals are l_1 and r_1.

set edge-reconstructible – graph or a parameter: a graph or a parameter that can be determined from the set of non-isomorphic subgraphs in the edge-deck.

set reconstructible – graph or a parameter: a graph or a parameter that can be determined from the set of non-isomorphic subgraphs in the deck.

sparse graph $G = (V, E)$: one in which the order of magnitude of $|E|$ is $|V|$ or less.

stabilization of a coloring σ: the coloring that results from iterating the refinement process until a stable coloring is obtained. It is denoted σ^*.

stable coloring: a graph coloring that is unchanged by the refinement operation.

structure: a triple (D, Γ, E) where D is a finite set, Γ is a group of permutations acting on D, and E is a subset of D.

k**-terminal recursive graph**: graph that has at most k special vertices called terminals, and that can be obtained by operations that fuse some of the terminals in its constituent k-terminal subgraphs. (See Definition 25 in §2.4.)

transitive closure of a graph G: a graph G^* that has an edge (i,j) if and only if there is a path of length 1 or more in G from i to j.

tree: a connected graph with no cycles, and sometimes with a designated *root*.

___, **recursively defined**: a graph with a single vertex r as its root r; or, a graph formed by joining the roots of two trees.

k**-tree** (recursively defined): the complete graph K_k; or, a graph constructed from a k-tree on n vertices by adding a vertex adjacent to all vertices of one of its K_k subgraphs, and only to those vertices.

___, **partial**: a subgraph of a k-tree.

tree-decomposition – of a graph $G = (V,E)$: a pair $(\{X_i \mid i \in I\}, T)$, such that $\{X_i \mid i \in I\}$ is a family of subsets of V and T is a tree with vertex set I such that

- $\bigcup_{i \in I} X_i = V$
- for all edges $(x,y) \in E$ there is an element $i \in I$ with $x, y \in X_i$
- for all triples $i, j, k \in I$, if j is on the path from i to k in T, then $X_i \bigcap X_k \subseteq X_j$.

treewidth – of a graph G: the minimum width taken over all tree-decompositions of G; measures how closely the graph resembles a tree.

treewidth-k **graph**: a graph whose treewidth is no greater than k.

vertex-deleted subgraph – of a graph G: a graph $G - v$ obtained from G by deleting a vertex v and all the edges incident to it; also called *vertex-deletion subgraph.*

k**-vertex-deleted subgraph** – of a graph G: a subgraph obtained from G by deleting k of its vertices and all the edges incident to them.

weakly edge-reconstructible graph – relative to a class $\mathcal{C}$: a graph $G \in \mathcal{C}$ such that every edge-reconstruction of G which is also in the class $\mathcal{C}$ is isomorphic to G.

weakly reconstructible graph – relative to a class $\mathcal{C}$: a graph $G \in \mathcal{C}$ such that every reconstruction of G which is also in the class $\mathcal{C}$ is isomorphic to G.

width$_1$ – of a branch decomposition (T, f): the maximum order of the edges of T.

width$_2$ – of a tree-decomposition $(\{X_i \mid i \in I\}, T)$: $\max_{i \in I}\{|X_i| - 1\}$.

Chapter 3

DIRECTED GRAPHS

3.1 BASIC DIGRAPH MODELS AND PROPERTIES

Jay Yellen, Rollins College

3.2 DIRECTED ACYCLIC GRAPHS

Stephen B. Maurer, Swarthmore College

3.3 TOURNAMENTS

K. B. Reid, California State University San Marcos

GLOSSARY

3.1 BASIC DIGRAPH MODELS AND PROPERTIES

Jay Yellen, Rollins College

3.1.1 Terminology and Basic Facts
3.1.2 A Sampler of Digraph Models
3.1.3 Binary Trees
References

Introduction

This section extends the basic terminology and properties begun in Chapter 1, and it describes several classical digraph models that preview later sections of the *Handbook*. Many of the basic methods and algorithms for digraphs closely resemble their counterparts for undirected graphs. Some general references for digraphs are [ChLe96], [GrYe99], and [We01]. A comprehensive and in-depth reference for digraphs is [BaGu01].

3.1.1 Terminology and Basic Facts

TERMINOLOGY NOTE: The term *arc* is used throughout this section instead of its synomym *directed edge*.

NOTATION: Often, when the digraphs under consideration do not have multi-arcs, an arc that is directed from vertex u to v is represented by the ordered pair (u, v) or by the juxtaposition uv.

TERMINOLOGY: An arc that is directed from vertex u to v is said to have ***tail*** u and ***head*** v.

Reachability and Connectivity

DEFINITIONS

D1: In a digraph, a ***directed walk*** from v_0 to v_n is an alternating sequence

$$W = \langle v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n \rangle$$

of vertices and arcs, such that $tail(e_i) = v_{i-1}$ and $head(e_i) = v_i$, for $i = 1, \ldots, n$.

TERMINOLOGY: A directed walk from a vertex x to a vertex y is also called an ***x-y directed walk***.

D2: The ***length of a directed walk*** is the number of arc-steps in the walk sequence.

D3: A ***connected digraph*** is a digraph whose underlying graph is connected. Elsewhere, the term ***weakly connected*** is often used to describe such digraphs.

D4: Let u and v be vertices in a digraph G. Then u and v are said to be ***mutually reachable*** in G if G contains both a directed u-v walk and a directed v-u walk. Every vertex is regarded as reachable from itself (by the trivial walk).

D5: A digraph is ***strongly connected*** if every two vertices are mutually reachable.

D6: A ***strong component*** of a digraph G is a maximal strongly connected subdigraph of G. Equivalently, a strong component is a subdigraph induced on a maximal set of mutually reachable vertices.

D7: Let $S_1, S_2, \ldots, S_r$ be the strong components of a digraph G. The ***condensation*** of G is the simple digraph G^* with vertex set $V_{G^*} = \{s_1, s_2, \ldots, s_r\}$, such that there is an arc in digraph G^* from vertex s_i to vertex s_j if and only if there is an arc in digraph G from a vertex in component S_i to a vertex in component S_j.

EXAMPLE

E1: Figure 3.1.1 shows a digraph G, its four strong components, S_1, S_2, S_3, S_4, and its condensation G^*. Notice that the vertex-sets of the strong components of G partition the vertex-set of G and that the edge-sets of the strong components do *not* include all the edges of G. This is in sharp contrast to the situation for an undirected graph G, in which the edge-sets of the components of G partition E_G.

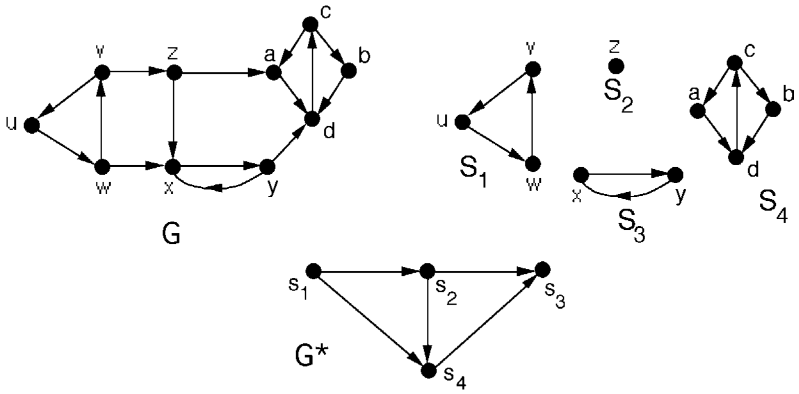


Figure 3.1.1 **A digraph, its four strong components, and its condensation.**

FACT

F1: Let G be a digraph. Then the mutual-reachability relation is an equivalence relation on V_G, and the strong components of digraph G are the subdigraphs induced on the equivalence classes of this relation.

Measures of Digraph Connectivity

We introduce a few basic measures of the connectedness of a digraph. Connectivity of graphs and digraphs is discussed extensively in §4.1 and §4.7. The concept of an *edge-cut* plays an important role in the study of flows in networks (§11.1) and in certain algebraic properties of a graph or digraph (§6.4).

DEFINITIONS

D8: A ***complete digraph*** is a simple digraph such that between each pair of its vertices, both (oppositely directed) arcs exist.

D9: A ***vertex-cut*** in a strongly connected digraph $G = (V, E)$ is a vertex subset $S \subset V$ such that the vertex-deletion subdigraph $G - S$ is not strongly connected, and an ***edge-cut*** (***arc-cut***) is an arc subset $F \subset E$ such that the arc-deletion subdigraph $G - F$ is not strongly connected.

D10: The ***(vertex-) connectivity*** of an n-vertex non-complete digraph $G = (V, E)$, denoted $\kappa_v(G)$, is the minimum size of a vertex subset S such that $G - S$ is neither strongly connected nor the trivial digraph. (The connectivity of a complete n-vertex digraph is $n - 1$.)

D11: The ***edge-connectivity*** of a non-trivial digraph, denoted $\kappa_e(G)$ is the minimum size of an edge subset F such that $G - F$ is not strongly connected.

NOTATION: When the context is clear, the vertex- and edge-connectivity are denoted κ_v and κ_e, respectively. Some other sections of the *Handbook* use the "traditional" κ and λ instead of κ_v and κ_e, respectively.

TERMINOLOGY NOTE: Synonyms for vertex-cut are ***cut*** and ***disconnecting set***. Synonyms for edge-cut are ***edge-disconnecting set*** (or ***arc-disconnecting set***) and ***cut-set***.

Directed Trees

DEFINITIONS

D12: A ***directed tree*** is a digraph whose underlying graph is a tree.

D13: A ***rooted tree*** is a directed tree having a distinguished vertex r, called the ***root***, such that for every other vertex v, there is a directed r-v path.

TERMINOLOGY NOTE: Occasionally encountered synonyms for rooted tree are ***out-tree***, ***branching***, and ***arborescence***.

REMARKS

R1: Since the underlying graph of a rooted tree is acyclic, the directed r-v path is unique.

R2: Designating a root in a directed tree does *not* necessarily make it a rooted tree.

Tree-Growing in a Digraph

Algorithm 3.1.1, shown below, is simply the basic tree-growing algorithm of §1.1 (Algorithm 1.1.1), recast for digraphs. Its output, as in Algorithm 1.1.1, is a rooted tree whose vertices are reachable from the starting vertex. But because the paths to these vertices are directed (i.e., one-way), the vertices in this *output tree* need not be mutually reachable from one another.

DEFINITION

D14: A ***frontier arc*** for a rooted tree T in a digraph is an arc whose tail is in T and whose head is not in T.

Algorithm 3.1.1: Basic Tree-Growing in a Digraph

```
Input: a digraph $G$ and a starting vertex $v \in V_G$.
Output: a rooted tree $T$ with root $v$ and a standard vertex-labeling of $T$.
    Initialize tree $T$ as vertex $v$.
    Write label 0 on vertex $v$.
    Initialize label counter $i := 1$
    While there is at least one frontier arc for tree $T$
        Choose a frontier arc $e$ for tree $T$.
        Let $w$ be $head(e)$ (which lies outside of $T$).
        Add arc $e$ and vertex $w$ to tree $T$.
        Write label $i$ on vertex $w$.
        $i := i + 1$
    Return tree $T$ and vertex-labeling of $T$.
```

COMPUTATIONAL NOTE: We assume that there is some implicit *default priority* for choosing vertices or edges, which is invoked whenever there is more than one frontier arc from which to choose.

EXAMPLE

E2: Figure 3.1.2 shows a digraph and all possible output trees that could result for each of the different starting vertices and each possible default priority. Two opposite extremes for possible output trees are represented here. When the algorithm starts at vertex u, the output tree spans the digraph. The other extreme occurs when the algorithm starts at vertex x (because x has outdegree 0). Notice that any two output trees in Figure 3.1.1 with the same vertex-set have roots that are mutually reachable.

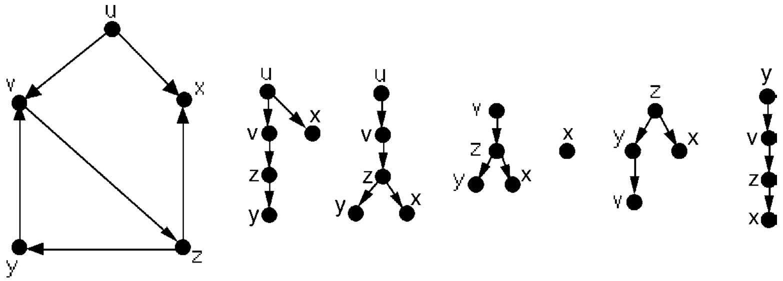


Figure 3.1.2 A digraph and all possible output trees.

FACTS

F2: Let u and v be two vertices of a digraph G. Then u and v are in the same strong component of G if and only if the output trees that result from starting Algorithm 3.1.1 at vertex u and at vertex v have the same vertex-set.

F3: If the digraph G is strongly connected, then the output tree is a spanning rooted tree of G, regardless of the starting vertex.

REMARK

R3: Example 2 above illustrates an important distinction between undirected and directed graphs: whereas tree-growing in an undirected graph provides a simple algorithm to determine the components of the graph, in a digraph this is not the case. Other differences were suggested earlier in Example 1. The use of tree-growing, specifically *depth-first search* (§10.1), in finding the strong components of a digraph is considerably more intricate than its undirected counterpart. For discussions of strong-component-finding algorithms, see, e.g., [BaGe99], [GrYe99, §11.4], and [St93].

Oriented Graphs

DEFINITIONS

D15: An ***oriented graph*** is a digraph obtained by choosing an orientation for each edge of an undirected simple graph. Thus, an oriented graph does not have both oppositely directed arcs between any pair of vertices, which means that an oriented tree is the same as a directed tree.

D16: A ***tournament*** is an oriented complete graph. That is, it has no self-loops, and between every pair of vertices, there is exactly one arc. See §3.3 for extensive coverage of tournaments.

D17: A graph G is ***strongly orientable*** if there exists an assignment of directions to the edge-set of G such that the resulting digraph is strongly connected.

EXAMPLE

E3: Of the three graphs shown in Figure 3.1.3, only the graph G_2 is strongly orientable.

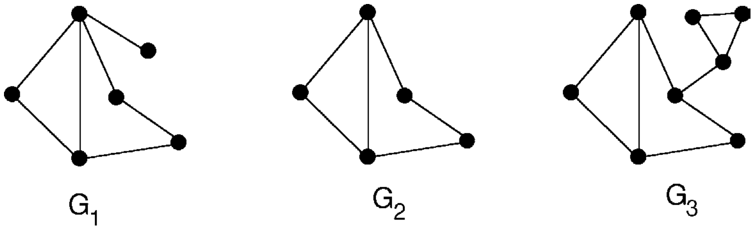


Figure 3.1.3 **Only the graph G_2 is strongly orientable.**

Notice that G_2 is the only graph in the example that does not have a cut-edge. In fact, the absence of cut-edges is a necessary and sufficient condition for a graph to be strongly orientable. This characterization of strongly orientable graphs was proved by H.E. Robbins in 1939.

FACT

F4: **Robbins's Theorem** [Ro39] A connected graph G is strongly orientable if and only if G has no cut-edges.

Adjacency Matrix of a Digraph

DEFINITION

D18: The ***adjacency matrix of a digraph*** $G = (V, E)$, denoted A_G, is given by

$$A_G[u,v] = \begin{cases} \text{the number of arcs from } \ u \ \text{ to } \ v \ \text{ if } u \neq v \\ \text{the number of self-loops at } \ v \ \text{ if } u = v \end{cases}$$

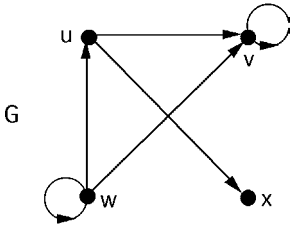
FACTS

F5: A row-sum in a directed adjacency matrix equals the outdegree of the corresponding vertex, and a column-sum equals the indegree.

F6: Let G be digraph with adjacency matrix A_G. Then the value of the entry $A_G^r[u,v]$ of the r^{th} power of matrix A_G equals the number of directed u-v walks of length r.

EXAMPLE

E4: The adjacency matrix of the digraph in Figure 3.1.4 uses the vertex ordering u, v, w, x. As an illustration of Fact 6, observe that the number of directed w-v walks equals 3, which is the (w,v)-entry of A_G^2.



$$A_G \quad = \quad \begin{array}{c} \\ u \\ v \\ w \\ x \end{array} \begin{array}{c} \begin{array}{cccc} u & v & w & x \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Figure 3.1.4 A digraph and its adjacency matrix.

REMARK

R4: Another matrix representation of a digraph is the *node-arc incidence matrix*, where the columns are labeled by the arcs, and the rows are labeled by the vertices (nodes). Connections between the incidence matrix of a graph or digraph and the structural properties of the graph are explored in §6.4.

3.1.2 A Sampler of Digraph Models

In this subsection, we sample a few of the digraph models. Acyclic digraph models are the focus of §3.2.

Markov Chains and Markov Digraphs

The topic of Markov processes is part of a more general area known as *stochastic processes*, a branch of mathematics and operations research with far-ranging applications and theoretical challenges. The reader may consult any of the standard texts in this subject for a formal presentation of these concepts (e.g., [Ci75], [Wi94]).

DEFINITIONS

D19: A sequence of random variables $\{X_t\}$, $t = 0, 1, 2, \ldots$, is a ***(finite) discrete-time Markov chain (DTMC)*** on a state-space $S = \{1, 2, \ldots, n\}$ if $X_t \in S$ for all times $t = 0, 1, 2, \ldots$, and the probability distribution of X_{t+1} depends only on the value of X_t. In particular,

$$prob(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = prob(X_{t+1} = j | X_t = i)$$

D20: A ***stationary DTMC*** satisfies the additional condition that for all states $i, j \in S$ and all times t, the ***transition probability*** $prob(X_{t+1} = j | X_t = i) = p_{ij}$ is independent of t.

D21: A ***Markov digraph*** $G = (V, E)$ of a stationary DTMC with state-space S and transition probabilities p_{ij} is a digraph with vertex-set $V = S$, arc-set $E = \{ij | p_{ij} > 0\}$, and to each arc $ij \in E$ is assigned the probability p_{ij}.

D22: The ***transition matrix*** of a Markov chain is the matrix whose ij^{th} entry is the transition probability p_{ij}.

EXAMPLE

E5: *A Gambler's Problem*: A gambler starts with \$3 and plays the following game. Two coins are tossed. If both come up heads, then he wins \$3; otherwise, he loses \$1. He plays until either he loses all his money or he reaches a total of at least \$5. Let X_t be the amount of money he has after t plays, with $X_0 = 3$. The state space is $S = \{0, 1, 2, 3, 4, 5\}$, and the sequence $\{X_t\}$ is a discrete-time Markov chain. The transition matrix and Markov digraph for this Markov chain are shown in Figure 3.1.5.

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ \geq 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & \geq 5 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ .75 & 0 & 0 & 0 & .25 & 0 \\ 0 & .75 & 0 & 0 & 0 & .25 \\ 0 & 0 & .75 & 0 & 0 & .25 \\ 0 & 0 & 0 & .75 & 0 & .25 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

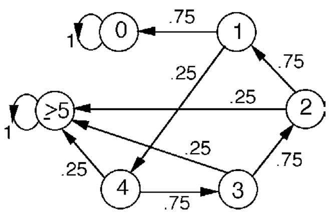


Figure 3.1.5 Gambler's transition matrix and Markov digraph.

Equipment-Replacement Policy

We present a digraph model that can be used to determine a replacement policy that minimizes the net cost of owning and operating a car for a pre-specified number of years.

EXAMPLE

E6: Suppose that today's price for a new car is \$16,000, and that the price will increase by \$500 for each of the next four years. The projected annual operating cost and resale value of this kind of car are shown in the table below. To simplify the setting, assume that these data do not change for the next five years.

Annual Operating Cost	Resale Value
\$600 (for 1st year of car)	\$13,000 (for a 1-year-old car)
\$900 (for 2nd year of car)	\$11,000 (for a 2-year-old car)
\$1200 (for 3rd year of car)	\$9,000 (for a 3-year-old car)
\$1600 (for 4th year of car)	\$8,000 (for a 4-year-old car)
\$2100 (for 5th year of car)	\$6,000 (for a 5-year-old car)

Digraph Model: The digraph has six vertices, labeled 1 through 6, representing the *beginning* of years 1 through 6. The beginning of year 6 signifies the end of the planning period. For each i and j with $i < j$, an arc is drawn from vertex i to vertex j and is assigned a weight c_{ij}, where c_{ij} is the total net cost of purchasing a new car at the beginning of year i and keeping it until the beginning of year j. Thus,

$$\begin{aligned} c_{ij} = &\text{ price of new car at beginning of year } i \\ &+ \text{sum of operating costs for years } i, i+1, \ldots, j-1 \\ &- \text{resale value at beginning of year } j \end{aligned}$$

Figure 3.1.6 shows the resulting digraph with seven of its 15 arcs drawn. The arc-weights are in units of \$100.

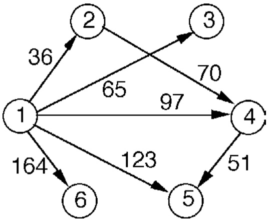


Figure 3.1.6 **Part of the digraph model for a car-replacement problem.**

The problem of determining the optimal replacement policy is reduced to finding the shortest (least-cost) path from vertex 1 to vertex 6. This is a simple task for ***Dijkstra's algorithm***, even for much larger instances of this kind of problem. Dijkstra's algorithm is discussed in §10.1.

The Digraph of a Relation and the Transitive Closure

Our focus here is on general relations and their transitive closure. Digraphs of *posets* (partially ordered sets) are discussed in §3.2.

DEFINITIONS

D23: A ***relation*** R on a finite set S is a subset of the cartesian product $S \times S$.

D24: The ***digraph representation of a relation*** R on a finite set S is the digraph whose vertices correspond to the elements of S, and whose arcs correspond to the ordered pairs in the relation; that is, an arc is drawn from vertex x to vertex y if $(x, y) \in R$.

Conversely, a digraph G induces a relation R on V_D in a natural way, namely, $(x, y) \in R$ if and only if there is an arc in digraph G from vertex x to vertex y.

D25: A ***transitive digraph*** is a digraph whose corresponding relation is transitive. That is, if there is an arc from vertex x to vertex y and an arc from y to z, then there is an arc from x to z.

D26: The ***transitive closure*** R^* of a binary relation R is the relation R^* defined by $(x, y) \in R^*$ if and only if there exists a sequence $x = v_0, v_1, v_2, \ldots, v_k = y$ such that $k \geq 1$ and $(v_i, v_{i+1}) \in R$, for $i = 0, 1, \ldots, k-1$. Equivalently, the transitive closure R^* of the relation R is the smallest transitive relation that contains R.

D27: Let G be the digraph representing a relation R. Then the digraph G^* representing the transitive closure R^* of R is called the ***transitive closure of the digraph*** G.

Thus, an arc (x, y), $x \neq y$ is in the transitive closure G^* if and only if there is a directed x-y path in G. Similarly, there is a self-loop in digraph D^* at vertex x if and only if there is a directed cycle in digraph G that contains x.

EXAMPLES

E7: Suppose a relation R on the set $S = \{a, b, c, d\}$ is given by

$$\{(a, a), (a, b), (b, c), (c, b), (c, d)\}$$

Then the digraph G representing the relation R and the transitive closure G^* are as shown in Figure 3.1.7.

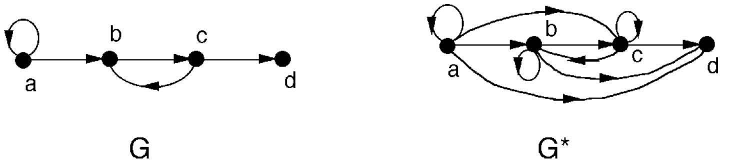


Figure 3.1.7 The digraph G and its transitive closure G^*.

E8: *Transitive Closure in a Paging Network:* Suppose that the arcs of an n-vertex digraph G represent the one-way direct links between specified pairs of nodes in an n-node paging network. Thus, an arc from vertex i to vertex j indicates that a page call can be transmitted from person i to person j.

To send an alert from person i to person j, it is not necessary to have a direct link from i to j. There need only be a directed i-j path. The transitive closure G^* of digraph G specifies all pairs i, j of vertices for which there exists a directed i-j path in G.

Constructing the Transitive Closure of a Digraph: Warshall's Algorithm

Let G be an n-vertex digraph with vertices $v_1, v_2, \ldots, v_n$. A computationally efficient algorithm, due to Warshall [Wa62], constructs a sequence of digraphs, $D_0, D_1, \ldots, D_n$, such that $D_0 = G$, D_{i-1} is a subgraph of D_i, $i = 1, \ldots, n$, and such that D_n is the transitive closure of D. Digraph D_i is obtained from digraph D_{i-1} by adding to D_{i-1} an arc (v_j, v_k) (if it is not already in D_{i-1}) whenever there is a directed path of length 2 in D_{i-1} from v_j to v_k, having v_i as the internal vertex.

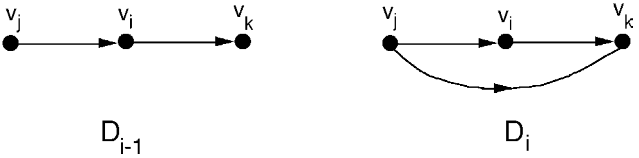


Figure 3.1.8 The arc (v_j, v_k) is added to digraph D_{i-1}.

Algorithm 3.1.2: Warshall's Transitive Closure [Wa62]

```
Input: an $n$-vertex digraph $D$ with vertices $v_1, v_2, \ldots, v_n$.
Output: the transitive closure of digraph $D$.

    Initialize digraph $D_0$ to be digraph $G$.
    For $i = 1$ to $n$
        For $j = 1$ to $n$
            If $(v_j, v_i)$ is an arc in digraph $D_{i-1}$
                For $k = 1$ to $n$
                    If $(v_i, v_k)$ is an arc in digraph $D_{i-1}$
                        Add arc $(v_j, v_k)$ to $D_{i-1}$ (if it is not already there).
    Return digraph $D_n$.
```

Activity-Scheduling Networks

In large projects, often there are some tasks that cannot start until certain others are completed. Figure 3.1.9 shows a digraph model of the *precedence relationships* among some tasks for building a house. Vertices correspond to tasks. An arc from vertex u to vertex v means that task v cannot start until task u is completed. To simplify the drawing, arcs that are implied by transitivity are not drawn. This digraph is the *cover diagram* of a partial ordering of the tasks. Section 3.2 discusses this model further and introduces a different model in which the tasks are represented by the arcs of a digraph.

	Activity
1	Foundation
2	Walls and ceilings
3	Roof
4	Electrical wiring
5	Windows
6	Siding
7	Paint interior
8	Paint exterior

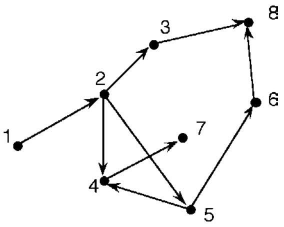


Figure 3.1.9 An activity digraph for building a house.

Scheduling the Matches in a Round-Robin Tournament

Suppose that each pair of n teams is to play one match in a tournament. Typically, one would like to schedule the matches so that all matches are completed in a minimum number of days (assume that each team plays at most one match on a given day). If the teams are from different cities, an additional objective is to have an equitable distribution of home and away matches. We preview here a strategy that is discussed in §5.6.5.

DEFINITIONS

D28: A ***compact schedule*** for a round-robin tournament is one in which each team plays a match each day.

D29: A team is said to have a ***break*** if it is either home for two consecutive matches or away for two.

D30: A ***proper arc-coloring*** of a digraph $G = (V, E)$ is an assignment of colors to the arcs in G so that any two arcs that have an endpoint in common are assigned different colors. Graph coloring is discussed in §5.1 and §5.2, and the related concept of graph factorization is discussed in §5.4.

REMARK

R5: An algorithm for constructing a compact schedule for a n-team round-robin tournament, where n is even, that minimizes the total number of breaks is given in §5.6 (Algorithm 5.6.1). The strategy is based on orienting the edges of a complete graph and then producing a proper arc-coloring so that each color is assigned to exactly $n/2$ arcs.

Flows in Networks

A pipeline network for transporting oil from a single source to a single sink is one prototype of a network model. Each arc represents a section of pipeline, and the endpoints of an arc correspond to the junctures at the ends of that section. The arc capacity is the maximum amount of oil that can flow through the corresponding section per unit time. A network could just as naturally represent a system of truck routes for transporting commodities from supply points to demand points, or it could represent a network of phone lines from one distribution center to another.

DEFINITIONS

D31: A ***cost flow network*** $G = (V, E, cap, c, b)$ is a directed graph with vertex-set V, arc-set E, a nonnegative capacity function $cap : E \to N$, a linear cost function $c : E \to Z$, and an integral supply vector $b : V \to Z$ that satisfies $\sum\limits_{w \in V} b(w) = 0$.

D32: An ***s-t flow network*** $G = (V, E, cap, s, t)$ is a directed graph (typically without the cost and supply functions) with a nonnegative capacity function $cap : E \to N$, that has a distinguished vertex s, called the ***source***, with nonzero outdegree, and a distinguished vertex t, called the ***sink***, with nonzero indegree.

D33: The ***maximum-flow problem*** is to determine the maximum flow that can be pushed through an s-t network from source s to sink t such that the flow into each intermediate node equals the flow out (*conservation of flow*) and the flow across any arc does not exceed the capacity of that arc. (See §11.1.)

D34: The ***minimum-cost-flow problem*** is to find an assignment of *flows* on the arcs of the flow network that satisfy the supply and demand (negative supply) requirements at minimum cost. (See §11.2.)

Software Testing and the Chinese Postman Problem

During execution, an application software's flow moves between various *states*, and the *transitions* from one state to another depend on the input. In testing software, one would like to generate input data that forces the program to test all possible transitions.

DEFINITIONS

D35: An ***eulerian tour*** of a digraph G is a closed directed walk that uses each arc *exactly* once.

D36: A ***postman tour*** (or ***covering walk***) is a closed directed walk that uses each arc *at least* once.

D37: Given a directed edge-weighted graph G, the ***Directed Chinese Postman Problem*** is to find a minimum-weight postman tour.

Digraph Model: The software's execution flow is modeled as a digraph, where the states of the program are represented by vertices, the transitions are represented by arcs, and each of the arcs is assigned a label indicating the input that forces the corresponding transition. Then the problem of finding an input sequence for which the program invokes all transitions and minimizes the total number of transitions is equivalent to the Directed Chinese Postman Problem, where all arc-weights equal one.

REMARKS

R6: Since certain transitions take more execution time than others, one might want to minimize the total time of execution during the testing (instead of the number of transitions). In that case, each arc is assigned a weight equal to the transition time corresponding to that arc.

R7: Under certain reasonable assumptions, the flow digraph modeling a program's execution can be assumed to be strongly connected, which guarantees the existence of a postman tour.

R8: Eulerian digraphs and graphs, along with algorithms to construct eulerian tours, are discussed in detail in §4.2, and various versions of the Chinese Postman Problem and its algorithms are discussed in §4.3.

Lexical Scanners

The source code of a computer program may be regarded as a string of symbols. A *lexical scanner* must scan these symbols, one at a time, and recognize which symbols go together to form a syntactic *token* or *lexeme.* We now consider a single-purpose scanner whose task is to recognize whether an input string of characters is a valid *identifier* in the C programming language. Such a scanner is a special case of a *finite-state recognizer* and can be modeled by a labeled digraph, as in Figure 3.1.10. One vertex represents the *start* state, in effect before any symbols have been scanned. Another represents the *accept* state, in which the substring of symbols scanned so far forms a valid C identifier. The third vertex is the *reject* state, indicating that the substring has been discarded because it is not a valid C identifier. Each arc label tells what kinds of symbols cause a transition from the tail state to the head state. If the final state after the input string is completely scanned is the accept state, then the string is a valid C identifier.

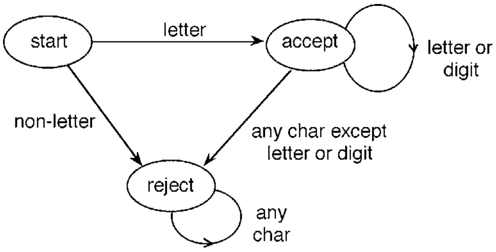


Figure 3.1.10 Finite-state recognizer for identifiers.

3.1.3 Binary Trees

At first glance, a discussion of binary trees does not seem to belong in a section on digraphs. In fact, binary trees are digraphs. In particular, they are special *rooted trees.* Here we describe a few applications.

Rooted Tree Terminology

DEFINITIONS

D38: In a rooted tree, the ***depth*** or ***level*** of a vertex v is its distance from the root, that is, the length of the unique path from the root to v. (Thus, the root has depth 0.)

D39: The ***height*** of a rooted tree is the length of a longest path from the root (which equals the greatest depth in the tree).

D40: If vertex v immediately precedes vertex w on the path from the root to w, then v is the ***parent*** of w and w is the ***child*** of v.

D41: A vertex w is called a ***descendant*** of a vertex v (and v is called an ***ancestor*** of w), if v is on the unique path from the root to w. If, in addition, $w \neq v$, then w is a ***proper descendant*** of v (and v is a ***proper ancestor*** of w).

D42: An ***ordered tree*** is a rooted tree in which the children of each vertex are assigned a fixed ordering.

D43: A ***standard plane representation*** of an ordered tree is a standard plane drawing of the tree such that at each level, the left-to-right order of the vertices agrees with their prescribed order.

D44: A ***binary tree*** is an ordered tree in which each vertex has at most two children, and each child is designated either a ***left-child*** or a ***right-child***.

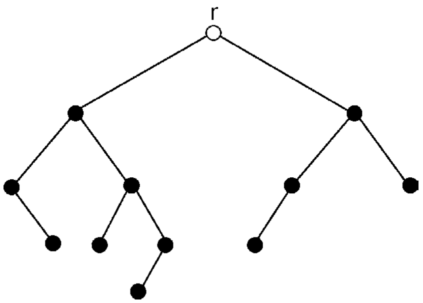


Figure 3.1.11 **A binary tree of height 4.**

D45: The ***left (right) subtree*** of a vertex v in a binary tree T is the binary subtree spanning the left (right)-child of v and all of its descendants.

FACT

F7: Every binary tree of height h has at most $2^{h+1} - 1$ vertices.

Binary Search

An entry in a *random-access table* consists of two fields. One field is for the actual data element, and the other one is for the *key*. An entry is found in a random-access table by searching for its key, and the most generally useful implementation of a random-access table uses the following information structure.

DEFINITIONS

D46: A ***binary-search tree*** (BST) is a binary tree, each of whose vertices is assigned a key, such that the key assigned to any vertex v is greater than the key at each vertex in the left subtree of v, and is less than the key at each vertex in the right subtree of v.

D47: A binary tree is ***balanced*** if for every vertex, the number of vertices in its left and right subtrees differ by at most one.

EXAMPLE

E9: Both of the binary-search trees in Figure 3.1.12 below store the keys:

$$3, 8, 9, 12, 14, 21, 22, 23, 28, 35, 40, 46$$

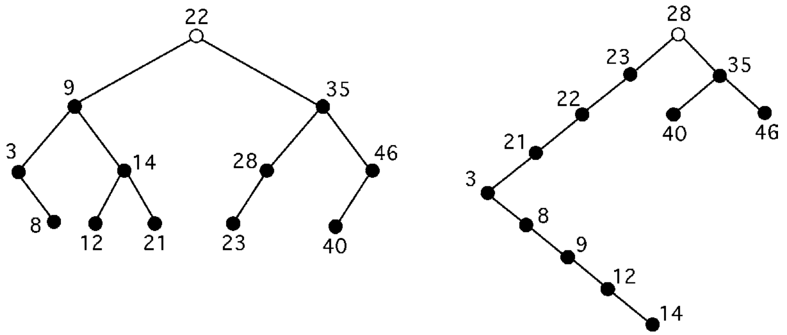


Figure 3.1.12 **A balanced binary-search tree and an unbalanced one.**

Algorithm 3.1.3: Binary-Search-Tree Search

Input: a binary-search tree T and a target key t.
Output: a vertex v of T such that $key(v) = t$ if t is found,
or NULL if t is not found.

```
$v := root(T)$
While ($v \neq$ NULL) and ($t \neq key(v)$)
     If $t > key(v)$
          $v := rightchild(v)$
     Else $v := leftchild(v)$
Return $v$.
```

COMPUTATIONAL NOTE: Since each comparison of a binary search performed on a binary-search tree moves the search down to the next level, the number of comparisons is at most the height h of the tree plus one. If the tree is balanced, then it is not hard to show that the number of vertices n is between 2^h and 2^{h+1}. Hence, the worst-case performance of the binary search on a perfectly balanced binary-search tree is $O(log_2 n)$. The other extreme occurs when each internal vertex of the binary tree has only one child.

Such a binary tree is actually an ordinary linked list, and therefore the performance of the search degenerates to $O(n)$.

References

[BaGe99] S. Baase and A. V. Gelder, *Computer Algorithms: Introduction to Design and Analysis*, Third Edition, Addison-Wesley, 1999.

[BaGu01] J. Bang-Jensen and G. Gutin, *Digraphs. Theory, Algorithms and Applications*, Springer-Verlag, London, 2001.

[ChLe96] G. Chartrand and L. Lesniak, *Graphs & Digraphs.* Third Edition, Chapman and Hall, London/New York, 1996.

[Ci75] E. Cinlar, *An Introduction to Stochastic Processes*, Prentice-Hall, 1975.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Ro39] H. E. Robbins, A theorem on graphs with an application to a problem of traffic control, *Amer. Math. Monthly* 46 (1939), 281–283.

[Ro76] F. Roberts, *Discrete Mathematical Models*, Prentice-Hall, 1976.

[Ro84] F. Roberts, *Applied Combinatorics*, Prentice-Hall, 1985.

[St93] H. J. Straight, *Combinatorics: An Invitation*, Brooks/Cole, 1993.

[ThSw92] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, John Wiley & Sons, 1992.

[Wa62] S. Warshall, A theorem of boolean matrices, *Journal of the ACM* 9 (1962), 11–12.

[We01] D. B. West, Introduction to Graph Theory, Second Edition, 2001, Prentice-Hall, (First Edition 1996).

[Wi94] W. L. Winston, Operations Research: Applications and Algorithms, Third Edition, Duxbury Press, 1994.

3.2 DIRECTED ACYCLIC GRAPHS

Stephen B. Maurer, Swarthmore College

3.2.1 Examples and Basic Facts
3.2.2 Rooted Trees
3.2.3 DAGs and Posets
3.2.4 Topological Sort and Optimization
References

Introduction

When a digraph has no directed cycles, it is called a directed acyclic graph, or a DAG. While being acyclic may seem to be a stringent condition, it arises quite naturally because vertices often have a natural ordering. For instance, vertices may represent events ordered in time or ordered by hierarchy. This ordering makes results and algorithms for DAGs relatively simple.

3.2.1 Examples and Basic Facts

DEFINITIONS

D1: A digraph is ***acyclic*** if it has no *directed* cycles.

D2: ***DAG*** is an acronym for directed acyclic graph.

D3: A ***source*** in a digraph is a vertex of indegree zero.

D4: A ***sink*** in a digraph is a vertex of outdegree zero.

D5: A ***basis*** of a digraph is a minimal set of vertices such that every other vertex can be reached from some vertex in this set by a directed path.

EXAMPLES

E1: *Operations Research*. A large project consists of many smaller tasks with a *precedence relation* — some tasks must be completed before certain others can begin. One graphical representation of such a project has a vertex for each task and an arc from u to v if task u must be completed before v can begin. For instance, in Figure 3.2.1, the food must be loaded and the cabin cleaned before passengers are loaded, but luggage unloading is independent of the timing of cabin activities. This model of a project will always be a DAG, because if there were a directed cycle, the project could not be done: every task on the cycle would have to be started before every other one on the cycle.

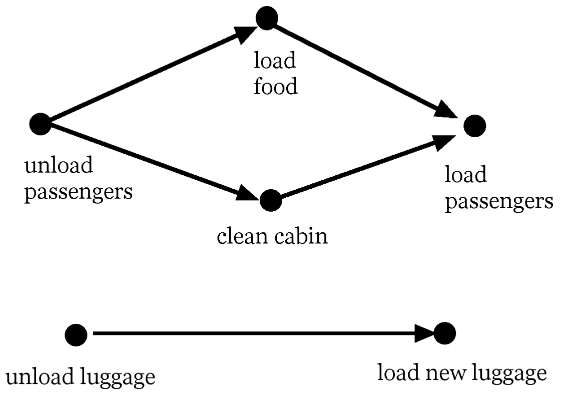


Figure 3.2.1 **A digraph of precedence in an airplane stopover.**

E2: *Sociology and Sociobiology*. A business (or army, or society, or ant colony) has a *hierarchical dominance* structure. The nodes are the employees (soldiers, citizens, ants) and there is an arc from u to v if u dominates v. If the chain of command is unique, with a single leader, and if only arcs representing immediate authority are included, then the result is a *rooted tree*, as in Figure 3.2.2. (Also see §3.2.2.)

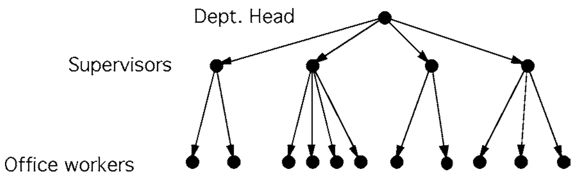


Figure 3.2.2 **A corporate hierarchy.**

E3: *Computer Software Design*. A large program consists of many subprograms, some of which can invoke others. Let the nodes of D be the subprograms, and let there be an arc from u to v if subprogram u can invoke subprogram v. Then this *call graph* D encapsulates all possible ways control can flow within the program. Must D be a DAG? No, but each directed cycle represents an indirect recursion and serves as a warning to the designer to ensure against infinite loops. See Figure 3.2.3, where Proc 2 can call itself indirectly. To determine if a digraph is a DAG or not, do a *topological sort* (§3.2.4).

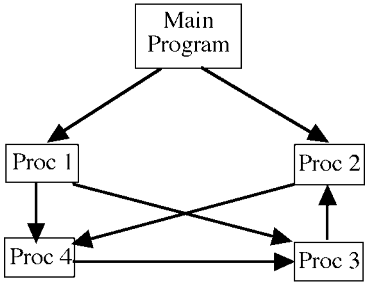


Figure 3.2.3 **The call graph of a computer program.**

E4: *Ecology*. A *food web* is a digraph in which nodes represent species and in which there is an arc from u to v if species u eats species v. Figure 3.2.4 shows a small food web. In general, food webs are acyclic, because animals tend to eat smaller animals or animals in some way "lower down" in the "food chain." The very fact that phrases like this are used indicates that there is a hierarchy, and thus no directed cycles.

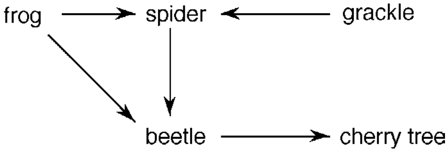


Figure 3.2.4 A small food web.

E5: *Genealogy.* A "family tree" is a digraph, where the orientation is traditionally given not by arrows but by the direction down for later generations. Despite the name, a family tree is usually not a tree, since people commonly marry distant cousins, knowingly or unknowingly. However, it is always a DAG, because if there were a cycle, everyone on it would be older than everyone else on the cycle.

E6: *State Diagrams.* Let the vertices of D be a set of states of some process, and let the arcs represent possible transitions. For instance, the process might be a board game, where the states are the configurations and each arc represents the transition of a single move. Then walks through D represent "histories" that the process/game can follow. If the game can never return to a previous configuration (e.g., as in tic-tac-toe), the state diagram of the game is a DAG.

FACTS

F1: Every DAG has at least one source and at least one sink.

F2: Every DAG has a unique basis, namely, the set of all its sources.

F3: Every subgraph of a DAG is a DAG.

F4: The transitive closure of a DAG is a DAG.

F5: A digraph is a DAG if and only if every walk in it is a path.

F6: A digraph is a DAG if and only if it is possible to order the vertices so that, in the adjacency matrix, all nonzero entries are above the main diagonal. (Topological sort in §3.2.4 finds the ordering.)

F7: The condensation of any digraph is a DAG. Figure 3.2.5 shows a digraph and its condensation.

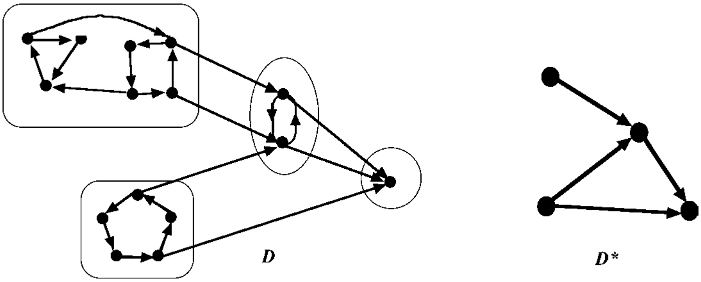


Figure 3.2.5 A digraph and its condensation.

F8: A digraph is a DAG if and only if it is isomorphic to its condensation.

F9: A digraph is strongly connected (unilateral, weakly connected) if and only if its condensation is strongly connected (unilateral, weakly connected).

F10: A DAG is never strongly connected, unless it consists of a single vertex.

F11: A DAG is unilateral if and only if it is a path.

F12: Every undirected graph without self-loops can be given an acyclic orientation, in fact, usually many. Namely, arbitrarily index the vertices as $v_1, v_2, \ldots, v_n$ and direct each edge from its lower indexed end to its higher indexed end.

REMARKS

R1: For more basic information on DAGs, see [Ha94, Ch. 16] and [Ro76, §2.2–3].

R2: Most of the acyclic orientations in Fact 12 are arbitrary and uninteresting, but occasionally an acyclic orientation is natural. In a tree, it is natural to orient edges away from a *root*; see §3.2.2. In a bipartite graph, it is natural to direct all edges from one side to the other. Still, most interesting orientations are already imposed by the nature of the problem, and the question is whether they are acyclic.

3.2.2 Rooted Trees

If the underlying graph of a digraph D is a tree, then D is certainly a DAG, because it doesn't even have any undirected cycles. However, the important tree DAGs have further restrictions on their edge directions.

For more on rooted trees, see [GrYe99, §3.2].

DEFINITIONS

D6: A ***directed tree*** is a digraph whose underlying graph is a tree.

D7: A ***rooted tree*** is a directed tree with a distinguished vertex r, called the ***root***, such that for every other vertex v, the unique path from r to v is a directed path from r to v.

CONVENTION: In drawing a rooted tree with the root marked, the arrows are usually omitted because the direction of each arc is always away from the root. In fact, if the direction is always down or left-to-right, as in Figure 3.2.6, it is not even necessary to indicate the root.

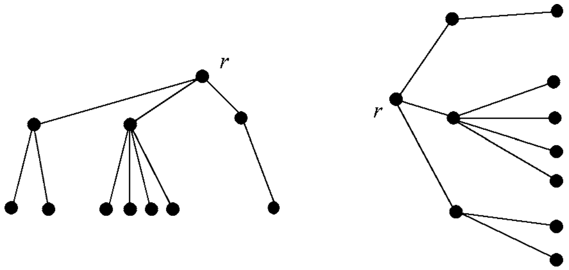


Figure 3.2.6 Two standard ways to draw a rooted tree.

D8: A rooted tree is also called an ***out-tree***. This alternative name is typically used when the arc directions are shown explicitly, for instance, when the tree is a spanning subgraph of a larger digraph.

D9: An ***in-tree*** is an out-tree with all the directions reversed, so that all paths are directed toward the root.

EXAMPLES

Previous Example 2 is about rooted trees. Here are some others.

E7: *Decision trees.* Any branching process leads to a rooted tree, where each node is a decision point, each arc from a node is an allowed decision, and the root is the start. For instance, the stages in a game may be represented this way. Figure 3.2.7 shows the first two moves in a game of tic-tac-toe, one by each player. Each node is represented by the way the board looks just *before* the decision. If we take into account symmetry, the figure is complete through the first two moves.

CONVENTION: In Figure 3.2.7 the two nodes on the bottom level (3rd move) illustrate that different nodes in the tree can represent the same state. While the board looks the same at these two nodes, the ordered sequence of decisions leading to these nodes are different. Thus in a decision tree, each node represents both a state and the complete history of how it was achieved. Compare with Example 6, where these nodes would be one, and the digraph would not be a tree.

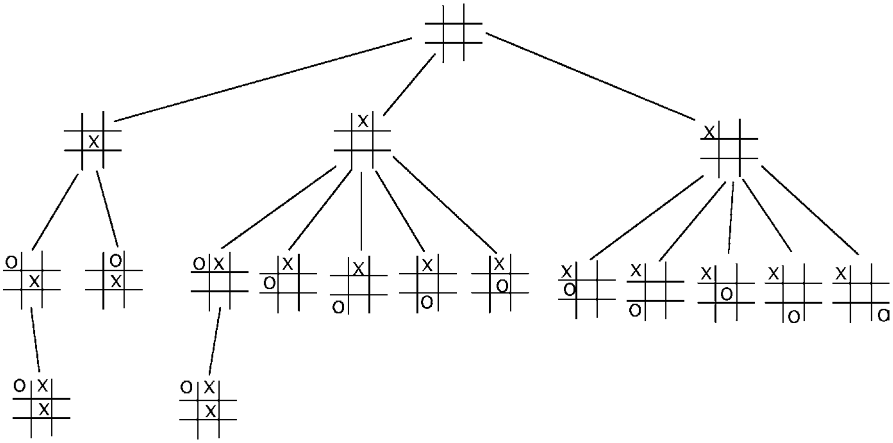


Figure 3.2.7 **The first two moves in the tic-tac-toe game tree, and a bit of the third level.**

E8: *Decomposition trees.* Any decomposition of an object or structure into finer and finer parts can be modeled with a rooted tree. Figure 3.2.8 shows an example of *sentence parsing*.

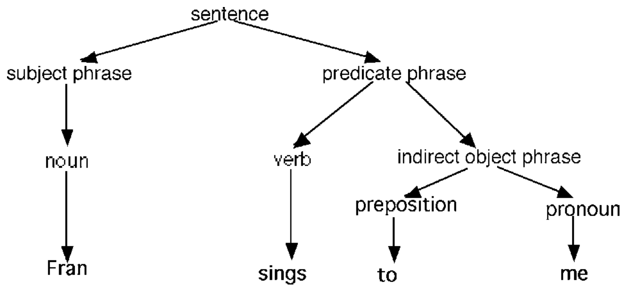


Figure 3.2.8 **A sentence parse tree.**

FACTS

F13: Every directed tree is a DAG.

F14: A digraph is a rooted tree if and only if its underlying graph is connected, exactly one vertex (the root) has indegree 0, and all others have indegree 1.

DEFINITIONS FOR ROOTED TREES

D10: The ***depth*** or ***level*** of a vertex v is its distance from the root, that is, the number of edges in the unique directed path from the root to v.

D11: The ***height*** of a rooted tree is the greatest depth of a vertex.

D12: If (u, v) is an edge, the u is the ***parent*** of v and v is the ***child*** of u.

D13: Vertices having the same parent are ***siblings***.

D14: If there is a directed path from vertex u to vertex v, then u is an ***ancestor*** of v and v is a ***descendant*** of u.

D15: A ***leaf*** is a vertex with outdegree 0 (no children).

D16: An ***internal vertex*** is a vertex that is not a leaf.

D17: An ***m-ary tree*** is a rooted tree in which every vertex has m or fewer children.

D18: A ***complete m-ary tree*** is an m-ary tree in which every internal vertex has exactly m children and all leaves are at the same level. See Figure 3.2.9.

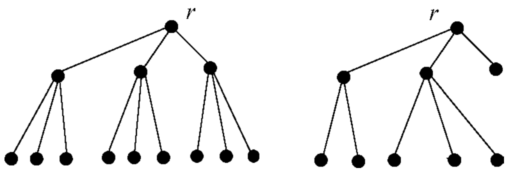


Figure 3.2.9 **Complete and incomplete ternary (3-ary) trees.**

D19: A ***ordered tree*** is a rooted tree in which the order of the children at each vertex makes a difference.

D20: A ***binary tree*** is an ordered 2-ary tree in which, even when a vertex has only one child, it makes a difference whether it is a ***left child*** or a ***right child***.

REMARKS

R3: Trees, rooted trees, ordered trees, and binary trees make finer and finer distinctions, which should only be used if the distinctions are important in the application being modeled. For instance, binary trees are used to model computations with binary operations, as in $3 \times (4/5)$. Since division is *noncommutative* ($4/5 \neq 5/4$), binary trees are an appropriate model for such computations.

R4: Figure 3.2.10 shows four graphs. As trees they are all the same (that is, *isomorphic*). However, as rooted trees, $G_1 = G_2$ and $G_3 = G_4$, so there are two rooted trees. There are three ordered trees, as G_1 and G_2 are still the same, but G_3, G_4 are different. Finally, as binary trees they are all different. In G_1, vertex c is a right child; in G_2 it is a left child.

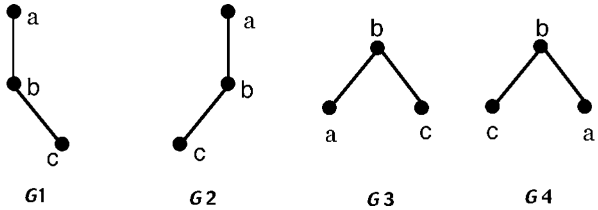


Figure 3.2.10 **Four trees: the same and not the same.**

FACTS

F15: An m-ary tree has at most m^k vertices at level k.

F16: Let T be an n-vertex m-ary tree of height h. Then

$$h + 1 \leq n \leq \frac{m^{h+1} - 1}{m - 1}.$$

The lower bound is attained if and only if T is a path. The upper bound is attained if and only if T is a complete m-ary tree.

Spanning Directed Trees

Since every connected graph has a spanning tree, every digraph has a spanning directed tree. In a graph, a spanning tree connects all the vertices, while using the minimum number of edges. However, in a digraph, a spanning directed tree may contain few directed *paths* and thus may allow fewer connections than the whole digraph does. So the more interesting question is whether a digraph has a spanning rooted tree. This question is answered algorithmically by the directed version of *depth first search*; see §10.1 and [GrYe99, §11.1]. It is answered algebraically by the directed matrix tree theorems; see §6.4. Here we simply state two key facts.

FACTS

F17: If digraph D has a spanning tree rooted at v, directed depth first search starting at v will find one.

F18: For every vertex of a digraph D there is a spanning tree rooted at that vertex if and only if D is strongly connected.

Functional Graphs

Closely related structurally to rooted trees, but devised for a different purpose, are functional graphs.

DEFINITION

D21: A ***functional graph*** is a digraph in which each vertex has outdegree one.

EXAMPLES

E9: For each function f from a finite domain U to itself, define a digraph D whose vertex set is U and for which (u, v) is an arc if and only if $f(u) = v$. By definition of a function, there is one such v for every $u \in U$. Hence, D is a functional graph (whence the name).

E10: Specifically, consider the doubling function on the positive integers, but consider only the effect on the ones digit. This function is completely described by its effect on the domain $\{0, 1, \ldots, 9\}$. Its functional graph is shown in Figure 3.2.11.

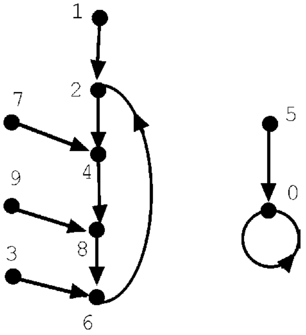


Figure 3.2.11 **The functional graph for doubling (mod 10).**

FACT

F19: Let D be a functional graph, and let G be the underlying undirected graph. Then each component of G contains exactly one cycle. In D this cycle is a directed cycle, and the removal of any arc in it turns that component into an in-tree.

3.2.3 DAGs and Posets

There is a very close connection between DAGs and posets. Every DAG represents a poset, and every poset can be represented by DAGs in several ways. For more information, see [Bo90, §7.1–2].

DEFINITIONS

D22: A ***partial order*** is a binary relation $\preceq$ on a set X that is

- *reflexive*: for all $x \in X$, $x \preceq x$;
- *antisymmetric*: for all $x, y \in X$, if $x \preceq y$ and $y \preceq x$, then $x = y$;
- *transitive*: for all $x, y, z \in X$, if $x \preceq y$ and $y \preceq z$, then $x \preceq z$.

D23: A ***poset***, or ***partially ordered set*** $P = (X, \preceq)$ is a pair consisting of a set X, called the ***domain***, and a partial order $\preceq$ on X.

D24: Elements x, y of P are ***comparable*** if either $x \preceq y$ or $y \preceq x$.

D25: Element x is ***less than*** element y, written $x \prec y$, if $x \preceq y$ and $x \neq y$.

D26: The ***comparability digraph*** of the poset $P = (X, \preceq)$ is the digraph with vertex set X such that there is an arc from x to y if and only if $x \preceq y$.

D27: The element y ***covers*** the element x in a poset if $x \prec y$ and there is no element z such that $x \prec z \prec y$.

D28: The ***cover graph*** of a poset $P = (X, \preceq)$ is the graph with vertex set X such that x, y are adjacent if and only if one of them covers the other.

D29: A ***Hasse diagram*** of poset P is a straight-line drawing of the cover graph such that the lesser element of each adjacent pair is lower in the drawing.

EXAMPLE

E11: Let $X = \{2, 4, 5, 8, 10, 20\}$ and let $\preceq$ be the *divisibility relation* on X. That is $x \preceq y$ if and only if y/x is an integer. The comparability digraph and the Hasse diagram for $P = (X, \preceq)$ are as shown in Figure 3.2.12.

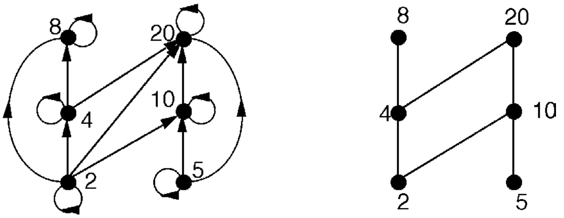


Figure 3.2.12 **Comparability digraph and Hasse diagram for a poset.**

FACTS

F20: If the loops are deleted, the comparability digraph of any poset is a DAG.

F21: Every Hasse diagram is a DAG if one considers all edges to be directed up (or all down).

F22: Every DAG D represents a poset in the following sense. The domain of P is the vertex set of D, and $x \preceq y$ if there is a directed path from x to y.

TERMINOLOGY NOTE: In passing from DAG D to poset P, null paths are included, so that $x \preceq x$ for all x. Alternatively, we obtain the poset by taking the transitive closure D^* of D. Then $x \prec y$ if and only if (x, y) is an arc of D^*.

3.2.4 Topological Sort and Optimization

In a DAG, the vertices can always be numbered consecutively so that all arcs go from lower to higher numbers. Using this numbering, many optimization problems can be solved by essentially the same algorithm, one that makes a single pass through the vertices in numbered order. For more general digraphs, algorithms for these optimization problems are less efficient or at least more complicated to describe.

DEFINITIONS

D30: A ***linear extension ordering*** of a digraph is a consecutive numbering of the vertices as $v_1, v_2, \ldots, v_n$ so that all arcs go from lower-numbered to higher-numbered vertices.

D31: A ***topological sort***, or ***topsort***, is any algorithm that assigns a linear extension ordering to a digraph when it has one. (This name is traditional, but the relation to topology in the sense understood by topologists is obscure.) A simple topological sort algorithm is shown as Algorithm 3.2.1. See also [Ro84, §11.6.2].

FACTS

F23: A digraph has a linear extension ordering if and only if it is a DAG.

F24: Topological sort determines if a digraph is a DAG and finds a linear extension ordering if it is.

Algorithm 3.2.1: Topological sort

Input: a digraph D.
Output: A linear extension ordering if D is a DAG; failure otherwise

```
H := D; k = 1
while V_H ≠ φ     {vertex set of H non-empty}
   v_k := any vertex in H of indegree zero.
      {If no such vertex exists, exit: D is not a DAG}
   H := H − v_k     {New H is a DAG if old H was}
   k := k+1
```

REMARK

R5: Because of the close connection between DAGs and posets, this whole discussion of linear extensions and topological sort can just as well be stated in the poset context. For instance, every poset has a linear extension, which may be found by a topological sort. See [GrYe99, pp. 373–376].

Optimization

There are many computational problems about graphs, with important real-world applications, when the graphs have *weights* on their vertices and/or edges. For DAGs, many of these problems can be solved by essentially the same single-pass algorithm. This algorithm is the basic form of the sort of staged algorithm called *dynamic programming* in operations research circles [HiLi95, Ch. 10]. Algorithms 3.2.2 and 3.2.3 provide templates for two versions of this algorithm. The examples that follow fill in the templates by giving specific formulas for updating the functions they compute.

In Algorithm 3.2.2, topsort is done first, and then the function F is computed vertex by vertex in topsort order. In Algorithm 3.2.3, the topsort is done simultaneously with improving F on vertices not yet sorted.

Algorithm 3.2.2: Basic Dynamic Programming, First Version

Input: DAG D with vertices numbered $v_1, v_2, \ldots, v_n$ in topsort order; weights $w(v)$ on vertices or $w(v,u)$ on arcs, as needed.

Output: Correct values of desired function F.

```
Initialize $F(v_1)$
For $k = 2$ to $n$
    Determine $F(v_k)$ in terms of weights and $F(v_i)$ for $i < k$.
```

Algorithm 3.2.3: Basic Dynamic Programming, Second Version

Input: DAG D with n vertices and weights $w(v)$ on vertices or $w(v,u)$ on arcs, as needed.

Output: Correct values of desired function F.

```
Initialize $F(v)$ for all $v$.
$H := D$
For $k = 1$ to $n$
    $v_k :=$ a source in $H$    {exists since $H$ is a DAG}
    Update $F(u)$ for all $u$ for which $(v_k, u)$ is an edge in $H$.
    $H := H - v_k$
```

EXAMPLES

For simplicity in the formulas, in all examples below we assume that the DAGs have no multiple edges.

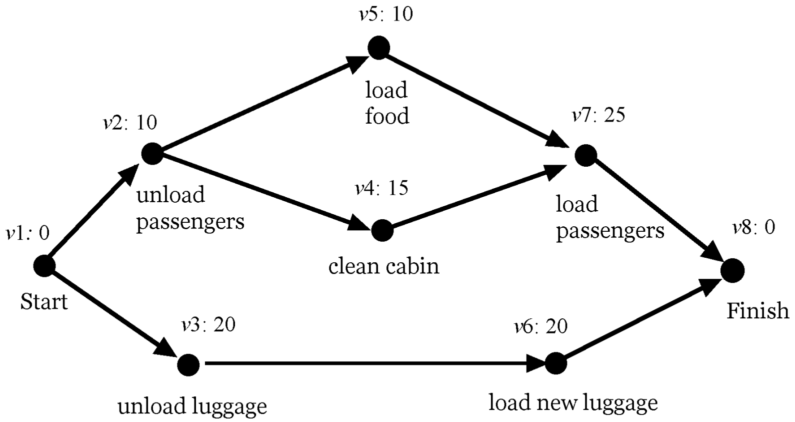


Figure 3.2.13 Airplane stopover as CPM graph.

E12: *Project Scheduling*. Consider Figure 3.2.13, which repeats Figure 3.2.1 with the following additions: Start and Finish vertices, a topsort ordering, and times for the tasks as weights on the vertices. Start and Finish, being merely marker vertices, take time 0. Recall that (u,v) is an arc if task u must be completed directly before task v begins, and that these tasks are the steps necessary to complete an airplane stopover. How quickly can the stopover be completed? The bottleneck is the *longest path* from

Start to Finish, where the length of a (directed) path is the sum of the weights on its vertices. Dynamic programming can answer this question as follows. Let

$$F(u) = \text{the length of the longest path (using vertex weights) from Start to } u.$$

Then in Algorithm 3.2.2 use

$$\begin{aligned} \text{Initialization:} &\quad F(v_1) = w(v_1) = 0, \quad (\text{Note: } v_1 = \text{Start}) \\ \text{Update:} &\quad F(v_k) = w(v_k) + \max\{F(v_i) \mid (v_i, v_k) \text{ is an arc}\}. \end{aligned}$$

In Algorithm 3.2.3 use

$$\begin{aligned} \text{Initialization:} &\quad \text{For all } v,\ F(v) = w(v), \\ \text{Update:} &\quad \text{For all } u \text{ such that } (v_k, u) \text{ is an arc, } F(u) = \max\{F(u),\ F(v_k){+}w(u)\}. \end{aligned}$$

For either algorithm, at termination the desired answer is $F(\text{Finish})$, that is, $F(v_n)$.

This method of finding the optimal schedule by iteratively finding the longest path is the essence of the *critical path method*, or *CPM* [HiLi95, Ch. 9]. This example uses the *activity on node* model, or *AoN*. See Example 13 for the *activity on arc* model, or *AoA*.

E13: *Project Scheduling, second model.* If *edges* represent subtasks, and tasks earlier on directed paths must be completed before those later are begun, then the longest path from the Start to Finish vertex is the shortest time in which the whole project can be completed, where now the length of a path is the sum of the weights on its edges. Let

$$F(u) = \text{the length of the longest path (using edge weights) from Start to } u.$$

Then in Algorithm 3.2.2 use

$$\begin{aligned} \text{Initialization:} &\quad F(v_1) = 0, \\ \text{Update:} &\quad F(v_k) = \max\{F(v_i) + w(v_i, v_k) \mid (v_i, v_k) \text{ is an arc}\}. \end{aligned}$$

In Algorithm 3.2.3 use

$$\begin{aligned} \text{Initialization:} &\quad \text{For all } v,\ F(v) = 0, \\ \text{Update:} &\quad \text{For all } u \text{ such that } (v_k, u) \text{ is an arc,} \\ &\quad F(u) = \max\{F(u),\ F(v_k){+}w(v_k, u)\}. \end{aligned}$$

For either algorithm, at termination the desired answer is $F(\text{Finish})$.

E14: *Shortest Paths.* What is the shortest directed path between two vertices u and u', where the length of a path is the sum of the weights on its edges? If a graph represents a road network, and the weights on the edges are the lengths of the road segments (or the travel times, or the toll on that segment), then shortest path means the shortest road distance (or least time, or lowest toll). If the graph is a DAG, and we make u the Start vertex (by eliminating earlier vertices in the topsort if necessary), then dynamic programming finds the shortest path as follows. Let

$$F(u) = \text{the length of the shortest path (using edge weights) from Start to } u.$$

Then in Algorithm 3.2.2 use

$$\begin{aligned} \text{Initialization:} &\quad F(v_1) = 0, \\ \text{Update:} &\quad F(v_k) = \min\{F(v_i) + w(v_i, v_k) \mid (v_i, v_k) \text{ is an arc}\}. \end{aligned}$$

In Algorithm 3.2.3 use

$$\begin{aligned}\text{Initialization:}\quad & F(u)=0,\ F(v)=\infty \text{ for } v\neq u,\\ \text{Update:}\quad & \text{For all } v \text{ such that } (v_k,v) \text{ is an arc,}\\ & F(v)=\min\{F(v),\ F(v_k){+}w(v_k,v)\}.\end{aligned}$$

For either algorithm, at termination the desired answer is the value of $F(u')$.

E15: What is the shortest directed path between two vertices, where the length of a path is the sum of the weights on its *vertices*? Dynamic programming solves this problem too for DAGs, with a slight change in the formulas in Example 14 (replace edge weights with vertex weights).

E16: *Counting Paths.* How many directed paths are there between a given pair of vertices? If the digraph is a DAG, and the vertices are Start and Finish, let

$$F(u) = \text{the number of directed paths from Start to } u.$$

Then in Algorithm 3.2.2 use

$$\begin{aligned}\text{Initialization:}\quad & F(v_1)=1,\ (v_1=\text{Start})\\ \text{Update:}\quad & F(v_k)=\textstyle\sum\{F(v_i)\mid (v_i,v_k) \text{ is an arc}\}.\end{aligned}$$

In Algorithm 3.2.3 use

$$\begin{aligned}\text{Initialization:}\quad & F(\text{Start})=1,\ F(v)=0 \text{ for } v\neq \text{Start},\\ \text{Update:}\quad & \text{For all } v \text{ such that } (v_k,v) \text{ is an arc,}\\ & F(v)=F(v)+F(v_k).\end{aligned}$$

For either algorithm, at termination the desired answer is the value of $F(\text{Finish})$.

E17: *Maximin Paths.* What is the directed path between two vertices for which the minimum edge weight on that path is maximum among all paths between those two vertices? This is called the *maximin path* and that maximum value is called the *maximin value.* In Figure 3.2.14 the maximin path from v_1 to v_6 is $v_1v_3v_4v_6$ and the maximin value is 4. If the edges represent railroad segments, and each edge weight is the weight limit on that railroad segment, then this is the path between the two points over which the heaviest load can be shipped.

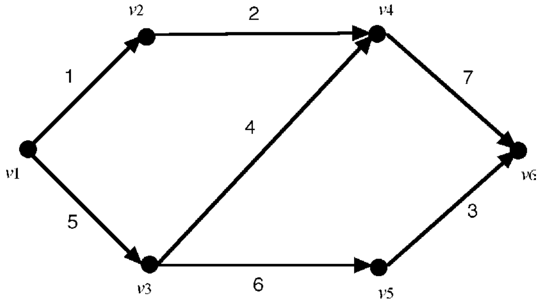


Figure 3.2.14 **The maximin path $v_1v_3v_4v_6$ has value 4 and the minimax path $v_1v_3v_5v_6$ has value 6.**

If the digraph is a DAG, and the vertices are Start and Finish, let

$$F(u) = \text{the maximin value for directed paths from from Start to } u.$$

Then in Algorithm 3.2.2 use

$$\text{Initialization:}\quad F(v_1) = 0,\ (v_1 = \text{Start})$$

$$\text{Update:}\quad F(v_k) = \max\{\min\{F(v_i), w(v_i, v_k)\} \mid (v_i, v_k) \text{ is an arc}\}.$$

In Algorithm 3.2.3 use

$$\text{Initialization:}\quad F(\text{Start}) = 0,\ F(v) = \infty \text{ for } v \neq \text{Start},$$

$$\text{Update:}\quad \text{For all } v \text{ such that } (v_k, v) \text{ is an arc,}$$

$$F(v) = \max\big\{F(v),\ \min\{F(v_k), w(v_k, v)\}\big\}.$$

For either algorithm, at termination the desired answer is the value of $F(\text{Finish})$.

E18: *Minimax Paths.* What is the directed path between two vertices for which the maximum edge weight on the path is minimum? This *minimax* question is relevant if the graph represents a pipeline network, and each edge weight is the maximum elevation on that segment, because the work necessary to push a fluid through a pipeline route is related to the maximum height to which the fluid must be raised along the way. In Figure 3.2.14 the minimax path from v_1 to v_6 is $v_1v_3v_5v_6$ and the minimax value is 6. Dynamic programming solutions to the minimax problem are found by interchanging the roles of min and max in the algorithms for Example 17. Also, in Algorithm 3.2.3, all $F(v)$ are initialized to 0.

FACTS

F25: Algorithms 3.2.2–3 each solve critical path problems and many other optimization and computation problems on DAGs. (See the examples above.)

F26: In project scheduling problems modeled by DAGs, the minimum completion time is the length of the longest path from the Start node to the Finish node.

F27: Any DAG may be augmented to have just one source and one sink (just create a new node named Start adjacent *to* all existing sources, and a new node named Finish adjacent *from* all existing sinks).

References

[Bo90] K. Bogart, *Introductory Combinatorics*, Harcourt Brace Jovanovich, 1990.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Ha94] F. Harary, *Graph Theory*, Perseus, 1994 (reprint of original edition, Addison Wesley, 1969).

[HiLi95] F. Hillier and G. Lieberman, *Introduction to Operations Research*, Sixth Edition, McGraw-Hill, 1995.

[Ro76] F. Roberts, *Discrete Mathematical Models*, Prentice-Hall, 1976.

[Ro84] F. Roberts, *Applied Combinatorics*, Prentice-Hall, 1985.

3.3 TOURNAMENTS

K. B. Reid, California State University San Marcos

3.3.1 Basic Definitions and Examples
3.3.2 Paths, Cycles, and Connectivity
3.3.3 Scores and Score Sequences
3.3.4 Transitivity, Feedback Sets, Consistent Sets of Arcs
3.3.5 Kings, Oriented Trees, and Reachability
3.3.6 Domination
3.3.7 Tournament Matrices
3.3.8 Voting
References

Introduction

Tournaments comprise a large and important class of directed graphs. Application areas in which tournaments arise as models include round-robin tournaments (hence the name), paired-comparison experiments, domination in some animal societies, majority voting, population ecology, and communication networks. Many early results were motivated by applications; more recently, much focus has been on the combinatorial structure of tournaments as a separate area of graph theory. J. W. Moon's excellent monograph [Mo68] contains most of the results on tournaments up to 1968. In large part because of the influence of that work, tournament theory has so flourished during the past 35 years that subsequent surveys covered only a fraction of the results available. However, these surveys remain good sources for results about tournaments and directed graphs related to tournaments (see [HaNoCa65], [HaMo66], [BeWi75], [ReBe79], [Be81], [ZhSo91], [Gu95], [BaGu96], and [Re96]). Much work has been done on generalizations and extensions of tournaments (see [BaGu98]). A good source for digraphs in general, with extensive coverage of tournaments and various generalizations, is the book by Bang-Jensen and Gutin [BaGu01].

3.3.1 Basic Definitions and Examples

NOTATION: An arc from vertex x to vertex y will be denoted (x, y) or by $x \to y$.

DEFINITIONS

D1: A ***tournament*** is an oriented complete graph, i.e., there is exactly one arc between every pair of distinct vertices (and no loops).

D2: The ***order of a tournament*** T is the number of vertices in T. A tournament of order n will be called an ***n-tournament***.

D3: A vertex x in a tournament T ***dominates*** (or ***beats***) vertex y in T whenever (x, y) is an arc of T. We also say that y is ***dominated*** (or beaten) by x.

D4: A vertex that dominates every other vertex in a tournament is called a ***transmitter***. A vertex that is dominated by every other vertex in a tournament is called a ***receiver***.

D5: The ***score*** (or ***out-degree***) of a vertex v in a tournament T is the number of vertices that v dominates. It is denoted by $d_T^+(v)$. Note that if the tournament T under consideration is clear from the context, then T will be dropped and the score of v will be denoted $d^+(v)$. The ***in-score*** (or ***in-degree***) of a vertex u in a tournament T is the number of vertices that dominate u. It is denoted $d_T^-(u)$ (or $d^-(u)$).

D6: The ***score sequence*** (or ***score vector***) of an n-tournament T is the ordered n-tuple $(s_1, s_2, \ldots, s_{n-1}, s_n)$, where s_i is the score of vertex v_i, $1 \le i \le n$, and

$$s_1 \le s_2 \le \ldots \le s_{n-1} \le s_n$$

D7: A tournament is ***reducible*** if its vertex-set can be partitioned into two non-empty subsets V_1 and V_2 such that every vertex in V_1 dominates every vertex in V_2. A tournament that is not reducible is said to be ***irreducible.***

D8: The ***out-set*** of a vertex x in a digraph D, denoted $O(x)$, is the set of all vertices that x dominates, and the ***in-set*** of x, denoted $I(x)$, is the set of all vertices that dominate x.

TERMINOLOGY: In a digraph D, the out-set of a vertex x is also called the ***neighborhood*** of x, denoted $N_D^+(x)$ (or $N^+(x)$ if D is understood).

FACTS

F1: There are $2^{\binom{n}{2}}$ different labeled n-tournaments using the same n distinct labels, since for each pair of distinct labels $\{a, b\}$, either the vertex labeled a dominates the vertex labeled b or b dominates a.

F2: [Da54] The number $t(n)$ of non-isomorphic (unlabeled) n-tournaments is given by a rather complicated formula involving a summation over certain partitions of n. Moreover,

$$t(n) > \frac{2^{\binom{n}{2}}}{n!} \quad \text{and} \quad \lim_{n\to\infty} \frac{t(n)}{2^{\binom{n}{2}} \Big/ n!} = 1$$

The first few values of $t(n)$ are given by

n	1	2	3	4	5	6	7	8	9	10
$t(n)$	1	1	2	4	12	56	456	6880	191536	9733056

EXAMPLES

E1: Tournaments of orders 1 through 4 are illustrated in Figure 3.3.1.

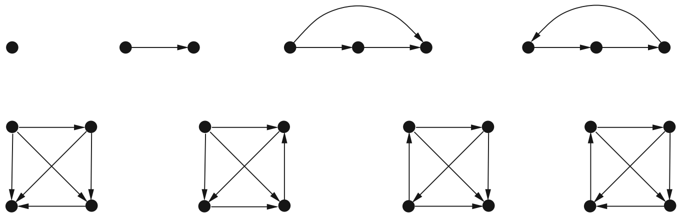


Figure 3.3.1 Tournaments of orders 1 through 4.

E2: *Tournament Scheduling.* To minimize the number of days of an n-tournament, games are scheduled in parallel. If n is even, then at least $n-1$ rounds are needed (since at most $n/2$ of the $n(n-1)/2$ games may be played at once). If n is odd, then at least n rounds are needed. There are several scheduling methods that achieve these compact schedules (see [Mo68]). Round-robin sports tournament scheduling and its relationship to edge-coloring is discussed in §5.5. Another related topic, *factors and factorization*, is discussed in §5.3.

Regular Tournaments

DEFINITIONS

D9: A ***regular tournament*** is a tournament T in which all scores are the same (i.e., there is an integer s so that $d^+(v) = s$ for all vertices $v \in V(T)$). An ***almost regular*** (or ***near regular***) ***tournament*** is a tournament T in which $\max\limits_{v \in V(T)} \{|d^+(v) - d^-(v)|\} = 1$.

D10: A ***doubly-regular tournament*** is a tournament in which all pairs of vertices jointly dominate the same number of vertices (i.e., there is an integer k so that $|O(x) \cap O(y)| = k$, for all distinct pairs of vertices x and y in T).

D11: Let G be an abelian group of odd order $n = 2m + 1$ with identity 0. Let S be an m-element subset of $G - \{0\}$ such that for every $x, y \in S$, $x + y \neq 0$. That is, choose exactly one element from each of the m 2-sets of the form $\{x, -x\}$, where x ranges over all $x \in G - \{0\}$. Form the digraph D with vertex-set $V(D) = G$ and arc-set $A(D)$ defined by: arc $(x, y) \in A(D)$ if and only if $y - x \in S$. Then D is called a ***rotational tournament*** with ***symbol set*** S and is denoted $R_G(S)$, or simply $R(S)$ if the group G is understood.

D12: Let $G = GF(p^k)$ be the finite field with p^k elements, where p is a prime, $p \equiv 3$ (modulo 4), and k is an odd positive integer, and let S be the set of elements that are multiplicative squares of G (called the *quadratic residues*). Then the rotational tournament $R_G(S)$ is called a ***quadratic residue tournament***.

FACTS

F3: The rotational tournament $R_G(S)$, where $|G| = n$, is a regular n-tournament.

F4: [ReBe79] If T is a doubly-regular n-tournament, then T is regular and n ≡ 3 (modulo 4). Moreover, there exists a doubly-regular (4k + 3) – tournament if and only if there exists a (4k + 4) by (4k + 4) matrix H of +1's and −1's such that HH^t = (4k + 4)I and H + H^t = 2I, where I is the identity matrix (such an H is called a skew-Hadamard matrix) [ReBr72].

REMARK

R1: Frequently, the group G for the rotational tournament $R_G(S)$ is taken to be Z_n, the integers modulo $n = 2m + 1$.

EXAMPLES

E3: The 9-tournament shown in Figure 3.3.2 is regular since every vertex has score 4, and it is also irreducible. Moreover, it is the rotational tournament $R_G(S)$, where $G = Z_9$ and $S = \{2, 4, 6, 8\}$.

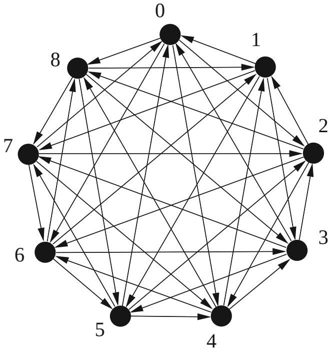


Figure 3.3.2 **The regular, rotational tournament $R(\{2, 4, 6, 8\})$.**

E4: The regular 7-tournament shown in Figure 3.3.3 is the quadratic residue tournament $R_G(S)$, where $G = GF(7)$ and $S = \{1, 2, 4\}$. Observe that it is irreducible, and it is a doubly-regular tournament since $|O(x) \cap O(y)| = 1$ for all distinct pairs of vertices x and y. The quadratic residue 7-tournament is notorious in tournament theory due to its occurrence as an exception to many results on tournaments.

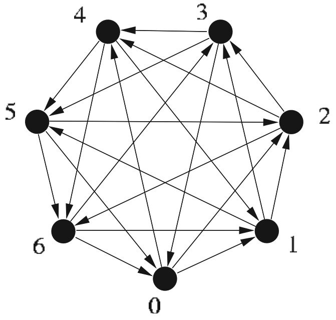


Figure 3.3.3 $R(\{1, 2, 4\})$.

Arc Reversals

Any n-tournament can be transformed into any other n-tournament by a sequence of reversals of arcs.

FACTS

F5: [Ry63] If T and W are two n-tournaments with the same score sequence, then T can be transformed into an isomorphic copy of W by a properly chosen sequence of reversals of arcs in 3-cycles.

F6: [Re73] If T and W are two n-tournaments and k is a fixed integer with $1 \leq k \leq n-1$, then T can be transformed into an isomorphic copy of W by a properly chosen sequence of reversals of arcs in k-paths.

REMARK

R2: C. Thomassen [Th88] extended these results to pairs of tournaments with the same vertex-set. He also described an infinite class of non-tournament digraphs for which the reversal of no arc decreases the total number of cycles. However, little progress has been made on the corresponding tournament problem (see [Re84]).

CONJECTURE

Ádám's Conjecture [Ád64]**:** Every non-transitive tournament contains at least one arc whose reversal reduces the total number of cycles.

3.3.2 Paths, Cycles, and Connectivity

Paths and cycles are fundamental sub-structures in tournaments and have been well studied in tournament theory. Many more results than given here have been collected by Bang-Jensen and Gutin in their survey [BaGu96] and their book [BaGu01].

NOTATION: All paths in this section are directed unless specified otherwise.

DEFINITIONS

D13: A ***hamiltonian path*** (or ***spanning path***) in a digraph D is a path that includes all vertices of D. A ***hamiltonian cycle*** (or ***spanning cycle***) in a digraph D is a cycle that includes all vertices of D. (Hamiltonian paths and cycles are discussed in §4.5.)

D14: A digraph D is ***strong*** (or ***strongly connected***) if for every pair of distinct vertices x and y of D, there is a path from x to y and a path from y to x.

EXAMPLE

E5: The two tournaments in Figures 3.3.2 and 3.3.3 are strong and irreducible. For example, the hamiltonian cycle in $R(\{2,4,6,8\})$ given by

$$0 \to 2 \to 6 \to 3 \to 5 \to 4 \to 8 \to 1 \to 7 \to 0$$

implies that for every pair of distinct vertices x and y, there is a path from x to y and a path from y to x.

REMARK

R3: Fact 7 is perhaps the most fundamental result about tournaments and is used frequently in their study. The first part has several inductive proofs.

FACTS

F7: [Ré34] Every tournament contains a hamiltonian path. Moreover, every tournament contains an odd number of hamiltonian paths.

F8: The following four statements are equivalent for any n-tournament T:
(a) T is strong.
(b) T is irreducible.
(c) T contains a hamiltonian cycle [Ca59].
(d) For every vertex x of T and for every integer k, $3 \leq k \leq n$, x is contained in a cycle of length k [Mo68]. (See also [HaMo66].)

F9: [Ga72] A curious fact: the number of n-tournaments containing a unique hamiltonian cycle is equal to the $(2n-6)^{\text{th}}$ Fibonacci number.

F10: [MoMo62] The fraction of labeled n-tournaments that are strong approaches 1 as $n \to \infty$.

F11: There is an $O(n^2)$ algorithm for finding a hamiltonian path in a tournament, and there is an $O(n^2)$ algorithm for finding a hamiltonian cycle in a tournament. (See [BaGu01] and [Ma92]).)

F12: [Vo02] Every arc of a strongly connected n-tournament is contained in a path of length

$$\lceil (n+3)/2 \rceil - 1.$$

Condensation and Transitive Tournaments

DEFINITIONS

D15: If T is a tournament with vertex partition $\{V_1, V_2, \ldots, V_k\}$, where each V_i induces a maximal strongly connected sub-tournament of T, then the ***condensation tournament*** of T, denoted T^*, is the k-tournament with vertex-set $\{u_1, u_2, \ldots, u_k\}$ and in which u_i dominates u_j whenever all of the vertices in V_i dominate all of the vertices in V_j in T.

D16: A tournament T is ***transitive*** if for all three distinct vertices x, y, and z in T, if x dominates y, and y dominate z, then x dominates z.

EXAMPLE

E6: Consider the 9-tournament T consisting of three vertex-disjoint 3-cycles A_1, A_2, A_3, in which every vertex of A_1 dominates every vertex of A_2 and every vertex of A_3, and every vertex of A_2 dominates every vertex of A_3. The vertex partition of T in Definition 15 is $V(A_1) \cup V(A_2) \cup V(A_3)$, and T^* is the transitive 3-tournament with vertex-set $\{u_1,\ u_2,\ u_3\}$, where u_1 dominates u_2 and u_3, and u_2 dominates u_3.

FACTS

F13: [HaNoCa65] The condensation T^* of a tournament T is a transitive tournament.

F14: The following five statements are equivalent for an n-tournament. See [Mo68] for references.
(a) T is transitive.
(b) T contains no cycles.
(c) T contains a unique hamiltonian path.
(d) T has score sequence $(0, 1, 2, 3, \ldots, n-2, n-1)$.
(e) The vertices of T can be labeled $v_1, v_2, v_3, \ldots, v_{n-1}, v_n$ so that v_i dominates v_j if and only if $1 \leq i < j \leq n$ (i.e., T is a complete [linear] order).

F15: Every (2^{n-1})-tournament contains a transitive sub-tournament of order n.

Cycles and Paths in Tournaments

FACTS

F16: [Al67] Every arc in a regular n-tournament, $n \geq 3$, is in cycles of all lengths m, $3 \leq m \leq n$. (See [Th80] for extensions.)

F17: [Ja72] Every arc in an almost regular n-tournament, $n \geq 8$, is in cycles of all lengths m, $4 \leq m \leq n$. (See [Th80] for extensions.)

F18: [AlReRo74] For every arc (x, y) of a regular n-tournament T, where $n \geq 7$, and for every integer m, $3 \leq m \leq n-1$, T contains a path of length m from x to y. (See [Th80] for extensions.)

F19: [Th80] For every arc (x, y) of an almost regular n-tournament T, where $n \geq 10$, and for every integer m, $3 \leq m \leq n-1$, T contains a path of length m from x to y. (See also [GuVo97].)

Hamiltonian Cycles and Kelly's Conjecture

CONJECTURE

Kelly's conjecture (see [Mo68]). The arc-set of a regular n-tournament can be partitioned into $(n-1)/2$ subsets, each of which induces a hamiltonian cycle.

EXAMPLE

E7: The arc-set of the quadratic residue rotational 7-tournament $R(\{1, 2, 4\})$ can be decomposed into 3 hamiltonian cycles:

$$0 \to 1 \to 2 \to 3 \to 4 \to 5 \to 6 \to 0;$$

$$0 \to 2 \to 4 \to 6 \to 1 \to 3 \to 5 \to 0;$$

$$0 \to 4 \to 1 \to 5 \to 2 \to 6 \to 3 \to 0.$$

REMARK

R4: Kelly's conjecture has stimulated much work in tournament theory. Evidence for the conjecture includes: it is true for $n \leq 9$ (B. Alspach, see [BeTh81]); every n-tournament, $n \geq 5$, contains two arc-disjoint hamiltonian cycles [Zh80]; regular or almost regular n-tournaments contain at least $\lfloor\sqrt{n/1000}\rfloor$ arc-disjoint hamiltonian cycles [Th82]. The best published result to date is the next result. A covering result then follows.

FACTS

F20: [Hä93] Then there exists a positive constant c, $c \geq 2^{-18}$, so that each regular n-tournament contains at least cn arc-disjoint hamiltonian cycles.

F21: [Th85] Each regular n-tournament T contains $12n$ hamiltonian cycles so that each arc of T is in at least one of the cycles.

Higher Connectivity

DEFINITION

D17: D is ***k*-strong** (or ***k*-strongly connected**) if for every subset S of $k-1$ or fewer vertices of D, $D-S$ is a strong digraph.

FACTS

F22: [Th80] Every arc in a 3-strong tournament is contained in a hamiltonian cycle. Moreover, this is false for infinitely many 2-strong tournaments. For every pair of distinct vertices x and y in a 4-strong tournament there is a hamiltonian path from x to y and there is a hamiltonian path from y to x. Moreover, this is false for infinitely many 3-strong tournaments.

F23: [FrTh87] If T is a k-strong tournament and B is any set of $k-1$ or fewer arcs of T, then the arc-deletion digraph $T-B$ contains a hamiltonian cycle.

F24: [Th84] There is a function h so that given any k independent arcs, $a_1, a_2, \ldots, a_k$, in an $h(k)$-connected tournament T, there is a hamiltonian cycle in T containing $a_1, a_2, \ldots, a_k$ in cyclic order.

F25: [So93] For any integer m, $3 \leq m \leq n-3$, every 2-strong n-tournament T, $n \geq 6$, contains two vertex-disjoint cycles of lengths m and $n-m$, unless T is isomorphic to the quadratic residue rotational tournament $RT(1,2,4)$. (This result is based on the case $m=3$, which was established earlier in [Re85]. See also [BaGu00].)

F26: [ChGoLi01] If T is a k-strong n-tournament with $n \geq 8k$, then T contains k vertex-disjoint cycles that use all of the vertices of T.

Anti-Directed Paths

During the last 30 years, researchers have also searched for copies of other orientations of undirected paths and cycles in tournaments. Initially, study focused on oriented paths and cycles that contain no directed path of length 2 (called *anti-directed* paths and cycles), and successes there led to more general results on arbitrary oriented paths and cycles.

TERMINOLOGY: In a digraph, a directed path of length k is sometimes called a ***k*-path**.

DEFINITION

D18: An ***anti-directed path*** (or ***cycle***) in a digraph D is a sequence of arcs that forms a path or cycle in the underlying graph of D but does not contain a directed path of length 2 in D.

EXAMPLE

E8: Two anti-directed paths and an anti-directed cycle are illustrated in Figure 3.3.4.

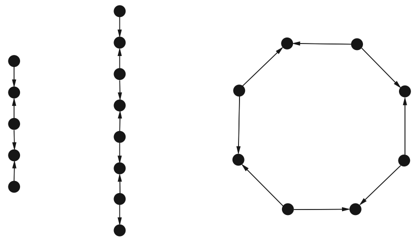


Figure 3.3.4 **Two anti-directed paths and an anti-directed cycle.**

FACTS

F27: [HaTh00a] Let T be a tournament different from the 3-cycle, the regular 5-tournament, and the quadratic residue rotational tournament $R(1,2,4)$. Then T contains every orientation of an undirected hamiltonian path. (This was first proved for anti-directed hamiltonian paths in 1971 [Gr71].)

F28: [Pe84] If n is even and $n \geq 16$, then every n-tournament contains an anti-directed hamiltonian cycle. (The major part of this result was first established for all even $n \geq 50$ [Th73], and then the result was improved to all even $n \geq 28$ [Ro74].)

F29: [Ha00a] Every n-tournament, $n \geq 68$, contains every orientation of a (undirected) hamiltonian cycle except possibly the (directed) hamiltonian cycle when the tournament is reducible. (This was first proved for $n \geq 2^{128}$ [Th86].)

3.3.3 Scores and Score Sequences

Fact 30, due to the mathematical sociologist H. G. Landau [La53], is another basic result that is useful in studies on tournaments. Nearly a dozen proofs appear in the literature (see the survey [Re96] and subsequent paper [GrRe99]).

FACTS

F30: [La53] A sequence of n integers $(s_1, s_2, \ldots, s_{n-1}, s_n)$, where $s_1 \leq s_2 \leq \ldots \leq s_{n-1} \leq s_n$, is the score sequence of some n-tournament if and only if

$$\sum_{i=1}^{k} s_i \geq \binom{k}{2}, \text{ for } k = 1, 2, \ldots, n-1, \quad \text{and} \quad \sum_{i=1}^{n} s_i = \binom{n}{2}$$

F31: [HaNoCa65] A sequence of n integers $\langle s_1, s_2, \ldots, s_{n-1}, s_n \rangle$, where $s_1 \leq s_2 \leq \ldots \leq s_{n-1} \leq s_n$, is the score sequence of some strong n-tournament if and only if

$$\sum_{i=1}^{k} s_i > \binom{k}{2}, \text{ for } k = 1, 2, \ldots, n-1, \quad \text{and} \quad \sum_{i=1}^{n} s_i = \binom{n}{2}$$

(See also [HaMo66].)

F32: Let $S = (s_1, s_2, s_3, \ldots, s_{n-1}, s_n)$ be a sequence of $n \geq 2$ nonnegative integers where $s_1 \leq s_2 \leq \ldots \leq s_{n-1} \leq s_n \leq n-1$, and let $m = s_n$. S is the score sequence of some n-tournament if and only if the new sequence

$$s_1, s_2, \ldots, s_m, s_{m+1}-1, s_{m+2}-1, \ldots, s_{n-1}-1$$

when arranged in non-decreasing order, is the score sequence of some $(n-1)$-tournament. (See [ReBe79].)

F33: [Av80] A score sequence $S = (s_1, s_2, s_3, \ldots, s_{n-1}, s_n)$ is the score sequence for exactly one n-tournament T if and only if each of the strong components of T is simple, and the simple strong score sequences are (0), (1, 1, 1), (1, 1, 2, 2), and (2, 2, 2, 2, 2). (See also [Te98].)

F34: [BrQi84] Let $S = (s_1, s_2, \ldots, s_n)$ be a score sequence. Every n-tournament with score sequence S has a unique hamiltonian cycle if and only if

$$S = (1, 1, 2, 3, \ldots, n-3, n-2, n-2)$$

F35: [Ya88] and [Ya89] Every non-empty set of nonnegative integers is the set of scores for some tournament.

EXAMPLE

E9: It is easy to verify that the sequence (1, 1, 1, 4, 4, 4) satisfies the conditions of Fact 30, and hence, it is the score sequence for some 6-tournament. In fact, the 6-tournament consisting of two vertex-disjoint 3-cycles, A and B, where every vertex of A dominates every vertex of B has score sequence (1, 1, 1, 4, 4, 4). Since the sum of the first three scores equals $\binom{3}{2}$, Fact 31 implies that no 6-tournament with score sequence (1, 1, 1, 4, 4, 4) is strong.

REMARKS

R5: The transitive n-tournament is the only n-tournament in which all of the scores are distinct. So, each score occurs with frequency 1. Regular n-tournaments are the only n-tournaments in which all of the scores are the same. So, each score in a regular n-tournament occurs with frequency n. Thus, each of the sets $\{1\}$ and $\{n\}$ is the set of frequencies of scores in some n-tournament. Given a non-empty set F of positive integers, the least possible order of a tournament with set of score frequencies given by F was explicitly determined in [AlRe78].

R6: *Tournament Rankings.* Given the results of a round-robin competition, one would like to rank the teams or at least pick a clearcut winner. Unfortunately, not one of the many ranking methods that have been proposed is entirely satisfactory. Ranking by the order of a hamiltonian path (whose existence is guaranteed by Fact 7 in the previous subsection) does not work unless the path is unique, which is only the case for transitive tournaments [Fact 14]. Ranking by score vector usually results in ties, and a team that beats only a few teams, with those few teams having lots of wins, might deserve a better ranking. This suggests considering the *second-order score vector*, where each team's score is the sum of the out-degrees of the teams it beats. One can continue by defining the nth-order score vectors recursively. There is an asymptotic ranking obtained this way, related to the eigenvalues of the digraph. See [Mo68] for more details and references.

The Second Neighborhood of a Vertex

DEFINITION

D19: Let x be a vertex in a digraph D. The ***second neighborhood*** of x, denoted $N_D^{++}(x)$, is the set of all vertices of D reachable from x by a 2-path but not a 1-path. That is, $N_D^{++}(x) = [\ \bigcup_{y \in N_D^+(x)} N_D^+(y)\ ] - N_D^+(x)$.

FACTS

F36: [Fi96] Every tournament T contains a vertex x so that $\left|N_T^{++}(x)\right| \geq \left|N_T^+(x)\right|$.

F37: [HaTh00b] If a tournament T contains no transmitter, then there are at least two vertices that satisfy the condition in Fact 36.

CONJECTURE

Seymour's second neighborhood conjecture (see [Fi96]): Every digraph D contains a vertex z for which $\left|N_D^{++}(z)\right| \geq \left|N_D^+(z)\right|$.

3.3.4 Transitivity, Feedback Sets, Consistent Arcs

In a tournament that represents the outcomes of a paired-comparison experiment (or the results of a round-robin competition or the results of majority voting by an electorate in which there are no ties), there is much interest in attempts to measure the consistency of choices by the subject who made the comparisons (or the consistency of wins among the participants or the consistency of the electorate's choice among the alternatives). Consistency corresponds to a lack of cycles. So, one measure is the largest number of vertices that induce a transitive sub-tournament in the outcome tournament. Another measure is the largest number of arcs of the outcome tournament that do not contain the arcs of a cycle.

DEFINITION

D20: A ***feedback set of arcs*** in a tournament T is a set S of arcs such that the digraph $T - S$ contains no cycle.

EXAMPLE

E10: Let T be the tournament with vertex-set $\{1, 2, \ldots, n\}$ in which j dominates k whenever $j > k$, except that i dominates $i + 1$, for $i = 1, 2, \ldots, n - 1$. Then the set of arcs $\{(i, i + 1) \mid 1 \leq i \leq n - 1)\}$ is a feedback set of arcs in T, and the smaller set of arcs $\{(i, i + 1) \mid 1 \leq i \leq n - 1,\ i \text{ odd}\}$ is also a feedback set of arcs in T.

Smallest Feedback Sets

Finding a smallest feedback set in an n-tournament T is equivalent to finding a transitive n-tournament (or linear order) W such that $V(W) = V(T)$ and the number of pairs $\{x, y\}$ of distinct vertices in which x dominates y in T but not in W is as small as possible.

The latter problem is known as a *Slater problem* in the voting literature involving tournaments. See [ChHuWo96]. Work on the computation of a minimum-weighted feedback set of arcs in an arc-weighted tournament is reported in [ChGuHuWo97] and [ShYu01].

FACTS

F38: A smallest set of arcs in a tournament T whose reversal yields a transitive tournament is a smallest feedback set in T. (See [BaHuIsRoTe95].)

F39: The number of arcs in a smallest feedback set in a tournament T is equal to the number of arcs in a smallest transversal of the cycles in T. (See [BaHuIsRoTe95].)

F40: [BaHuIsRoTe95] If R is a smallest feedback set in a tournament T, then every arc of R is contained in some 3-cycle of T.

F41: [BaHuIsRoTe95] A digraph D is acyclic if and only if its arc set is a smallest feedback set of some tournament.

Acyclic Subdigraphs and Transitive Sub-Tournaments

DEFINITION

D21: A set of arcs in a digraph D is a ***consistent set of arcs*** if it induces an acyclic subdigraph of D.

FACTS

F42: [Sp71/72, Fe83] If $g(n)$ denotes the largest integer so that every n-tournament contains a consistent set of $g(n)$ arcs, then there are positive constants c_1 and c_2 so that $\frac{1}{2}\binom{n}{2} + c_1 n^{3/2} \leq g(n) \leq \frac{1}{2}\binom{n}{2} + c_2 n^{3/2}$. Moreover, values for $n \leq 12$ are as follows ([Re69, Be72]):

N	2	3	4	5	6	7	8	9	10	11	12
$g(n)$	1	2	4	6	9	12	20	24	30	35	44-46

F43: [PaRe70, Ne94, Sá94] If $v(n)$ denotes the largest integer such that every n-tournament contains a transitive sub-tournament with at least $v(n)$ vertices, then

$$v(n) = \begin{cases} 3 & \text{for } 4 \leq n \leq 7 \\ 4 & \text{for } 8 \leq n \leq 13 \\ 5 & \text{for } 14 \leq n \leq 27 \\ 6 & \text{for } 28 \leq n \leq 31 \end{cases}$$

$$\lfloor \log_2(16n/7) \rfloor \leq v(n) \leq 2\lfloor \log_2 n \rfloor + 1 \quad \text{for } 32 \leq n \leq 54$$

$$v(n) \geq \lfloor \log_2(n/55) \rfloor + 7 \quad \text{for } n \geq 55$$

F44: [AoHa98] and [GuGyThWe98] For any tournament score sequence $S = (s_1, s_2, \ldots, s_{n-1}, s_n)$, where $s_1 \leq s_2 \leq s_3 \leq \ldots \leq s_{n-1} \leq s_n$, there exists a tournament T on vertex set $\{1, 2, \ldots, n\}$ so that the score of vertex i is s_i and the sub-tournaments of T on both the even and the odd indexed vertices are transitive, i.e., i dominates j whenever $i > j$ and $i \equiv j \pmod 2$. (See also [BrSh01] for a shorter proof, and see [BaBeHa92] for origins of the result in terms of the so-called cyclic chromatic number of a tournament.)

Arc-Disjoint Cycles

NOTATION: For a given tournament T, $a(T)$ denotes the maximum number of arc-disjoint cycles in T, and $c(T)$ denotes the number of arcs in a smallest feedback set in T. Also, let $a(n) = \max\{a(T)\}$ and $c(n) = \max\{c(T)\}$, where the maxima are taken over all n-tournaments T.

REMARK

R7: Note that the quantity $a(n)$ equals the maximum number of edge-disjoint (undirected) cycles in the complete graph of order n, which has been shown by [ChGeHe71] to equal $\lfloor (n/3) \lfloor (n-1)/2 \rfloor \rfloor$.

FACTS

F45: For any tournament T, $a(T) \leq c(T)$.

F46: [BeKo76] For $n \geq 10$, $a(n) < c(n)$. That is, for each $n \geq 10$, there exists an n-tournament T such that a smallest feedback set in T contains more arcs than in a largest collection of arc-disjoint cycles in T. (See also [Be75] and the discussions in [BaGu01] and [Is95].)

3.3.5 Kings, Oriented Trees, and Reachability

Kings arose in an attempt to determine the "strongest" individuals in certain small animal societies in which there exists a pairwise "pecking" relationship (see work referenced in [La53]). The delightful article by Maurer [Ma80] stimulated early interest in the topic. Extensions of the idea led to new investigations into combinatorial sub-structures in tournaments involving oriented trees and other "reachability sub-structures". The concept of a king appeared independently as "uncovered vertices" in some of the voting theory literature (see §3.3.8 below). Moreover, there is current interest in kings and generalizations in other digraphs, particularly in multi-partite tournaments (e.g., see the discussion in [Re96]).

DEFINITIONS

D22: A ***king*** in a tournament T is a vertex x such that for every other vertex y, there is a 1-path or a 2-path from x to y in T.

D23: A ***serf*** in a tournament is a vertex x such that for every other vertex y, there is a 1-path or a 2-path from y to x.

D24: A tournament is ***k*-stable**, $k \geq 1$, if every vertex is a king and more than k arcs must be reversed in order to reduce the number of kings.

FACTS

F47: [Va52] Every tournament contains a king. In fact, every vertex of maximum score is a king [La53].

F48: [Ma80] For positive integers k and n, there exists an n-tournament with exactly k kings if and only if $1 \leq k \leq n$, $k \neq 2$, and $(k, n) \neq (4, 4)$ (see also [Re82]).

F49: [Re80] For integers $0 \le b \le s \le k \le n$, there exists an n-tournament with exactly k kings, exactly s serfs, and exactly b vertices that are both kings and serfs if and only if the following four conditions are satisfied: (1) $k + s - b \le n$, (2) $s \ne 2$ and $k \ne 2$, (3) either $n = k = s \ne 4$ or $k < n$ and $b < s$, (4) (n, k, s, b) is none of $(n, 4, 3, 2)$, $(5, 4, 1, 0)$, or $(7, 6, 3, 2)$. (See [Mo93] for a shorter proof.)

F50: [ReBr84] A k-stable tournament must have at least $4k + 3$ vertices. Moreover, the following three statements are equivalent: (1) There exists a k-stable $(4k + 3)$-tournament. (2) There exists a $(4k+4)$ by $(4k+4)$ skew-symmetric Hadamard matrix. (3) There exists a doubly regular $(4k + 3)$-tournament.

F51: [Ma80] In almost all tournaments every vertex is a king. (See also [Mo68].) In fact, for each positive integer k, almost all tournaments are k-stable [ReBr84].

REMARKS

R8: Bounds on the least number of vertices that need be adjoined to an n-tournament T to form a new super tournament W so that the set of kings in W is exactly the vertices of T were described in [Re82] and [Wa84]. The least order of a tournament Z in which all vertices of Z are kings and that contains T as a sub-tournament was determined in [Re80]. For other work on kings in tournaments the reader is referred to the references in [Re96].

R9: By definition, a king is the root of a rooted spanning tree of depth at most 2. So, it is natural to consider the existence of other oriented trees in tournaments.

Tournaments Containing Oriented Trees

TERMINOLOGY: An ***out-branching*** (or ***out-tree***) in a digraph is a rooted spanning tree, and an ***in-branching*** (or ***in-tree***) is an out-branching with all the arcs reversed.

CONJECTURE

Sumner's conjecture (see [ReWo83]): Every $(2n - 2)$-tournament contains every orientation of every tree of order n.

EXAMPLE

E11: All of the 8 oriented trees of order 4 are shown in Figure 3.3.5. A copy of each can be found in any 6-tournament.

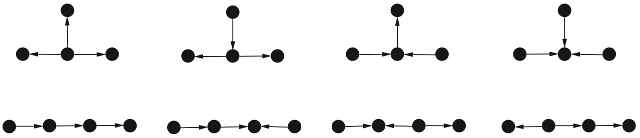


Figure 3.3.5 **The oriented trees of order 4.**

REMARK

R10: Note that no integer smaller than $2n - 2$ will suffice in the statement of the conjecture, for a score of at least $n-1$ is required to accommodate the "out-orientation" of the tree $K_{1,n-1}$; any regular $(2n-3)$-tournament fails to have a score of $n - 1$. Over the last 20 years several papers reported partial and related results (see the references in [HaTh00b]), all of which support the conjecture.

FACTS

F52: [HaTh00b] If $f(n)$ denotes the least integer m so that every m-tournament contains every orientation of every tree of order n, then $f(n) \leq (7n-5)/2$. (Earlier efforts yielded $f(n) \leq 12n$ [HäTh91] and then $f(n) \leq \frac{38}{5}n - 6$ [HaTh00b].)

F53: [LuWaPa00] Every n-tournament, $n \geq 800$, contains a spanning rooted 2-tree of depth 2 so that, with at most one exception, all vertices that are not a leaf or the root have out-degree 2.

F54: [Pe02] Each rotational $(2n+1)$-tournament contains all rooted trees of order $2n+1$ in which there are at most n branches, each of which is a directed path.

F55: [Ba91] A tournament T contains an out-branching and an in-branching that are arc-disjoint, both rooted at a specified vertex v, if and only if T is strong and for each arc (x, y) of T, the digraph $T - \{(x, y)\}$ contains either an out-branching rooted at v or an in-branching rooted at v. If T is 2-arc-strong, then for every pair of vertices x and y, there is an out-branching rooted at x and an arc disjoint in-branching rooted at y.

Arc-Colorings and Monochromatic Paths

CONJECTURE

P. Erdös conjecture (see [SaSaWo82]): for each positive integer k, there is a least positive integer $s(k)$ so that every arc-colored tournament involving k colors contains a set S of $s(k)$ vertices with the property that for every vertex y not in S, there is a monochromatic path from y to some vertex in S.

REMARK

R11: This conjecture considers reachability in tournaments via monochromatic paths. Since every tournament contains a hamiltonian path, $s(1) = 1$. Fact 56 below implies that $s(2) = 1$. A certain coloring of the 9-tournament that is the lexicographic product of a 3-cycle with a 3-cycle shows that $s(3) > 2$ (see [SaSaWo82]). In particular, Erdös asked if $s(3) = 3$. It is not even known that $s(k)$ is finite for $k \geq 3$. Some progress on this conjecture is included below. (See also [LiSa96] and [Re00] for a relaxation of the problem and several open questions.)

FACTS

F56: [SaSaWo82] If the arcs of a tournament T are colored with two colors, then there exists a vertex x in T so that for every vertex $y \neq x$ in T, there is a monochromatic path from y to x. (See [Re84] for another proof.)

F57: [Sh88] If the arcs of a tournament T are colored with three colors and T does not contain a 3-cycle or a transitive sub-tournament of order 3 whose arcs use all three colors, then there exists a vertex x in T so that for every vertex $y \neq x$ in T, there is a monochromatic path from y to x.

3.3.6 Domination

Issues concerning domination have played an important role in the development of tournament theory. However, exact results on domination numbers of tournaments are scarce. For example, the problem of determining the smallest order of a tournament T with domination number $\gamma(T) = k$ for a given integer k has only some partial results. Bounds are known, some of which are constructive, but the exact value is known only for small values of k. Domination in general (undirected) graphs is discussed in §9.2.

DEFINITIONS

D25: A ***dominating set*** in a tournament T is a set S of vertices in T such that every vertex not in S is dominated by some vertex in S.

D26: The ***domination graph*** of a tournament T is an undirected graph G that has the same vertex-set as T, and x is adjacent to y in G whenever $\{x, y\}$ is a dominating set in T.

D27: A ***spiked cycle*** is a connected (undirected) graph with the property that when all vertices of degree 1 are removed, a cycle results.

D28: The ***domination number*** of a tournament is the minimum cardinality of a dominating set in T, denoted $\gamma(T)$.

EXAMPLE

E12: Let T denote the transitive n-tournament with vertex-set $\{1, 2, \ldots, n\}$ in which j dominates i whenever $j > i$. Reversal of the arc $(n, 1)$ yields a strong n-tournament W in which vertex 1 can reach vertex n via a 1-path, and 1 can reach vertices $2, 3, \ldots, (n-1)$ via 2-paths (through vertex n). So, 1 is a king in W. Since W has no transmitter, and every vertex in W is dominated by 1 or n, $\{1, n\}$ is a dominating set in W and $\gamma(W) = 2$.

FACTS

F58: [GrSp71] For a positive integer k, let p denote the smallest prime number greater than $k^2 2^{2k-2}$, where $p \equiv 3 \pmod 4$. The domination number of the quadratic residue p-tournament is greater than k. (See also [ReMcHeHe02].)

F59: [FiLuMeRe98] The domination graph of a tournament is either a spiked odd cycle with or without isolated vertices, or a forest of caterpillars. In particular, the domination graph of an n-tournament has at most n edges. Furthermore, any spiked odd cycle with or without isolated vertices is the domination graph of some tournament.

F60: [FiLuMeRe99] A connected graph is the domination graph of a tournament if and only if it is either a spiked odd cycle, a star, or a caterpillar whose spine has positive length and has at least three vertices of degree 1 adjacent to one end of its spine. The tournaments that have a connected domination graph were characterized in [JiLu98].

F61: [SzSz65] If T is an n-tournament with $n \geq 2$, then the domination number of T satisfies $\gamma(T) \leq log_2 n - log_2 log_2 n + 2$.

3.3.7 Tournament Matrices

Some early work on tournament matrices included the results by Brauer and Gentry and by J. W. Moon and N. J. Pullman (Fact 22 and Fact 23 in §3.3.1; see also [BrGe72]); work by H. J. Ryser [Ry64] on tournament matrices with given row and column sum that minimize the number of "upsets", i.e., the number of 1's above the main diagonal; and work by D. R. Fulkerson [Fu65] that described the tournament matrices with prescribed row sums that minimize and maximize the number of upsets. R. A. Brualdi and Q. Li [BrLi83a] continued the upset theme and expanded on the work of Ryser and Fulkerson. These last three references may be thought of as papers on ranking since minimizing upsets gives rise to orderings of the vertices that minimize the number of losses by stronger players to weaker players.

DEFINITION

D29: A ***tournament matrix*** is a square matrix $M = (m_{ij})$ of 0's and 1's, with 0's on the main diagonal and $m_{ij} + m_{ji} = 1$, for all distinct i and j.

TERMINOLOGY: For a given ordering of the vertices, $v_1, v_2, \ldots, v_n$, of a tournament T, the ***adjacency matrix*** $M = [m_{ij}]$ of T is the 0-1 matrix given by

$$m_{ij} = \begin{cases} 1 & \text{if } v_i \text{ dominates } v_j \\ 0 & \text{otherwise} \end{cases}$$

Thus, a tournament matrix is the adjacency matrix of some tournament for a given ordering of the vertices.

EXAMPLE

E13: A tournament matrix of order 6 is shown in Figure 3.3.6.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Figure 3.3.6 **A tournament matrix.**

REMARK

R12: An elementary observation about an n by n tournament matrix M is that $M + M^t + I_n = J_n$, where I_n is the n by n identity matrix, M^t is the transpose matrix of M, and J_n is the n by n matrix of all 1's. Moreover, any adjacency matrix M of T can be obtained from any other adjacency matrix N of T by permuting the order of the vertices used to obtain N, i.e., there is a permutation matrix P such that $M = P^{-1}NP$. Thus, the eigenvalues are the same for all of the tournament matrices corresponding to a particular tournament.

FACTS

F62: [BrGe68] Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the eigenvalues of an n by n tournament matrix A, where $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$. Then $0 < \lambda_1 \leq (n-1)/2$, and $|\lambda_j| \leq \lfloor n(n-1)/2j \rfloor^{1/2}$, $j = 2, 3, \ldots, n$. Moreover, if M is an n by n tournament matrix and λ_M denotes $\max\{|\lambda| : \lambda \text{ an eigenvalue of } M\}$, then for odd n, $\max\{\lambda_M : M \text{ an } n \text{ by } n \text{ tournament matrix}\}$ is attained by the regular tournament matrices.

F63: [CaGrKiPuMa92] For all $n \geq 3$, each irreducible n by n tournament matrix M has at least three distinct eigenvalues; such a matrix has exactly three distinct eigenvalues if and only if it is a Hadamard tournament matrix (i.e., $M^tM = nI$). There is an irreducible n by n tournament matrix with exactly n distinct eigenvalues.

F64: [Mi95] If A is an n by n tournament matrix, then the rank of A is equal to $(n-1)/2$ if and only if n is odd and $AA^t = 0$. Equality implies that the characteristic of the field divides $(n-1)/2$ (without the hypothesis of regularity). Examples of order n having rank $(n-1)/2$ for $n \equiv 1 \pmod 4$ for fields of characteristic p, where p divides $(n-1)/4$, can be obtained from doubly regular tournament matrices of order $(n-2)$ by adding an $(n-1)$st row of n 0's (and hence, an $(n-1)$st column of all 1's save for the 0 in the $(n-1, n-1)$ position) followed by an n^{th} row of $(n-1)$ consecutive 1's and a 0 in the (n, n) position (and hence, an n^{th} column of n 0's).

F65: [Sh92] A tournament matrix is singular if more than one-fourth of the triples of vertices in the corresponding tournament induce 3-cycles. All tournament matrices realizing a given score sequence are nonsingular if and only if the scores are "sufficiently close" to one another. The spectral radius of a singular n by n tournament matrix is less than or equal to $(1/2)(n-1)$, and equality implies that exactly one-fourth of the triples of vertices in the corresponding tournament induce 3-cycles.

3.3.8 Voting

Work on acyclic digraphs in tournaments, including transitive sub-tournaments, is of interest in voting theory since such structures give a measure of group consistency by the voters. Readers can find a rich source of problems and issues in selected articles in the social choice literature that treats voting theory, particular examples of such literature include the periodicals *Social Choice and Welfare*, *Mathematical Social Sciences*, *Public Choice*, and *The American Journal of Political Science.*

Deciding Who Won

A central issue in voting theory is to pick a "best" alternative (or subset of the alternatives) given that voter preferences have been aggregated. A "best" alternative or subset of alternatives is called a solution and is thought of as the "winners". Several tournament solutions have been considered in the literature. Each is to be non-empty, invariant under isomorphism, and uniquely the Condorcet winner if there is one in the tournament. Some of these solutions are: the vertices of largest score (the Copeland solution), vertices based on the maximum eigenvalue of the adjacency matrix of the tournament, vertices associated with a Markov method, the Condorcet winner of a transitive tournament that is "closest" to the given tournament (called the *Slater solution*), vertices that are uncovered relative to a certain "covering" relation, vertices that are transmitters of maximal transitive sub-tournaments (the Banks set), and vertices satisfying a special axiomatic formulation (the tournament equilibrium set). These are discussed in detail in J.-F. Laslier's monograph [La97].

DEFINITIONS

D30: The ***Condorcet winner*** is a candidate (or alternative) x such that for every other candidate (or alternative) y, x is preferred over y by a majority of the voters.

D31: The ***majority digraph*** D of a set of n-tournaments, all with the same vertex-set V, has vertex-set V, and vertex x dominates vertex y in D if and only if x dominates y in a majority of the n-tournaments.

D32: A digraph D is ***induced*** by a set of voters if D is the majority digraph based on a collection of linear orderings of the vertices of D, exactly one for each voter. (The linear orders represent preferences by the voters for the alternatives that are the vertices of D. Different voters might have the same linear order.)

D33: The ***Condorcet paradox*** is that the voters may be consistent in their preferences (i.e., each of their rankings of the n candidates is a linear order), but the amalgamation of voters' preferences using majority rule can result in inconsistencies (i.e., cycles in the majority digraph).

REMARK

R13: In the definition of a majority digraph, the common vertex set may be thought of as a finite set of n "alternatives", and each n-tournament may be thought of as the pairwise preferences of the alternatives by a "voter". So, the resulting majority digraph represents voters' preferences under majority voting. If there are an odd number of voters or there are no ties, then the majority digraph is a tournament.

EXAMPLE

E14: Figure 3.3.7 illustrates the majority tournament T of the set of three transitive 5-tournaments T_1, T_2, and T_3. In the drawings of T_1, T_2, and T_3, the long lines directed downward are to mean that each vertex dominates exactly the vertices below it. For example, in the second tournament from the left, vertex c dominates exactly vertices a and d, and vertex a is dominated by exactly vertices b, e, and c. The existence of cycles in the majority digraph illustrates the Condorcet paradox. For instance, it shows that a majority of voters prefer a to b, a majority prefer b to c, and yet, a majority prefer c to a.

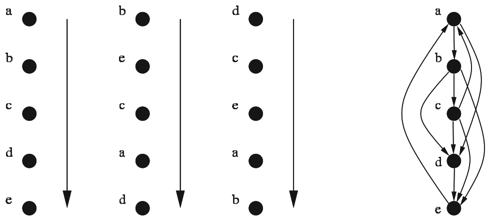


Figure 3.3.7 **The majority digraph of 3 tournaments.**

Tournaments That Are Majority Digraphs

FACTS

F66: [St59] Every n-tournament (indeed, every oriented graph of order n) is the majority digraph of some collection of $n+1$ tournaments, for n odd, and of $n+2$ tournaments, for n even.

NOTATION: Let $m(n)$ denote the smallest integer such that any n-vertex digraph can be induced as a majority digraph by a collection of $m(n)$ or fewer voters, and let $g(n)$ denote the smallest integer such that any n-tournament can be induced as a majority digraph by a collection of $g(n)$ or fewer voters.

F67: For large n, $m(n) > (.55n/logn)$ [St59], and there exists a constant c so that $m(n) < (cn/logn)$ [ErMo64].

F68: [Mo68] The integer $g(n)$ is always odd, $g(3) = g(4) = g(5) = 3$, $g(n+1) \leq g(n) + 2$, and $m(n) \leq 2g(n)$.

F69: [Ma99] In contrast to the situation for majority tournaments, for any λ, $1/2 < \lambda \leq 1$, there exists an integer n and a labeled n-tournament T, so that for every collection C of transitive tournaments on the same label set as T, there is an arc (u, v) of T such that the proportion of C in which u dominates v is less than λ. In short, T is not the λ-*majority tournament* for any collection of transitive n-tournaments.

Agendas

DEFINITIONS

D34: An ***agenda*** is an ordered list of alternatives (i.e., an ordered list of the vertices of a majority tournament).

D35: An ***amendment procedure*** of voting is a sequential voting process in which, given an agenda $(a_1, a_2, \ldots, a_n)$ of alternatives, alternative a_1 is pitted against a_2 in the first vote, then the winner is pitted against a_3 in the second vote, then the winner is pitted against a_4 in the third vote, etc.

D36: Given a majority n-tournament T and an agenda $(a_1, a_2, \ldots, a_n)$ of alternatives given by the vertices of T, the ***sincere decision*** is the alternative surviving the last vote (i.e., the $(n-1)^{\text{th}}$ vote) in an amendment procedure of voting using majority voting at each stage. It is a function of the agenda and T. The ***decision tree*** is the spanning, rooted subtree of T, rooted at the sincere decision, induced by the $n-1$ arcs of T that describe the $n-1$ votes taken in the amendment procedure using T and $(a_1, a_2, \ldots, a_n)$.

EXAMPLE

E15: Given the agenda (b, e, c, a, d) and the majority tournament shown in Figure 3.3.8, alternative a is the sincere decision. The corresponding decision tree rooted at vertex a is also shown.

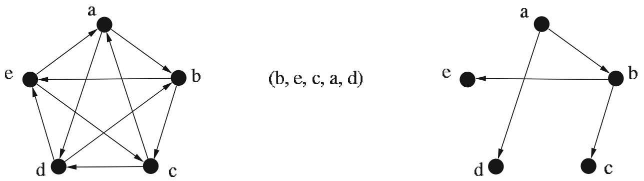


Figure 3.3.8 **Majority tournament, agenda, and decision tree.**

FACT

F70: [Mi77] For any tournament T, the set of vertices that can be obtained as the sincere decision under amendment procedure is exactly the set of vertices in the initial strong component of T. (For another proof, see [Re91a].)

Division Trees and Sophisticated Decisions

DEFINITIONS

D37: Given an agenda $(a_1, a_2, \ldots, a_n)$, the ***division tree*** of $(a_1, a_2, \ldots, a_n)$ is the labeled, balanced, binary, rooted tree on $2^n - 1$ vertices labeled by non-empty subsequences of the agenda $(a_1, a_2, \ldots, a_n)$; the root is labeled $(a_1, a_2, \ldots, a_n)$; and, for $0 \leq j \leq n-2$, a vertex at level j which is labeled by a subsequence of $(a_1, a_2, \ldots, a_n)$, say $(b_1, b_2, b_3, \ldots, b_{n-j})$, dominates exactly two vertices at level $j+1$, one labeled $(b_1, b_3, \ldots, b_{n-j})$, and one labeled $(b_2, b_3, \ldots, b_{n-j})$.

D38: Let T be a majority n-tournament and let $(a_1, a_2, \ldots, a_n)$ be an agenda of alternatives given by the vertices of T. The ***sophisticated decision*** is the anticipated decision at the root of the division tree relative to $(a_1, a_2, \ldots, a_n)$ and T, where the anticipated decision at each vertex at level $n-2$ of the division tree is the majority choice in T between the two alternatives that make up the ordered pair labeling that vertex in the division tree; and inductively, for $0 \leq j < n-2$, the anticipated decision at each vertex v of level j in the division tree is the majority choice in T between the anticipated decisions at the two vertices at level $j+1$ that are dominated by v.

FACTS

F71: [Ba85] The set of vertices in a tournament T that can be obtained as the sophisticated voting decision under amendment procedure relative to some agenda is equal to the set of vertices of T that are transmitters of maximal transitive sub-tournaments of T.

F72: No alternative is unanimously preferred to the sophisticated voting decision. (Observed in [Mi77] and [Mi80] and proved in [Re91b].)

F73: [Re97] A tournament T admits an agenda for which the sincere voting decision and the sophisticated voting decision are identical if and only if the initial strong component of T is not a 3-cycle. As a result, asymptotically, most tournaments admit such an agenda.

EXAMPLES

E16: The division tree of the agenda (x, y, z) is shown in Figure 3.3.9. Given the majority tournament shown, the anticipated decisions at levels 1 and 0 of the division tree are underlined in the vertex labels. The anticipated decision at the root is y, so y is the sophisticated decision relative to this tournament and agenda. Note that the sincere decision is z, which illustrates Fact 73.

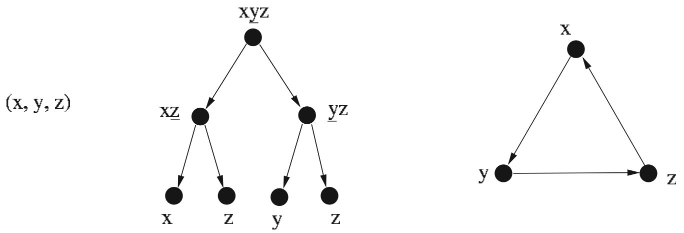


Figure 3.3.9 **Agenda, division tree, and majority tournament.**

E17: The majority 4-tournament shown in Figure 3.3.10 illustrates the positive case for Fact 73. As before, the anticipated decisions are underlined. For the agenda (y, v, u, x), the sincere decision and the sophisticated decision are both u.

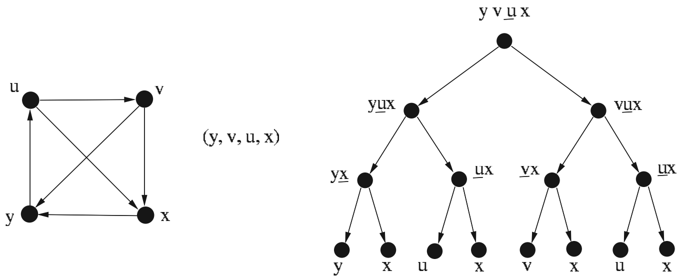


Figure 3.3.10 **A majority 4-tournament.**

Inductively Determining the Sophisticated Decision

The following result yields an algorithm for determining the sophisticated decision that is much more straightforward than using the definition. (Recall that $I(z)$ denotes the in-set of vertex z.)

FACT

F74: [ShWe84] Let T be a majority n-tournament and let $(a_1, a_2, \ldots, a_n)$ denote an agenda composed of the alternatives that make up the vertices of T. Inductively define the sequence $\langle z_1, z_2, \ldots, z_n \rangle$ as follows: $z_n = a_n$, and for $1 \leq j < n$,

$$z_j = \begin{cases} a_j & \text{if } a_j \in \bigcap_{i=j+1}^{n} I(z_i) \\ z_{j+1} & \text{otherwise} \end{cases}$$

Then z_1 is the sophisticated decision.

References

[Ád64] A. Ádám, Problem, 12–18 in *Theory of Graphs and Applications, Proc. Coll. Smolenice*, Czech. Acad. Sci. Publ., 1964.

[Al67] B. Alspach, Cycles of each length in regular tournaments, *Canad. Math. Bull.* 10 (1967), 283–285.

[AlRe78] B. Alspach and K. B. Reid, Degree frequencies in digraphs and tournaments, *J. Graph Theory* 2 (1978), 241–249.

[AlReRo74] B. Alspach, K. B. Reid and D.P. Roselle, Bypasses in asymmetric digraphs, *J. Combin. Theory B* 17 (1974), 11–18.

[AlTa82] B. Alspach and C. Tabib, A note on the number of 4-circuits in a tournament, *Annals Discrete Math.* 12 (1982), 13–19.

[AoHa98] S. Ao and D. Hanson, Score vectors and tournaments with cyclic chromatic number 1 or 2, *Ars Combin.* 49 (1998), 185–191.

[Av80] P. Avery, Condition of a score sequence to be simple, *J. Graph Theory* 4 (1980), 157–164.

[Ba85] J. S. Banks, Sophisticated voting outcomes and agenda control, *Social Choice and Welfare* 1 (1985), 295–306.

[Ba91] J. Bang-Jensen, Edge-disjoint in- and out-branchings in tournaments and related path problems, *J. Combin. Theory B* 51 (1991), 1–23.

[BaBeHa92] K. S. Bagga, L. W. Beineke and F. Harary, Two problems on coloring tournaments, Vishrva Internat. *J. Graph Theory 1* (1992), 83–94.

[BaGu00] J. Bang-Jensen, Y. Guo, A. Yeo, Complementary cycles containing prescribed vertices in tournaments, *Discrete Math.* 214 (2000), 77–87.

[BaGu01] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms, and Applications*, Springer, 2001.

[BaGu96] J. Bang-Jensen and G. Gutin, Paths, trees, and cycles in tournament, *Congressus Numer.* 115 (1996), 131–170.

[BaGu98] J. Bang-Jensen and G. Gutin, Generalizations of tournaments: a survey, *J. Graph Theory* 28 (1998), 171–202.

[BaHuIsRoTe95] J.-P. Barthélémy, O. Hurdy, G. Isaak, F. S. Roberts, B. Tesman, The reversing number of a digraph, *Discrete Appl. Math.* (1995), 39–76.

[Be72] J. C. Bermond, Ordes à distance minimum d'un tournoi et graphes partiels sans circuits maximal, *Math. Inform. Sci. Humaines* 37 (1972), 5–25.

[Be75] J.-C. Bermond, The circuit hypergraph of a tournament, 165–180 in *Infinite and Finite Sets*, North Holland, 1975.

[Be81] L. W. Beineke, A tour through tournaments or bipartite and ordinary tournaments: a comparative survey, 41–55 in *Combinatorics*; Swansea, 1981, *London Math. Soc. Lecture Notes* 52, Cambridge University Press, Cambridge, 1981.

[BeKo76] J.-C. Bermond and Y. Kodratoff, Une heuristic pour le calcul de l'indice de transitivite d'un tournoi, *R. A. I. R. O. Informatique Theorique* 10 (1976), 83–92.

[BeTh81] J.-C. Bermond and C. Thomassen, Cycles in digraphs: a survey, *J. Graph Theory* 5 (1981) 1–43.

[BeWi75] L. W. Beineke and R. J. Wilson, A survey of recent results on tournaments, 31–48 in *Recent Advances in Graph Theory, Proc. Symp. Prague 1974*, Academia, 1975.

[BrGe68] A. Brauer and I. C. Gentry, On the characteristic roots of tournament matrices, *Bull. Amer. Math. Soc.* 74 (1968), 1133–1135.

[BrGe72] A. Brauer and I. C. Gentry, Some remarks on tournament matrices, *Linear Algebra Appl.* 5 (1972), 311–318.

[BrLi83a] R. A. Brualdi and Q. Li, Upsets in round robin tournaments, *J. Combin. Theory B* (1983), 62–77.

[BrQi84] R. A. Brualdi and Q. Li, The interchange graph of tournaments with the same score vector, 129–151 in *Progress in Graph Theory*, Academic Press, 1984.

[BrSh01] R. A. Brualdi and J. Shen, Landau's inequalities for tournament scores and a short proof of a theorem on transitive sub-tournaments, *J. Graph Theory* 38 (2001), 244–254.

[Ca59] P. Camion, Chemins et circuits hamiltonioens des graphes complets, *C. R. Acad. Sci.* Paris 249 (1959), 2151–2152.

[CaGrKiPuMa92] D. de Caen, D. A. Gregory, S. J. Kirkland, N. J. Pullman, J. S. Maybe, Algebraic multiplicity of the eigenvalues of a tournament matrix, *Linear Algebra Appl.* 169 (1992), 179–193.

[ChGeHe71] G. Chartrand, D. Geller and S. Hedetniemi, Graphs with forbidden subgraphs, *J. Combin. Theory B* 10 (1971), 12–41.

[ChGoLi01] G. Chen, R. J. Gould and H. Li, Partitioning vertices of a tournament into independent cycles, *J. Combin. Theory B* 83 (2001), 213–220.

[ChGuHuWo97] I. Charon-Fournier, A. Guénoche, O. Hudry and F. Woirgard, New results on the computation of median orders, *Discrete Math.* 165/166 (1997), 139–153.

[ChHuWo96] I. Charon-Fournier, O. Hudry and F. Woirgard, Ordres medians et orders de Slater des tournois, *Math. Inform. Sci. Humaines* 133 (1996), 23–56.

[Da54] R. L. Davis, Structures of dominance relations, *Bull. Math. Biophys.* 16 (1954), 131–140.

[ErMo64] P. Erdös and L. Moser, On the representation of directed graphs as unions of orderings, *Publ. Math. Inst. Hung. Acad. Sci.* 9 (1964), 125–132.

[Fe83] W. Fernandez de la Vega, On the maximum cardinality of a consistent set of arcs in a random tournament, *J. Combin. Theory B* (1983), 328–332.

[Fi96] D. C. Fisher, Squaring a tournament: proof of Dean's conjecture, *J. Graph Theory* 23 (1996), 43–48.

[FiLuMeRe98] D. C. Fisher, J. R. Lundgren, S. Merz and K. B. Reid, The domination and competition graphs of a tournament, *J. Graph Theory* 29 (1998), 103–110.

[FiLuMeRe99] D. C. Fisher, J. R. Lundgren, S. Merz, and K. B. Reid, Connected domination graphs of tournaments, *J. Comb. Math. and Combin. Comput.* 31 (1999), 169–176.

[FrTh87] P. Fraise and C. Thomassen, Hamiltonian dicycles avoiding prescribed arcs in tournaments, *Graphs and Combin.* 3 (1987), 239–250.

[Fu65] D. R. Fulkerson, Upsets in round robin tournaments, *Canad. J. Math.* 17 (1965), 957–969.

[Ga72] M. R. Gary, On enumerating tournaments that admit exactly one Hamiltonian circuit, *J. Combin. Theory B* 13 (1972), 266–269.

[Gr71] B. Grünbaum, Anti-directed Hamiltonian paths in tournaments, *J. Combin. Theory B* 11 (1971), 249–257.

[GrRe99] J. R. Griggs and K. B. Reid, Landau's Theorem revisited, *Australasian J. Combin.* 20 (1999), 19–24.

[GrSp71] R. L. Graham and J. H. Spencer, A constructive solution to a tournament problem, *Canad. Math. Bull.* 14 (1971), 45–48.

[Gu95] G. Gutin, Cycles and paths in semicomplete multipartite digraphs, theorems, and algorithms: a survey, *J. Graph Theory* 19 (1995), 481–505.

[GuGyThWe98] B. Guiduli, A. Gyárfás, S. Thomassé and P. Weidl, 2-partition-transitive tournaments, *J. Combin. Theory B* 72 (1998), 181–196.

[GuVo97] Y. Guo and L. Volkman, Bypasses in tournaments, *Discrete Appl. Math.* 79 (1997), 127–135.

[Ha00a] F. Havet, Oriented Hamiltonian cycles in tournaments, *J. Combin. Theory B* (2000), 1–31.

[Hä93] R. Häggkvist, Hamiltonian cycles in oriented graphs, *Combin. Probab. Comput.* 2 (1993), 25–32.

[HaMo66] F. Harary and L. Moser, The theory of round robin tournaments, *Amer. Math. Monthly* 73 (1966), 231–246.

[HaNoCa65] F. Harary, R. Z. Norman, D. Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs*, John Wiley & Sons, 1965.

[HaTh00a] F. Havet and S. Thomassé, Oriented Hamiltonian paths in tournaments: a proof of Rosenfeld's conjecture, *J. Combin. Theory B* 78 (2000), 243–273.

[HaTh00b] F. Havet and S. Thomassé, Median orders of tournaments: a tool for the second neighborhood problem and Sumner's conjecture, *J. Graph Theory* 35 (2000), 244–256.

[HäTh91] R. Häggkvist and A. Thomason, Trees in tournaments, *Combinatorica* 11 (1991), 123–130.

[HäTh97] R. Häggkvist and A. Thomason, Oriented Hamiltonian cycles in oriented graphs, 339–353 in *Combinatorics, geometry, and probability (Cambridge, 1993)*, Cambridge University Press, 1997.

[Is95] G. Isaak, Tournaments as feedback arc sets, Electron. *J. Combin.* 2 (1995), 19.

[Ja72] O. S. Jakobsen, Cycles and paths in tournaments, *Thesis*, University of Aarhus, 1972.

[JiLu98] G. Jimenez and J. R. Lundgren, Tournaments which yield connected domination graphs, *Congressus Numer.* 131 (1998), 123–133.

[La53] H. G. Landau, On dominance relations and the structure of animal societies. III. The condition for a score structure, *Bull. Math. Biophys.* 15 (1953), 143–148.

[La97] J.-F. Laslier, *Tournament Solutions and Majority Voting*, Springer, 1997.

[LiSa96] V. Linek and B. Sands, A note on paths in edge-colored tournaments, *Ars Comb.* 44 (1996), 225–228.

[LuWaPa00] X. Lu, D.-W. Wang and J. Pan, Rooted spanning trees in tournaments, *Graphs and Combinatorics* 16 (2000), 411–427.

[Ma80] S. Maurer, The king chicken theorems, *Math. Mag.* 53 (1980), 67–80.

[Ma99] J. Mala, On λ-majority voting paradoxes, *Math. Soc. Sci.* 37 (1999), 39–44.

[Ma92] Y. Manoussakis, A linear-time algorithm for finding Hamiltonian cycles in tournaments, *Discrete Appl. Math.* 36 (1992), 199–201.

[Mi77] N. R. Miller, Graph-theoretical approaches to the theory of voting, *Amer. J. Polit. Sci.* 21 (1977), 769–803.

[Mi80] N. R. Miller, A new solution set for tournaments and majority voting: further graph-theoretical approaches to the theory of voting, *Amer. J. Polit. Sci.* 24 (1980), 68–96.

[Mi95] T. S. Michael, The ranks of tournament matrices, *Amer. Math. Monthly* 102 (1995), 637–639.

[Mo68] J. W. Moon, *Topics on Tournaments*, Holt, Rinehart, and Winston, 1968.

[MoMo62] J. W. Moon and L. Moser, Almost all tournaments are irreducible, *Canad. Math. Bull.* 5 (1962), 61–65.

[Ne94] V. Neumann Lara, A short proof of a theorem of Reid and Parker on tournaments, *Graphs and Combin.* 10 (1994), 363–366.

[PaRe70] E. T. Parker and K. B. Reid, Disproof of a conjecture of Erdös and Moser on tournaments, *J. Combin. Theory* 9 (1970), 225–238.

[Pe02] V. Petrovic, Claws in rotational tournaments, *Graphs and Combin.* (2002), 591–596.

[Pe84] V. Petrovic, Antidirected Hamiltonian circuits in tournaments, 259–269 in *Graph Theory, Novi Sad, 1983*, Univ. Novi Sad, Novi Sad, 1984.

[Re00] K. B. Reid, Monotone reachability in arc-colored tournaments, *Congressus Numer.* 146 (2000), 131–141.

[Ré34] L. Rédei, Ein kombinatorischer satz, *Acta Litterarum ac Scientiarum*, Szeged. 7 (1934), 39–43.

[Re69] K. B. Reid, On sets of arcs containing no cycles in a tournament, *Canad. Math. Bull.* 12 (1969), 261–267.

[Re73] K. B. Reid, Equivalence of n-tournaments via k-path reversals, *Discrete Math.* 6 (1973), 263–280.

[Re80] K. B. Reid, Tournaments with prescribed numbers of kings and serfs, *Congressus Num.* 29 (1980), 809–826.

[Re82] K. B. Reid, Every vertex a king, *Discrete Math.* 38 (1982), 93–98.

[Re84] K. B. Reid, Monochromatic reachability, complementary cycles and single arc reversals in tournaments, *Springer-Verlag Lecture Notes in Mathematics* 1073 (1984), 11–21.

[Re85] K. B. Reid, Two complementary cycles in two-connected tournaments, 171–211 in *Cycles in Graphs*, vol. 115 of *North-Holland Math. Stud.*, North-Holland, 1985.

[Re91a] K. B. Reid, Majority tournaments: Sincere and sophisticated voting decisions under amendment procedure, *Math. Soc. Sci.* 21 (1991), 1–19.

[Re91b] K. B. Reid, The relationship between two algorithms for decisions via sophisticated majority voting with an agenda, *Discrete Appl. Math.* 31 (1991), 23–28.

[Re96] K. B. Reid, Tournaments: scores, kings, generalizations and special topics, *Congressus Numer.* 115 (1996), 171–211.

[Re97] K. B.. Reid, Equitable agendas: agendas ensuring identical sincere and sophisticated voting decisions, *Social Choice and Welfare* 14 (1997), 363–378.

[ReBe79] K. B. Reid and L. W. Beineke, Tournaments, 169–204 in *Selected Topics in Graph Theory*, Academic Press, London, 1979.

[ReBr72] K. B. Reid and E. Brown, Doubly regular tournaments are equivalent to skew Hadamard matrices, *J. Combin. Theory* 12 (1972), 332–338.

[ReBr84] K. B. Reid and M. F. Bridgland, Stability of kings in tournaments, 117–128 in *Progress in Graph Theory*, Academic Press, 1984.

[ReMcHeHe02] K. B. Reid, A. A. McRae, S. M. Hedetniemi and S. T. Hedetniemi, Domination and irredundance in tournaments, preprint, 2002.

[ReWo83] K. B. Reid and N. Wormald, Embedding oriented n-trees in tournaments, *Studia Sci. Math. Hungarica* 18 (1983), 377–387.

[Ro74] M. Rosenfeld, Anti-directed Hamiltonian circuits in tournaments, *J. Combin. Theory B* 16 (1974), 234–242.

[Ry64] H. J. Ryser, Matrices of zeros and ones in combinatorial mathematics, 103–124 in *Recent Advances in Matrix Theory*, Univ. Wisconsin Press, 1964.

[Sá94] A. Sánchez-Flores, On tournaments and their largest transitive subtournaments, *Graphs and Combin.* 10 (1994), 367–376.

[SaSaWo82] B. Sands, N. Sauer and R. Woodrow, On monochromatic paths in edge-colored digraphs, *J. Combin. Theory B* 45 (1982), 108–111.

[Sh88], M. Shen, On monochromatic paths in m-coloured tournaments, *J. Combin. Theory B* (1988), 108–111.

[Sh92] B. L. Shader, On tournament matrices, *Linear Algebra Appl.* 162/164 (1992), 335–368.

[ShWe84] K. A. Shepsle and B. Weingast, Uncovered sets and sophisticated voting outcomes with implications for agenda institutions, *Amer. J. Polit. Sci.* 28 (1984), 49–74.

[ShYu01] I. Sharon (I. Charon-Fournier) and O. Yudri (O. Hudry), The branch and bound method for solving the problem of linear ordering on weighted tournaments (Russian), *Diskretn. Anal. Issled. Oper.* 8 (2001), 73–91.

[So93] Z. M. Song, Complementary cycles of all lengths in tournaments, *J. Combin. Theory B* (1993), 18–25.

[Sp71/72] J. Spencer, Optimal ranking of tournaments, *Networks* 1 (1971/72), 135–138.

[St59] R. Stearns, The voting problem, *Amer. Math. Monthly* 66 (1959), 761–763.

[SzSz65] E. Szekeres and G. Szekeres, On a problem of Schütte and Erdös, *Math. Gaz.* 49 (1965), 290–293.

[Te98] P. Tetali, A characterization of unique tournaments, *J. Combin. Theory B* 72 (1998), 157–159.

[Th73] C. Thomassen, Antidirected Hamiltonian circuits and paths in tournaments, *Math. Ann.* 201 (1973), 231–238.

[Th80] C. Thomassen, Hamiltonian-connected tournaments, *J. Combin. Theory B* 28 (1980), 142–163.

[Th82] C. Thomassen, Edge-disjoint Hamiltonian paths and cycles in tournaments, *Proc. London Math. Soc.* 45 (1982), 151–168.

[Th84] C. Thomassen, Connectivity in tournaments, 305–313 in *Graph Theory and Combinatorics*, Cambridge, 1983, Academic Press, 1984.

[Th85] C. Thomassen, Hamiltonian circuits in regular tournaments, *Annals Discrete Math.* 27 (1985), 159–162.

[Th86] A. Thomason, Paths and cycles in tournaments, *Trans. Amer. Math. Soc.* 296 (1986), 167–180.

[Th88] C. Thomassen, Arc reversals in tournaments, *Discrete Math.* 71 (1988), 73–86.

[Va52] H. E. Vaughan, On well-ordered subsets and maximal elements of ordered sets, *Pacific J. Math.* 2 (1952), 407–412.

[Vo02] L. Volkman, Cycles in multipartite tournaments: results and problems, *Discrete Math.* 245 (2002), 19–53.

[Wa84] K. Wayland, Getting your chickens elected, *Congressus Numer.* 45 (1984), 311–318.

[Ya88] T. X. Yao, Reid's conjecture on score sets in tournaments (in Chinese), *Kexue Tongbao* 33 (1988), 481–484.

[Ya89] T. X. Yao, On Reid conjecture of score sets for tournaments, *Chinese Sci. Bull.* 34 (1989), 804–808.

[Zh80] C. Q. Zhang, Every regular tournament has two arc-disjoint Hamiltonian cycles, *J. Qufu Normal College, Special Issue Oper. Res.* (1980), 70–81.

[ZhSo91] K.-M. Zhang and Z.-S. Song, Cycles in digraphs – a survey, *J. Nanjing Univ., Natural Sci.* 27 (1991) 188–215.

GLOSSARY FOR CHAPTER 3

Activity on Arc: a digraph scheduling model in which arcs represent subtasks to be scheduled as part of a large project.

Activity on Node: a digraph scheduling model in which nodes represent subtasks to be scheduled as part of a large project.

adjacency matrix – of a digraph: the $|V| \times |V|$ matrix in which the ij entry is the number of arcs from v_i to v_j.

agenda – in voting: an ordered list of alternatives (i.e., an ordered list of the vertices of a majority tournament).

almost regular tournament (or *near regular*): see *regular tournament.*

amendment procedure – for voting: a sequential voting process in which, given an agenda $(a_1, a_2, \ldots, a_n)$ of alternatives, alternative a_1 is pitted against a_2 in the first vote, then the winner is pitted against a_3 in the second vote, then the winner is pitted against a_4 in the third vote, etc.

ancestor of a vertex v – in a rooted tree: a vertex that lies on the unique path from v to the root; see also *descendant.*

anti-directed cycle – in a digraph D: a sequence of arcs that forms a cycle in the underlying graph of D but does not contain a directed path of length 2 in D.

anti-directed path – in a digraph D: a sequence of arcs that forms a path in the underlying graph of D but does not contain a directed path of length 2 in D.

antisymmetric relation R: one in which, for all x, y, if xRy and yRx, then $x = y$.

AoA: Activity on Arc.

AoN: Activity on Node.

arc: see *directed edge.*

arc-cut: synonym for edge-cut.

basis of a digraph: a minimal set of vertices such that every other vertex can be reached from some vertex in this set by a directed path.

beats: synonym for *dominates.*

binary tree: an ordered tree in which each vertex has at most two children, and each child is designated either a *left-child* or a *right-child.*

___, **balanced**: a binary tree such that for every vertex, the number of vertices in its left and right subtrees differ by at most one.

binary-search tree (BST): a binary tree, each of whose vertices is assigned a key, such that the key assigned to any vertex v is greater than the key at each vertex in the left subtree of v, and is less than the key at each vertex in the right subtree of v.

child of vertex v – in a rooted tree: a vertex to which there is an edge from v; see also *parent.*

___, **left** – in a binary tree: a child which is designated to be on the left, whether or not there is another child.

comparability digraph – of a poset $(X, \preceq)$: the digraph with vertex set X such that there is an arc from x to y if and only if $x \preceq y$.

comparable elements – of a poset $(X, \preceq)$: elements x, y such that either $x \preceq y$ or $y \preceq x$.

complete m-ary tree: an m-ary tree in which every internal vertex has exactly m children and all leaves are at the same level.

complete digraph: a simple digraph such that between each pair of its vertices, there is an arc in both directions.

condensation – of a digraph G whose strong components are $S_1, S_2, \ldots, S_r$: a digraph G^* with vertex-set $V_{G^*} = \{s_1, s_2, \ldots, s_r\}$ such that $(s_i, s_j) \in E(G^*)$ if and only if there is an arc in digraph G from a vertex in component S_i to a vertex in component S_j.

condensation of a tournament T: a tournament T^* whose vertex-set $\{u_1, u_2, \ldots, u_k\}$ corresponds to a vertex partition $\{V_1, V_2, \ldots, V_k\}$ of $V(T)$, where each V_i induces a maximal strongly connected sub-tournament of T, and in which vertex u_i dominates u_j whenever all of the vertices in V_i dominate all of the vertices in V_j in T.

Condorcet paradox: the possibility that the voters may be consistent in their preferences (i.e., each of their rankings of the n candidates is a linear order), but the amalgamation of voters' preferences using majority rule can result in inconsistencies (i.e., cycles in the majority digraph).

Condorcet winner – in voting: a candidate (or alternative) x such that for every other candidate (or alternative) y, x is preferred over y by a majority of the voters.

connectivity (or *vertex-connectivity*) – of a non-complete digraph: the minimum size of a vertex subset S such that $G - S$ is neither strongly connected nor the trivial digraph. (The connectivity of a complete n-vertex digraph is $n-1$.) Denoted $\kappa_v(G)$ or $\kappa(G)$. Synonyms for vertex-cut are *cut* and *disconnecting set.*

consistent set of arcs – in a digraph D: a set of arcs that induces an acyclic subdigraph of D.

cost flow network: see *network.*

cover graph – of a poset $(X, \preceq)$: the graph with vertex set X such that x, y are adjacent if and only if one of them covers the other.

covering – in a poset $(X, \preceq)$: the element y *covers* the element x if $x \prec y$ and there is no element z such that $x \prec z \prec y$.

CPM: Critical Path Method.

Critical Path Method: a method for scheduling models where subtasks have fixed times and precedence is known. The whole project is modeled as an AoA or AoN digraph, and an single-pass iterative algorithm is used to find the longest path from start to finish.

DAG: a directed acyclic graph.

decision tree – for a sincere decision corresponding to a given majority n-tournament T and an agenda $(a_1, a_2, \ldots, a_n)$ of alternatives: the spanning, rooted subtree of T, rooted at the sincere decision, induced by the $n-1$ arcs of T which describe the $n-1$ votes taken in the amendment procedure using T and $(a_1, a_2, \ldots, a_n)$.

depth (or *level*) – of a vertex v in a rooted tree: the length (i.e., number of arcs) of the unique directed path from the root to v.

descendant of a vertex v – in a rooted tree: a vertex w such that v is on the unique path from the root to w; the vertex v is called an *ancestor* of w.

digraph: a directed graph.

___, **acyclic**: a digraph with no directed cycles, i.e., a directed acyclic graph, a *DAG*.

___, **connected**: a digraph whose underlying graph is connected. The term *weakly connected* is also used.

___, **representation of a relation** R on a finite set S: the digraph whose vertices correspond to the elements of S, and whose arcs correspond to the ordered pairs in the relation.

___, **weak**: short form of *weakly connected* digraph.

___, **weakly connected**: a digraph whose underlying graph is connected; synonym for *connected digraph.*

directed acyclic graph: a digraph without directed cycles.

Directed Chinese Postman Problem: to find a minimum-weight postman tour in a given weighted digraph.

directed cycle: a closed directed path.

directed edge (or *arc*): an edge e, one of whose endpoints is designated as the *tail*, and whose other endpoint is designated as the *head.* In a line drawing, the arrow points toward the head.

directed path: a path in a digraph or partial digraph in which all edges are oriented in the same direction.

directed tree: a digraph whose underlying graph is a tree.

directed walk – from v_0 to v_n: an alternating sequence $\langle v_0, e_1, v_1, e_2, ..., v_{n-1}, e_n, v_n \rangle$ of vertices and arcs, such that $tail(e_i) = v_{i-1}$ and $head(e_i) = v_i$, for $i = 1, 2 \ldots, n$. Also called a v_0-v_n directed walk.

division tree – in voting: see §3.3, Definition 37.

dominating set – in a tournament T: a set S of vertices in T such that every vertex not in S is dominated by some vertex in S.

domination (or *beating*)– a vertex y in a tournament: a property that a vertex x has if there is an arc from x to y.

___, **graph** – of a tournament T: an undirected graph G that has the same vertex-set as T, and x is adjacent to y in G whenever $\{x, y\}$ is a dominating set in T.

___, **number** – of a tournament: the minimum cardinality of a dominating set in T; denoted $\gamma(T)$.

doubly-regular tournament: see *regular tournament.*

edge-connectivity – of a non-trivial digraph: the minimum size of an edge subset F such that $G - F$ is not strongly connected. Denoted $\kappa_e(G)$ or $\lambda(G)$.

edge-cut (or *arc-cut*) – in a strongly connected digraph: an arc subset whose deletion results in a digraph that is not strongly connected. Synonyms are *edge-disconnecting set*, *arc-disconnecting set*, and *cut-set.*

eulerian tour of a digraph G: a closed directed walk that uses each arc *exactly* once.

feedback set of arcs – in a tournament T: a set S of arcs such that the digraph $T - S$ contains no cycle.

flow network: see *network.*

frontier arc – relative to a rooted tree T in a digraph: an arc whose tail is in T and whose head is not in T.

functional graph: a digraph in which each vertex has outdegree 1.

hamiltonian cycle (or *spanning cycle*) – in a digraph D: a cycle that includes all vertices of D.

hamiltonian path (or *spanning path*) – in a digraph D: a directed path that includes all vertices of D.

Hasse diagram – of a poset: a straight-line drawing of the cover graph such that the lesser element of each adjacent pair is lower in the drawing.

head: see *directed edge.*

height – of a rooted tree: the length of a longest path from the root.

in-branching (or *in-tree*) – in a digraph: a rooted spanning tree with all the arcs reversed.

in-score – of a vertex v in a tournament T: the number of vertices that dominate v (i.e., its *indegree*; denoted $d_T^-(v)$ (or $d^-(v)$ when T is understood).

in-set – of a vertex x in a digraph D: the set of all vertices that dominate x; denoted $I(x)$.

internal vertex – in a tree or rooted tree: a non-leaf.

in-tree: synonym for *in-branching.*

k**-strong tournament**: see *strong tournament.*

king – in a tournament T: a vertex x such that for every other vertex y, there is a 1-path or a 2-path from x to y in T.

leaf – in a rooted tree: a vertex with outdegree 0.

left subtree – of a vertex v in a binary tree: the binary subtree spanning the left-child of v and all of its descendants.

length of a directed walk: the number of arc-steps in the walk sequence.

level of a vertex – in a rooted tree: synonym for depth.

linear extension ordering – of a digraph: a consecutive numbering of the vertices as $v_1, v_2, \ldots, v_n$ so that all arcs go from lower-numbered to higher-numbered vertices.

linear ordering: a consecutive numbering.

m**-ary tree**: see *rooted tree.*

majority digraph D – of a set of n-tournaments, all with the same vertex-set V: a digraph with vertex-set V and such that vertex x dominates vertex y in D if and only if x dominates y in a majority of the n-tournaments.

Markov digraph: a complete digraph with a self-loop at each vertex and whose arcs are assigned probabilities such that the out-probabilities at each vertex sum to one; models a stationary Markov chain.

maximum-flow problem: to determine the maximum flow that can be moved through an s-t network from source s to sink t such that the flow into each intermediate node equals the flow out (*conservation of flow*) and the flow across any arc does not exceed the capacity of that arc.

minimum-cost-flow problem: to find an assignment of *flows* on the arcs of the flow network that satisfy the supply and demand (negative supply) requirements at minimum cost.

mutually reachable vertices – in a digraph G: vertices that have a directed walk from one to the other and vice versa. Every vertex is regarded as mutually reachable with itself (via the trivial walk).

neighborhood: see *out-set.*

network: a digraph $G = (V, E)$ used to model a variety of network flow problems; vertices might have supply or demand, and arcs might have capacities and or flow costs.

___, *s-t* **flow**: a network $G = (V, E, cap, s, t)$ with a nonnegative capacity function $cap : E \to N$, a distinguished vertex s, called the *source*, with nonzero outdegree, and a distinguished vertex t, called the *sink*, with nonzero indegree.

___, **capacitated cost flow** $G = (V, E, cap, c, b)$: a directed graph with vertex-set V, arc-set E, a nonnegative capacity function $cap : E \to N$, a linear cost function $c : E \to Z$, and an integral supply vector $b : V \to Z$ that satisfies $\sum\limits_{w \in V} b(w) = 0$.

___, **cost flow**: a network $G = (V, E, cap, c, b)$ with nonnegative capacity function $cap : E \to N$, a linear cost function $c : E \to Z$, and an integral supply vector $b : V \to Z$ that satisfies $\sum\limits_{w \in V} b(w) = 0$.

order of a tournament: the number of vertices it contains. A tournament of order n is an *n-tournament.*

ordered tree: a rooted tree in which the children of each vertex are assigned a fixed ordering.

ordering: a linear ordering.

orientation – of a graph: an assignment of directions to its edges, thereby making it a digraph.

oriented graph: a digraph obtained by choosing an orientation for each edge of an undirected simple graph.

out-branching (or *out-tree*) – in a digraph: synonym for rooted spanning tree.

out-set (or *neighborhood*) – of a vertex x in a digraph D: the set of all vertices that x dominates; denoted $O(x)$ or $N^+(x)$ (or with a subscripted "D" if necessary).

out-tree: a rooted tree, especially when the arc directions are shown explicitly.

parent of a vertex w – in a rooted tree: a vertex v that immediately precedes w on the path from the root to w; also, w is the *child* of v.

partial order: a binary relation $\preceq$ on a set X that is reflexive, antisymmetric, and transitive.

partially ordered set: a pair $(X, \preceq)$ consisting of a set X and a partial order $\preceq$ on X.

path in a digraph: a directed path.

___, *k*-: a directed path of length k.

poset: a partially ordered set.

postman tour (or *covering walk*): a closed directed walk that uses each arc *at least* once.

proper arc-coloring – of a digraph: an assignment of colors to the arcs such that any two arcs that have an endpoint in common are assigned different colors.

receiver – in a tournament: a vertex that is dominated by every other vertex in a tournament.

reflexive relation R: one in which, for all x, xRx.

regular tournament: a tournament T in which all scores are the same.

___, **almost** (or *near*): a tournament T in which $\max_{v \in V(T)} \{|d^+(v) - d^-(v)|\} = 1$.

___, **doubly-**: a tournament in which all pairs of vertices jointly dominate the same number of vertices (i.e., there is an integer k so that $|O(x) \cap O(y)| = k$, for all distinct pairs of vertices x and y in T).

right child – in a binary tree: a child which is designated to be on the right, whether or not there is another child.

right subtree – of a vertex v in a binary tree: the binary subtree spanning the right-child of v and all of its descendants.

root: see *rooted tree.*

rooted tree: a directed tree having a distinguished vertex r, called the *root*, such that for every other vertex v, there is a directed r-v path. Occasionally encountered synonyms for rooted tree are *out-tree*, *branching*, and *arborescence.*

___, ***m*-ary**: a rooted tree in which every vertex has m or fewer children; also called an *m-ary tree.*

rotational tournament: denoted $R_G(S)$, or simply $R(S)$ if the group G is understood; see §3.3, Definition 11.

***s*-*t* flow network**: see *network.*

score of a vertex v in a tournament T: the number of vertices that v dominates (i.e., its *outdegree*). Denoted $d_T^+(v)$ (or $d^+(v)$ when T is understood).

score sequence (or *score vector*) – of an n-tournament: the ordered n-tuple $(s_1, s_2, \ldots, s_{n-1}, s_n)$, where s_i is the score of vertex v_i, $1 \leq i \leq n$, and $s_1 \leq s_2 \leq \ldots \leq s_{n-1} \leq s_n$.

score vector: synonym for *score sequence.*

second neighborhood – of a vertex x in a digraph D: the set of all vertices of D reachable from x by a 2-path but not a 1-path; denoted $N_D^{++}(x)$.

serf – in a tournament T: a vertex x such that for every other vertex y, there is a 1-path or a 2-path from y to x.

siblings – in a rooted tree: children of the same parent.

simple digraph: a digraph with no self-loops and no multi-arcs.

simple digraph: a digraph with no self-loops or multi-arcs.

sincere decision – for a given majority n-tournament T and an agenda $(a_1, a_2, \ldots, a_n)$ of alternatives given by the vertices of T: the alternative surviving the last vote (i.e., the $(n-1)^{\text{th}}$ vote) in an amendment procedure of voting using majority voting at each stage.

sink – in a digraph: a vertex of outdegree zero.

sophisticated decision – in voting: see §3.3, Definition 38.

source – in a digraph: a vertex of indegree zero.

spanning subgraph – of a graph or digraph: a subgraph that includes all the vertices of the original graph.

spiked cycle: a connected (undirected) graph with the property that when all vertices of degree 1 are removed, a cycle results.

standard plane representation of an ordered tree: a standard plane drawing of the tree such that at each level, the left-to-right order of the vertices agrees with their prescribed order.

strong component – of a digraph G: maximal strongly connected subdigraph of G.

strong digraph: short form of *strongly connected digraph.*

strong orientation – of a graph: an orientation that results in a strong digraph.

strong tournament: a tournament that is a *strongly connected* digraph.

___, k-: a strong tournament such that the removal of any set of $k-1$ or fewer vertices results in a strong digraph.

strongly connected digraph: a digraph in which every two vertices are mutually reachable, i.e., there is a directed path from each of the two vertices to the other.

strongly orientable graph a graph for which there exists an assignment of directions to the edges such that the resulting digraph is strongly connected.

symbol set – for a rotational tournament: see §3.3, Definition 11.

tail: see *directed edge.*

topological sort or *topsort*: any algorithm that assigns a linear extension ordering to a digraph when it has one.

topsort: short form of *topological sort.*

tournament matrix: a square matrix $M = (m_{ij})$ of 0's and 1's, with 0's on the main diagonal and $m_{ij} + m_{ji} = 1$, for all distinct i and j (i.e., the adjacency matrix of some tournament).

tournament: a simple digraph such that between each pair of vertices there is exactly one arc.

___, **irreducible**: a tournament that is not a *reducible tournament.*

___, **quadratic residue**: a special rotational tournament; see §3.3, Definition 12.

___, **reducible**: a tournament whose vertex-set can be partitioned into two non-empty subsets V_1 and V_2 such that every vertex in V_1 dominates every vertex in V_2.

___, k**-stable**: a tournament in which every vertex is a king and more than k arcs must be reversed in order to reduce the number of kings, where $k \geq 1$.

___, n-: a tournament of order n, i.e., an n-vertex tournament.

transitive closure – of a graph of digraph D: the smallest supergraph of D that is transitive.

transitive digraph: a digraph in which, if (u, v) and (v, w) are arcs, then so is (u, w).

transitive orientation – of a graph: an orientation that results in a transitive digraph.

transitive relation R: a relation in which, for all x, y, z, if xRy and yRz, then xRz.

transitive tournament: a tournament such that for every set of three distinct vertices x, y, and z, if x dominates y, and y dominate z, then x dominates z.

transmitter – in a tournament: a vertex that dominates every other vertex in a tournament.

unilateral digraph: a digraph in which, for all pairs of vertices u, v, there is a directed path between them in at least one direction.

vertex-cut – in a strongly connected digraph: a vertex subset whose deletion results in a digraph that is not strongly connected.

weights – in a graph or digraph: numbers on the vertices or edges or arcs, often representing something that is to be maximized or minimized.

Chapter 4

CONNECTIVITY and TRAVERSABILITY

4.1 CONNECTIVITY: PROPERTIES AND STRUCTURE

Josep Fàbrega, Technical University of Catalonia, Spain
Miguel Angel Fiol, Technical University of Catalonia, Spain

4.2 EULERIAN GRAPHS

Herbert Fleischner, Technical University of Vienna, Austria

4.3 CHINESE POSTMAN PROBLEMS

R. Gary Parker, Georgia Institute of Technology

4.4 DEBRUIJN GRAPHS and SEQUENCES

A. K. Dewdney, University of Waterloo, Canada

4.5 HAMILTONIAN GRAPHS

Ronald J. Gould, Emory University

4.6 TRAVELING SALESMAN PROBLEMS

Gregory Gutin, Royal Holloway, University of London, UK

4.7 FURTHER TOPICS in CONNECTIVITY

Josep Fàbrega, Technical University of Catalonia, Spain
Miguel Angel Fiol, Technical University of Catalonia, Spain

GLOSSARY

4.1 CONNECTIVITY: PROPERTIES AND STRUCTURE

Josep Fàbrega, Technical University of Catalonia, Spain
Miguel Angel Fiol, Technical University of Catalonia, Spain

4.1.1 Connectivity Parameters
4.1.2 Characterizations
4.1.3 Structural Connectivity
4.1.4 Analysis and Synthesis
References

Introduction

Connectivity is one of the central concepts of graph theory, both from a theoretical and a practical point of view. Its theoretical implications are mainly based on the existence of nice *max-min* characterization results, such as Menger's theorems. In these theorems, one condition which is clearly necessary also turns out to be sufficient. Moreover, these results are closely related to some other key theorems in graph theory: Ford and Fulkerson's theorem about flows and Hall's theorem on perfect matchings. With respect to the applications, the study of connectivity parameters of graphs and digraphs is of great interest in the design of reliable and fault-tolerant interconnection or communication networks.

Since graph connectivity has been so widely studied, we limit ourselves here to the presentation of some of the key results dealing with finite simple graphs and digraphs. For results about infinite graphs and connectivity algorithms the reader can consult, for instance, [AhDi94], [Ha00], and [Gi85], [Wi92], [HeRaGa00]. For further details, we refer the reader to some of the good textbooks and surveys available on the subject: [Tu66], [Be76], [Ma79], [BeHoPe89], [Fr90], [Fr94], [Fr95], [Lo93], [Oe96], [GrYe99].

4.1.1 Connectivity Parameters

In this first subsection the basic notions of connectivity and edge-connectivity of simple graphs and digraphs are reviewed.

NOTATION: Given a graph or digraph G, the vertex-set and edge-set are denoted $V(G)$ and $E(G)$, respectively. Often, when there is no ambiguity, we omit the argument and refer to these sets as V and E.

Preliminaries

DEFINITIONS

D1: A graph is ***connected*** if there exists a walk between every pair of its vertices. A graph that is not connected is called ***non-connected***.

D2: The subgraphs of G which are maximal with respect to the property of being connected are called the ***components*** of G.

D3: Let $G = (V, E)$ be a graph and $U \subset V$. The ***vertex-deletion subgraph*** $G - U$ is the graph obtained from G by deleting from G the vertices in U. That is, $G - U$ is the subgraph induced on the vertex subset $V - U$. If $U = \{u\}$, we simply write $G - u$.

D4: Let $G = (V, E)$ be a graph and $F \subset E$. The ***edge-deletion subgraph*** $G - F$ is the subgraph obtained from G by deleting from G the edges in F. Thus, $G - F = (V, E - F)$. As in the case of vertex deletion, if $F = \{e\}$, it is customary to write $G - e$ rather than $G - \{e\}$.

D5: A ***disconnecting (vertex-)set*** (or ***vertex-cut***) of a connected graph G is a vertex subset U such that $G - U$ has at least two different components.

D6: A vertex v is a ***cut-vertex*** of a connected graph G if $\{v\}$ is a disconnecting set of G.

D7: A ***disconnecting edge-set*** (or ***edge-cut***) of a connected graph G is an edge subset F such that $G - F$ has at least two different components.

D8: An edge e is a ***bridge*** (or ***cut-edge***) of a connected graph G if $\{e\}$ is a disconnecting edge-set of G.

FACTS

F1: Every nontrivial connected graph contains at least two vertices that are not cut-vertices.

F2: An edge is a bridge if and only if it lies on no cycle.

Vertex- and Edge-Connectivity

The simplest way of quantifying connectedness of a graph is by means of its parameters *vertex-connectivity* and *edge-connectivity*.

DEFINITIONS

D9: The ***(vertex-)connectivity*** $\kappa(G)$ of a graph G is the minimum number of vertices whose removal from G leaves a non-connected or trivial graph.

D10: The ***edge-connectivity*** $\lambda(G)$ of a nontrivial graph G is the minimum number of edges whose removal from G results on a non-connected graph.

NOTATION: When the context is clear, we suppress the "G" and simply use κ and λ.

NOTATION: In some other sections of the *Handbook*, $\kappa_v(G)$ and $\kappa_e(G)$ are used instead of $\kappa(G)$ and $\lambda(G)$.

EXAMPLE

E1: Figure 4.1.1 shows an example of a graph with $\kappa = 2$ and $\lambda = 3$.

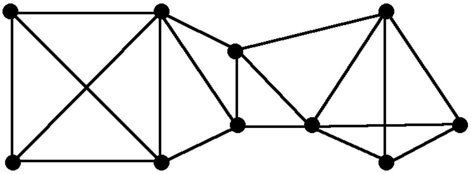


Figure 4.1.1 $\kappa = 2$ **and** $\lambda = 3$.

FACTS

F3: $\kappa = 0$ if and only if G is non-connected or $G = K_1$. If G has order n, then $\kappa = n - 1$ if and only if G is the complete graph K_n. In this case, the removal of $n - 1$ vertices results in the trivial graph K_1. Moreover, if $G \neq K_n$ is a connected graph, then $1 \leq \kappa \leq n - 2$ and there exists a disconnecting set U of κ vertices.

F4: If $G \neq K_1$ we have $\lambda = 0$ if G is non-connected. By convention, we set $\lambda(K_1) = 0$.

F5: If $G \neq K_1$ is connected, then the removal of a λ edges results in a non-connected graph with precisely two components.

F6: The parameters κ and λ can be computed in polynomial time.

Relationships Among the Parameters

NOTATION: The ***minimum degree*** of a graph G is denoted $\delta(G)$. When the context is clear, we simply write δ. (In some other sections of the *Handbook*, the notation $\delta_{min}(G)$ is used.)

FACTS

F7: [Wh32] For any graph, $\kappa \leq \lambda \leq \delta$.

F8: [ChHa68] For all integers a, b, c such that $0 < a \leq b \leq c$, there exists a graph G with $\kappa = a$, $\lambda = b$, and $\delta = c$.

DEFINITIONS

D11: G is ***maximally connected*** when $\kappa = \lambda = \delta$, and G is ***maximally edge-connected*** when $\lambda = \delta$.

D12: A graph G with connectivity $\kappa \geq k \geq 1$ is called ***k-connected***. Equivalently, G is k-connected if the removal of fewer than k vertices leaves neither a non-connected graph nor a trivial one. Analogously, if $\lambda \geq k \geq 1$, G is said to be ***k-edge-connected***.

D13: A connected graph without cut-vertices is called a ***block***.

Some Simple Observations

The following facts are simply restatements of the definitions.

FACTS

F9: A nontrivial graph is 1-connected if and only if it is connected.

F10: If G is k-connected, either $G = K_{k+1}$ or it has at least $k + 2$ vertices and $G - U$ is still connected for any $U \subset V$ with $|U| < k$.

F11: G is k-edge-connected if the deletion of fewer than k edges does not disconnect it.

F12: Every block with at least three vertices is 2-connected.

Internally-Disjoint Paths and Whitney's Theorem

DEFINITIONS

D14: An ***internal vertex*** of a path is a vertex that is neither the initial nor the final vertex of that path.

D15: The paths $P_1, P_2, \ldots, P_k$ joining the vertices u and v are said to be ***internally-disjoint*** (or ***openly-disjoint***) u–v paths if no two paths in the collection have an internal vertex in common. Thus, $V(P_i) \cap V(P_j) = \{u, v\}$ for $i \neq j$.

FACTS

F13: [Whitney] A graph G with order $n \geq 3$ is 2-connected if and only if any two vertices of G are joined by at least two internally-disjoint paths.

F14: Whitney's theorem implies that every 2-connected graph is a block.

F15: A graph G with at least three vertices is a block if and only if every two vertices of G lie on a common cycle.

Strong Connectivity in Digraphs

For basic concepts on digraphs, see, for example, the textbooks [ChLe96], [HaNoCa68], [BaGu01].

DEFINITIONS

D16: In a digraph G, vertices u and v are ***mutually reachable*** if G contains both a directed u–v walk and a directed v–u walk.

D17: A digraph G is said to be ***strongly connected*** if every two vertices u and v are mutually reachable.

D18: For a strongly connected digraph G, the ***(vertex) connectivity*** $\kappa = \kappa(G)$ is defined as the minimum number of vertices whose removal leaves a non-strongly connected or trivial digraph. Analogously, if G is not trivial, its ***edge-connectivity*** $\lambda = \lambda(G)$ is the minimum number of directed edges (or *arcs*) whose removal results in a non-strongly connected digraph.

D19: Let G be an undirected graph. The ***associated symmetric digraph*** $\overset{\leftrightarrow}{G}$ is the digraph obtained from G by replacing each edge $uv \in E(G)$ by the two directed edges (u, v) and (v, u) forming a "digon".

REMARKS

R1: In our context, the interest for studying digraphs is that we can deal with an undirected graph G by considering $\overset{\leftrightarrow}{G}$. In particular, $\kappa(\overset{\leftrightarrow}{G}) = \kappa(G)$, and, since a minimum edge-disconnecting set cannot contain digons, we also have $\lambda(\overset{\leftrightarrow}{G}) = \lambda(G)$.

NOTATION: δ^+ and δ^- denote the minimum outdegree and indegree among the vertices of a digraph G. Then, the ***minimum degree*** of G is defined as $\delta = \min\{\delta^+, \delta^-\}$.

R2: Note that, if G is a strongly connected digraph, then $\delta \geq 1$. The following result, due to Geller and Harary, is the analogue of (and implies) Fact 7.

FACT

F16: [GeHa70] For any digraph G, $\kappa \leq \lambda \leq \delta$.

TERMINOLOGY: A digraph G is said to be ***maximally connected*** when $\kappa = \lambda = \delta$, and G is ***maximally edge-connected*** when $\lambda = \delta$.

An Application to Interconnection Networks

The interconnection network of a communication or distributed computer system is usually modeled by a (directed) graph in which the vertices represent the switching elements or processors, and the communication links are represented by (directed) edges. *Fault-tolerance* is one of the main factors that have to be taken into account in the design of an interconnection network. See, for instance, the survey of Bermond, Homobono, and Peyrat [BeHoPe89]. Indeed, it is generally expected that the system be able to work even if several of its elements fail. Thus, it is often required that the (di)graph associated to the interconnection network be sufficiently connected, and, in most cases, a good design requires that this (di)graph has maximum connectivity. Communication networks are discussed in §11.4 of the *Handbook*.

4.1.2 Characterizations

When a graph G is k-connected we need to delete at least k vertices to disconnect it. Clearly, if any pair u, v of vertices can be joined by k internally-disjoint u–v paths, G is k-connected. It turns out that the converse statement is also true. That is, in a k-connected graph any two vertices can be joined by k internally-disjoint paths. We review in this subsection some key theorems of this type that characterize k-connectedness.

Menger's Theorems

DEFINITION

D20: Let u and v be two non-adjacent vertices of a connected graph $G \neq K_n$. A $(u|v)$***-disconnecting set*** X, or simply $(u|v)$***-set***, is a disconnecting set $X \subset V - \{u, v\}$ whose removal from G leaves u and v in different components.

NOTATION: For any pair of non-adjacent vertices u and v, $\kappa(u|v)$ denotes the minimum number of vertices in a $(u|v)$-set.

NOTATION: For any two vertices u and v, $\kappa(u{-}v)$ denotes the maximum number of internally-disjoint $u{-}v$ paths.

FACTS

F17: For any graph G, the connectivity $\kappa = \min\{\kappa(u|v) : u, v \in V$, nonadjacent$\}$.

F18: [Menger's theorem, Me27] For any pair of non-adjacent vertices u and v,

$$\kappa(u{-}v) = \kappa(u|v)$$

F19: Although $\kappa(u{-}v)$ can be arbitrarily smaller than the minimum of the degrees of u and v, Mader proved that every finite graph contains vertices for which equality holds:

F20: [Ma73] Every connected non-trivial graph contains adjacent vertices u and v for which $\kappa(u{-}v) = \min\{deg(u), deg(v)\}$.

NOTATION: For any pair of distinct vertices u and v, $\lambda(u|v)$ denotes the minimum number of edges whose removal from G (G non-trivial) leaves u and v in different components and $\lambda(u{-}v)$ denotes the maximum number of *edge-disjoint* $u{-}v$ paths.

F21: For any non-trivial graph G, $\lambda(G) = \min\{\lambda(u|v),\ u, v \in V\}$.

F22: (Edge-analogue of Menger's theorem) [ElFeSh56,FoFu56] $\lambda(u{-}v) = \lambda(u|v)$.

REMARKS

R3: Digraph versions of Menger's theorems are the same except that all paths are directed paths.

R4: The edge form and arc form of Menger's theorem were proved by Ford and Fulkerson [FoFu56] using *network-flow* methods. Network flow is discussed in Chapter 11.

Other Versions and Generalizations of Menger's Theorem

In addition to the ones given below, there exist other versions and generalizations of Menger's theorem, see for example [Di97], [Fr95], [McCu84]. A comprehensive survey about variations of Menger's theorem can be found in [Oe03].

DEFINITIONS

D21: Given $A, B \subset V$, an $A{-}B$ ***path*** is an $u{-}v$ path P such that u is the only vertex of P belonging to A, and v is the only vertex of P that belongs to B.

D22: A set $X \subset V$ ***separates*** A from B (or is $(A|B)$***-separating***) if every $A{-}B$ path in G contains a vertex of X.

D23: An A***-path*** is an $A{-}B$ path with $A = B$.

D24: A subset $X \subset V - A$ ***totally separates*** A if each component of $G - X$ contains at most one vertex of A (or, equivalently, every A-path contains some vertex of X).

D25: A vertex subset is an ***independent set*** if no two of its vertices are adjacent.

NOTATION: The maximum number of (internally-)disjoint $A{-}B$ paths is denoted $\kappa(A{-}B)$, and the size of a minimum $(A|B)$-separating set is denoted $\kappa(A|B)$.

FACTS

F23: The minimum number of vertices separating A from B is equal to the maximum number of disjoint A–B paths. That is, $\kappa(A{-}B) = \kappa(A|B)$.

F24: If A is an independent set, then maximum number of internally-disjoint A-paths is at most the minimum number of vertices in a totally A-separating set, i.e., $\kappa(A{-}A) \leq \kappa(A|A)$.

F25: The corresponding Menger-type result does not hold and inequality can be strict. In fact, there exist examples for which $\kappa(A{-}A) = \kappa(A|A)/2$.

F26: Gallai [Ga61] conjectured that Fact 25 corresponds to the "extremal" situation and that always $\kappa(A{-}A) \geq \kappa(A|A)/2$, and Lovász [Lo76] conjectured that $\lambda(A{-}A) \geq \lambda(A|A)/2$. Both conjectures were proved by Mader.

F27: [Ma78b,Ma78c] $\kappa(A{-}A) \geq \kappa(A|A)/2$ and $\lambda(A{-}A) \geq \lambda(A|A)/2$.

REMARK

R5: The classical version of Menger's theorem (Fact 18) is easily derived from Fact 23 by taking A and B as the sets of vertices adjacent to u and v, respectively.

Another Menger-Type Theorem

NOTATION: For any pair of vertices u and v, $\kappa_n(u{-}v)$ denotes the maximum number of internally-disjoint u–v paths of length less than or equal to n. For any pair of non-adjacent vertices u and v, $\kappa_n(u|v)$ denotes the minimum number of vertices of a set $X \subset V - \{u, v\}$ such that every u–v path in $G - X$ has length greater than n.

FACTS

F28: There are examples for which we have the strict inequality $\kappa_n(u{-}v) < \kappa_n(u|v)$. However, for $n = d(u, v) \geq 2$ (i.e., for shortest u–v paths), we have $\kappa_n(u{-}v) = \kappa_n(u|v)$. This Menger-type result is equivalently restated as Fact 29.

F29: [EnJaSl77,LoNePl78] The maximum number of internally-disjoint shortest u–v paths is equal to the minimum number of vertices (different from u and v) necessary to destroy all shortest u–v paths.

Whitney's Theorem

In a connected graph, there exists a path between any pair of its vertices, and if the graph is 2-connected, then there exists at least two internally-disjoint paths between two distinct vertices (Fact 13). As a corollary of Menger's theorem, we have the remarkable result that this property can be generalized to k-connected graphs, which was independently proved by Whitney. It provides a natural and intrinsic characterization of k-connected graphs.

FACTS

F30: [Whitney's theorem, Wh32] A non-trivial graph G is k-connected if and only if for each pair u, v of distinct vertices there are at least k internally-disjoint u–v paths (or, alternatively, if and only if every cut-set has at least k vertices).

F31: (Edge version of Whitney's theorem) A nontrivial graph G is k-edge-connected if and only if for each pair u, v of distinct vertices there exist at least k edge-disjoint $u{-}v$ paths.

F32: (The Fan Lemma) Let G be a k-connected graph ($k \geq 1$). Let $v \in V$ and let $B \subset V$, $|B| \geq k$, $v \notin B$. Then there exist distinct vertices $b_1, b_2, \ldots, b_k$ in B and a $v{-}b_i$ path P_i, $i = 1, 2 \ldots, k$, such that the paths $P_1, P_2, \ldots, P_k$ are internally-disjoint (i.e., with only vertex v in common) and $V(P_i) \cap B = \{b_i\}$ for $i = 1, 2, \ldots k$.

Other Characterizations

Another interesting characterization of k-connected graphs was independently conjectured by Frank and Maurer. The conjecture was proved by Lovász and by Györy (who worked independently), and it appears as Fact 33. Su proved a characterization of k-edge-connectivity for digraphs (Fact 34).

FACTS

F33: [Lo77,Gy78] A graph G with $n \geq k + 1$ vertices is k-connected if and only if, for any distinct vertices $u_1, u_2, \ldots, u_k$ and any positive integers $n_1, n_2, \ldots, n_k$ such that $n_1 + n_2 + \cdots + n_k = n$, there is a partition $V_1, V_2, \ldots, V_k$ of $V(G)$ such that $u_i \in V_i$, $|V_i| = n_i$, and the induced subgraph $G(V_i)$ is connected, $1 \leq i \leq n$.

F34: [Su97] A digraph G with at least k edges is k-edge-connected if and only if, for any k distinct arcs $e_i = (u_i, v_i)$, $1 \leq i \leq k$, the digraph $G - \{e_1, e_2, \ldots, e_k\}$ contains k edge-disjoint spanning arborescences (rooted trees) $T_1, T_2, \ldots, T_k$ such that T_i is rooted at v_i, $1 \leq i \leq n$.

4.1.3 Structural Connectivity

Here our purpose is to give results about certain configurations that must be present in a k-connected or k-edge-connected graph.

Cycles Containing Prescribed Vertices

The first is a classical result by Dirac, which generalizes Fact 15.

FACTS

F35: [Di60] Let G be a k-connected graph, $k \geq 2$. Then G contains a cycle through any given k vertices.

F36: [WaMe67] Let G be a k-connected graph with $k \geq 3$. Then G has a cycle containing a given set H with $k + 1$ vertices if and only if there is no set $T \subset V - H$ with $|T| = k$ vertices whose removal separates the vertices of H from each other.

Cycles Containing Prescribed Edges — The Lovász-Woodall Conjecture

Lovász [Lo74] and Woodall [Wo77] independently conjectured that every k-connected graph has a cycle containing a given set F of k *independent edges* (i.e., no two edges have a vertex in common), if and only if F is not an edge-disconnecting set of odd cardinality.

REMARK

R6: Lovász [Lo74,Lo77] first showed the *Lovász-Woodall Conjecture* is true for $k = 3$. The conjecture was also shown to be true for $k = 4$ [ErGy85,Lo90] and $k = 5$ [Sa96]. Subsequently, Häggkvist and Thomassen [HaTh82] arrived at the same conclusion by assuming that G is $(k+1)$-connected (without restriction on the the edge set F). More recently, the conjecture seems to have been settled by Kawarabayashi in a series of four papers (only one of which has been published to date).

FACT

F37: [Ka02a,Ka03a,Ka03b,Ka03c] Let G be a k-connected graph with $k \geq 2$, and let F be a set of k independent edges. Then G has a cycle containing F if and only if F is not an edge-disconnecting set of odd cardinality.

TERMINOLOGY: A subset of independent edges is also called a ***matching***. Matchings are discussed in Section 11.3.

Paths with Prescribed Initial and Final Vertices

Given any two subsets $A, B \subset V$ of k vertices of a k-connected graph, the existence of k disjoint paths P_i $(1 \leq i \leq k)$ connecting A and B is guaranteed by Menger's theorem. Menger's theorem does not, however, ensure that each of these paths can be so chosen to join a fixed u_i, v_i pair of vertices, $u_i \in A$, $v_i \in B$, $(1 \leq i \leq k)$. Now we consider the existence of paths with prescribed end-vertices.

DEFINITIONS

D26: A graph G is called ***k*-linked** if it has at least $2k$ vertices, and for every sequence $u_1, u_2, \ldots, u_k, v_1, v_2, \ldots, v_k$ of $2k$ different vertices, there exists a u_i–v_i path P_i, $i = 1, 2, \ldots, k$, such that the k paths are vertex-disjoint.

D27: A graph is ***weakly k-linked*** if it has at least $2k$ vertices, and for every k pairs of vertices (u_i, v_i), there exists a u_i–v_i path P_i, $1 \leq i \leq k$, such that the k paths are edge-disjoint.

D28: A graph is said to be ***k*-parity-linked** if one can find k disjoint paths with prescribed end-vertices and prescribed parities of the lengths.

D29: The ***bipartite index*** of a graph is the smallest number of vertices whose deletion creates a bipartite graph.

FACTS

F38: A k-linked graph is always $(2k-1)$-connected, but the converse is not true.

F39: [LaMa70], [Ju70] (independently) For each k, there exists an integer $f(k)$ such that if $\kappa \geq f(k)$ then G is k-linked.

F40: Thomassen [Th80a] and Seymour independently characterized the graphs that are not 2-linked. This is the first problem in the so-called ***k*-paths problem** that has been solved using the Robertson-Seymour theory [RoSe85].

NOTATION: For $k \geq 1$, $g(k)$ denotes the smallest integer such that every $g(k)$-edge-connected graph G is weakly k-linked.

CONJECTURE

[Th80a] For every integer $k \geq 1$, $g(2k+1) = g(2k) = 2k+1$.

FACTS

F41: [Ok84,Ok85,Ok87] If $k \geq 3$ is odd, $u_1, u_2, \ldots, u_k, v_1, v_2 \ldots, v_k$ are (not necessarily distinct) vertices from a set T with $|T| \leq 6$, and $\lambda(u_i, v_i) \geq k$ $(1 \leq i \leq k)$, then there exists a $u_i - v_i$ path for $1 \leq i \leq k$ such that the k paths are edge-disjoint.

F42: [Hu91] For every integer $k \geq 1$, $g(2k+1) \leq 2k+2$ and $g(2k) \leq 2k+2$.

F43: [Ok88,Ok90a] For every integer $k \geq 1$,

(a) $g(2k+1) \leq 3k$ and $g(2k+2) \leq 3k+2$,

(b) $g(3k) \leq 4k$ and $g(3k+2) \leq 4k+2$.

F44: [Th01] Every $f(k)$-connected graph (defined in Fact 39) with bipartite index at least $4k-3$ is k-parity-linked.

F45: [Su97] Let G be a k-edge-connected digraph. Then, for any k triples (u_1, f_1, v_1), (u_2, f_2, v_2), $\ldots$, (u_k, f_k, v_k), where $u_1, u_2 \ldots, u_k$, $v_1, v_2 \ldots, v_k$ are not necessarily distinct vertices, and $f_1, f_2, \ldots, f_k$ are different arcs of the form $f_i = (u_i, v_i)$, $i = 1, \ldots, k$, there exist k edge-disjoint u_i–v_i paths P_i in G such that $f_i \in E(P_i)$, $i = 1, \ldots, k$.

Subgraphs

High connectivity implies a large minimum degree (Fact 7). Conversely, a large minimum degree does not guarantee high connectivity (Fact 8). However, it does ensure the existence of a highly connected subgraph.

FACT

F46: [Ma72a] Every graph of minimum degree at least $4k$ contains a k-connected subgraph.

REMARK

R7: A very short proof of this result was given by Thomassen in [Th88]. In fact, Mader proved that if the average of the degrees of the vertices of G is at least $4k$, then G contains a k-connected subgraph.

4.1.4 Analysis and Synthesis

An interesting question in the study of graph connectivity is to describe how to obtain every k-(edge-)connected graph from a given "simple" one by a succession of elementary operations preserving k-connectedness. A classical result in this topic is Tutte's theorem, which states how to construct all 3-connected graphs, starting with a *wheel graph*. We also consider some relevant results dealing with deletion of edges or vertices. Finally, some facts concerning minimally and critically k-connected graphs, as well as a reference to connectivity augmentation problems, are considered.

Contractions and Splittings

DEFINITIONS

D30: The ***contraction*** of an edge uv consists of the identification of its endpoints u and v (keeping the old adjacencies but removing the self-loop from $u = v$ to itself). Let G be a k-connected graph. An edge of G is said to be ***k-contractible*** if its contraction results in a k-connected graph.

D31: The converse operation is called ***splitting***: A vertex w with degree δ is replaced by an edge uv in such a way that some of the vertices adjacent to w are now adjacent to u and the rest are adjacent to v. Moreover, if the new vertices u, v have degrees at least $k = \delta/2 + 1$ we speak about a ***k-vertex-splitting***.

D32: For any integer $n \geq 4$, the ***wheel graph*** W_n is the n-vertex graph obtained by joining a vertex to each of the $n - 1$ vertices of the cycle graph C_{n-1}.

FACTS

F47: If G is a k-connected graph, the operations of k-vertex splitting and edge addition always produce a graph that is also (at least) k-connected. In fact, as shown below, for $k = 3$ these operations suffice to derive all 3-connected graphs.

F48: [Th80b] Every 3-connected graph distinct from K_4 has a 3-contractible edge.

F49: [Th81] Every *triangle-free* (no 3-cycles) k-connected graph has a k-contractible edge.

F50: [Tu61] Every 3-connected graph can be obtained from a wheel by a finite sequence of 3-vertex-splittings and edge additions.

EXAMPLE

E2: In Figure 4.1.2, the cube graph Q_3 is synthesized from the wheel graph W_5 in four steps. All but the second step are 3-vertex-splittings.

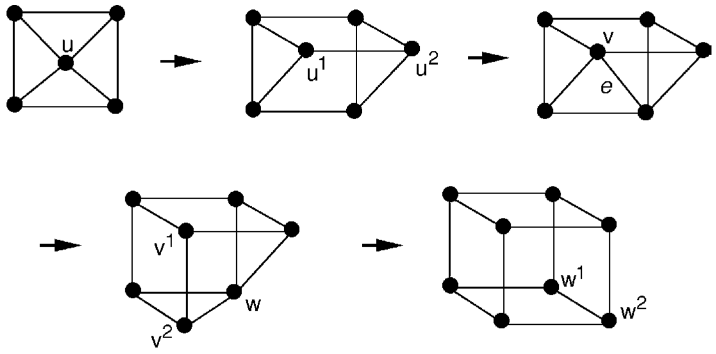


Figure 4.1.2 **A 4-step Tutte synthesis of the cube graph Q_3.**

REMARKS

R8: In general, k-connectedness does not assure the existence of k-contractible edges.

R9: Thomassen used Fact 48 to give a short proof of Kuratowski's theorem on planarity. Fact 48 can also be derived from Tutte's theorem (Fact 50).

R10: Since Tutte's paper, the distribution of contractible edges in graphs of given connectivity has been extensively studied. For a comprehensive survey of this subject, we refer the reader to [Kr02], where the author also considers subgraph contractions (see below).

R11: Fact 50 is a reformulation of the following proposition [Tu61]: a 3-connected graph is either a wheel, or it contains an edge whose removal leaves a 3-connected subgraph, or it contains a 3-contractible edge that is not in a cycle of length 3.

R12: Slater [Sl74] gave a similar result for constructing all 4-connected graphs starting from K_5, but in this case three more operations are required. For $k \geq 5$ the problem is still open. However, Lovász [Lo74] and Mader [Ma78a] managed to construct all k-edge-connected pseudographs (loops and multiple edges allowed) for every k even and odd, respectively.

Subgraph Contraction

The contraction of a subgraph is a natural generalization of edge contraction.

DEFINITION

D33: A connected subgraph H of a k-connected graph G is said to be ***k*-contractible** if the contraction of H into a single vertex results in a k-connected graph.

FACTS

F51: [McOt94] Every 3-connected graph on $n \geq 9$ vertices has a 3-contractible path of length two.

F52: [ThTo81] Every 3-connected graph with minimum degree at least 4 contains a 3-contractible cycle.

F53: [Kr00] Every 3-connected graph of order at least eight has a 3-contractible subgraph of order four.

CONJECTURE

[McOt94] For every n, a 3-connected graph of sufficiently large order has a 3-contractible subgraph of order n.

Edge Deletion

DEFINITION

D34: A subgraph H of a k-edge-connected graph G is said to be ***ρ*-reducible** if the graph obtained from G by removing the edges of H is $(k - \rho)$-connected.

FACTS

F54: [Ma74] Every k-connected graph G with minimum degree at least $k+2$ contains a cycle C such that $G - E(C)$ is k-connected.

F55: [Ok88] Let G be a k-edge-connected graph with $k \geq 4$ even. Let $\{u, v\} \subset V$ and $\{e_1, e_2, f\} \subset E$, $e_i \neq f$ $(i = 1, 2)$. Then,

(a) There exists a 2-reducible cycle containing e_1 and e_2, but not f.

(b) There exists a 2-reducible u–v path containing e_1, but not f.

F56: [Ok90b] Let G be a k-edge-connected graph with $k \geq 2$ even. If $\{u_1, v_1, u_2, v_2\}$ are distinct vertices, with edges $e_0 = v_1v_2$, $e_i = u_iv_i$ $(i = 1, 2)$, and there is no edge-cut with k or $k + 1$ elements containing $\{e_0, e_1, e_2\}$, then there exists a 2-reducible cycle containing $\{e_0, e_1, e_2\}$.

F57: [HuOk92] For each odd $k \geq 3$, there exists a k-edge-connected graph containing two vertices u and v such that every cycle passing through u, v is ρ-reducible with $\rho \geq 3$.

REMARK

R13: For the case of three consecutive edges e_1, e_2, e_3 of a k-connected graph, Okamura [Ok95] also found a nontrivial equivalent reformulation of the condition that no cycle of G containing e_1, e_2, and e_3 is 2-reducible.

Vertex Deletion

FACTS

F58: [ChKaLi72] Every 3-connected graph of minimum degree at least 4 has a vertex v such that $G - v$ is 3-connected.

F59: [Th81] Every $(k + 3)$-connected graph has an induced (chordless) cycle whose deletion results in a k-connected graph.

F60: [Eg87] Every $(k + 2)$-connected triangle-free graph has an induced cycle whose deletion results in a k-connected graph.

REMARK

R14: Fact 59 was conjectured by Lovász, and Thomassen used Fact 49 to prove it.

Minimality and Criticality

A standard technique used to study a certain property $\mathcal{P}$ is to consider those graphs that are edge-minimal or vertex-minimal (critical) with respect to $\mathcal{P}$, in the sense that the removal of any vertex or edge produces a graph for which $\mathcal{P}$ does not hold.

DEFINITIONS

D35: A graph or digraph G is said to be ***minimally k-connected*** if $\kappa(G) \geq k$ but, for each edge $e \in E$, $\kappa(G - e) < k$. Analogously, G is ***minimally k-edge-connected*** if $\lambda(G) \geq k$, but for each $e \in E$, $\lambda(G - e) < k$.

D36: A vertex u of a digraph has ***half degree*** k if either $deg^+(u) = k$ or $deg^-(u) = k$.

FACTS

F61: [Ma71,Ma72b] Every minimally k-connected (or k-edge-connected) graph contains at least $k + 1$ vertices of degree k.

F62: [Ma72b] Every cycle of a minimally k-connected graph contains a vertex of degree k.

F63: Every cycle in a k-connected graph G contains either a vertex of degree k or an edge whose removal does not lower the connectivity of G.

F64: [Ha81] Every minimally k-connected digraph contains at least $k+1$ vertices of half degree k.

REMARKS

R15: Halin [Ha69,Ha00] proved the existence of a vertex of degree k in every minimally k-connected graph, and the corresponding theorem for minimally k-edge-connected graphs was proved by Lick [Li72]. Both results were then improved by Mader (Fact 61).

R16: Fact 64, proved by Hamidoune, is the digraph analogue of (and implies) Mader's theorem (Fact 61) about the existence of vertices of degree k. The existence of at least one of vertex of half degree k had been previously asserted by Kameda [Ka74].

Vertex-Minimal Connectivity – Criticality

Maurer and Slater [MaSl77] introduced the general concept of *critically connected* and *critically edge-connected graphs*, i.e., graphs whose connectivity decreases when one or more vertices are removed.

DEFINITION

D37: A graph G is called ***k-critically n-connected***, or an (n,k)***-graph***, if, for each vertex subset U with $|U| \leq k$, we have $\kappa(G-U) = n - |U|$. When $k=1$, we simply refer to the graph as ***critically n-connected***.

FACTS

F65: [MaSl77] The only (n,n)-graph is the complete graph K_{n+1}.

F66: The "cocktail party graph" (obtained from K_{2n+2} by removing a 1-factor [perfect matching]) is a $(2n,n)$-graph but not a $(2n,n+1)$-graph.

F67: [Su88] The complete graph on $k+1$ vertices is the unique k-critically n-connected graph with $n < 2k$.

F68: [Ma77] If G is a $(n,3)$-graph, then its order is at most $6n^2$. Thus, for each n, there are only finitely many of $(n,3)$-critical graphs.

REMARKS

R17: A survey about (n,k)-graphs, along with some conjectures and open problems, can be found in [Ma84].

R18: Fact 66 led Slater to conjecture that, apart from K_{n+1}, there is no (n,k)-graph with $k > n/2$, which, after some partial results, was finally proved by Su (Fact 67).

R19: Fact 68 was generalized by Mader to the class of all finite n-connected graphs.

Connectivity Augmentation

We conclude the section by referring the reader to [Fr94] for an in-depth discussion of *connectivity augmentation.* In the *edge-connectivity augmentation problem*, we are given a graph $G = (V, E)$ and a positive integer k, and the goal is to find the smallest set of edges F that we can add to G such that $G' = (V, E \cup F)$ is k-connected. Due to its applicability to the design of fault-tolerant networks, connectivity augmentation has also been widely investigated from an algorithmic point of view. Watanabe and Nakamura [WaNa87] gave the first polynomial-time algorithm solving the edge-connectivity augmentation problem. In the same paper, the authors formulated a necessary and sufficient condition to decide if a given graph G can be made k-connected by adding at most a certain number of edges. The same question for digraphs was solved in [Fr92].

References

[AhDi94] R. Aharoni and R. Diestel, Menger's theorem for a countable source set, *Combin. Probab. Comput.* 3 (1994), 145–156.

[BaGu01] J. Bang-Jensen and G. Gutin, *Digraphs. Theory, Algorithms and Applications*, Springer-Verlag, London, 2001.

[Be76] C. Berge, *Graphs and Hypergraphs*, Second Edition, North-Holland Pub. Co., New York, 1976.

[BeHoPe89] J.-C. Bermond, N. Homobono, and C. Peyrat, Large fault-tolerant interconnection networks, *Graphs Combin.* 5 (1989), 107–123.

[ChHa68] G. Chartrand and F. Harary, Graphs with prescribed connectivities, pp. 61–63 in *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, Academic Press, New York, 1968.

[ChKaLi72] G. Chartrand, A. Kaugars, and D. R. Lick, Critically n-connected graphs, *Proc. Amer. Math. Soc.* 32 (1972), 63–68.

[ChLe96] G. Chartrand and L. Lesniak, *Graphs & Digraphs*, Third Edition, Chapman and Hall, London/New York, 1996.

[Di00] R. Diestel, *Graph Theory*, Second Edition, Graduate Texts in Mathematics, Volume 173, Springer-Verlag, New York, 2000.

[Di60] G. A. Dirac, In abstrakten Graphen vorhandene vollständige 4-Graphen und ihre Unterteilungen, *Math. Nachr.* 22 (1960), 61–85.

[ElFeSh56] P. Elias, A. Feinstein, and C. E. Shannon, A note on the maximum flow through a network, *IRE Trans. Inform. Theory* IT–2 (1956), 117–119.

[Eg87] Y. Egawa, Cycles in k-connected graphs whose deletion results in a $(k-2)$-connected graph, *J. Combin. Theory Ser. B* 42 (1987), 371–377.

[EnJaSl77] R. Entringer, D. Jackson, and P. Slater, Geodetic connectivity of graphs, *IEEE Trans. Circuits and Systems* 24 (1977), 460–463.

[ErGy85] P. L. Erdös and E. Györi, Any four independent edges of a 4-connected graph are contained in a circuit, *Acta Math. Hung.* 46 (1985), 311–313.

[FoFu56] L. R. Ford and D. R. Fulkerson, Maximal flow through a network, *Canad. J. Math.* 8 (1956), 399–404.

[Fr90] A. Frank, Packing paths, circuits, and cuts – a survey, pp. 47–100 in B. Korte, L. Lovász, H-J. Prömel, and A. Schrijver (Eds.), *Paths, Flows and VLSI-Layouts*, Springer, Berlin, 1990.

[Fr92] A. Frank, Augmenting graphs to meet edge-connectivity requirements, *SIAM J. Discrete Math.* 5 (1992), 22–53.

[Fr94] A. Frank, Connectivity augmentation problems in network design, pp. 34–63 in J.R. Birge and K.G. Murty (Eds.), *Mathematical Programming: State of the Art 1994*, The University of Michigan, Ann Arbor, 1994.

[Fr95] A. Frank, Connectivity and network flows, pp. 111–177 in R. Graham, M. Grötschel and L. Lovász (Eds.), *Handbook of Combinatorics*, Elsevier Science B.V., 1995.

[Ga61] T. Gallai, Maximum-Minimum Sätze und verallgemeinerte Faktoren von Graphen, *Acta Math. Sci. Hungar.* 12 (1961), 131–173.

[GeHa70] D. Geller and F. Harary, Connectivity in digraphs, *Lec. Not. Math.* 186, Springer, Berlin (1970), 105–114.

[Gi85] A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, Cambridge, 1996.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, Boca Raton, 1999.

[Gy78] E. Györi, On division of graphs to connected subgraphs, pp. 485–494 in *Combinatorics*, North-Holland, Amsterdam, 1978.

[Ha00] R. Halin, Miscellaneous problems on infinite graphs, *J. Graph Theory* 35 (2000), 128–151.

[Ha69] R. Halin, A theorem on n-connected graphs, *J. Combin. Theory* 7 (1969), 150–154.

[Ha81] Y. O. Hamidoune, Quelques problèmes de connexité dans les graphes orientés, *J. Combin. Theory Ser. B* 30 (1981), 1–10.

[HaNoCa68] F. Harary, R. Z. Norman, and D. Cartwright, *Introduction à la Théorie des Graphes Orientés*, Dunod, Paris, 1968.

[HaTh82] R. Häggkvist and C. Thomassen, Circuits through specified edges, *Discrete Math.* 41 (1982), 29–34.

[HeRaGa00] M. R. Henzinger, S. Rao, and H. Gabow, Computing vertex connectivity: new bounds from old techniques, *J. Algorithms* 34 (2000), 222–250.

[Hu91] A. Huck, A sufficient condition for graphs to be weakly k-linked, *Graphs and Combin.* 7 (1991), 323–351.

[HuOk92] A. Huck and H. Okamura, Counterexamples to a conjecture of Mader about cycles through specified vertices in n-edge-connected graphs, *Graphs Combin.* 8 (1992), 253–258.

[Ju70] H. A. Jung, Eine Verallgemeinerung des n-fachen Zusammenhangs für Graphen, *Math. Ann.* 187 (1970), 95–103.

[Ka02a] K. Kawarabayashi, One or two disjoint cycles cover independent edges: Lovász-Woodall Conjecture, *J. Combin. Theory Ser. B* 84 (2002), 1–44.

[Ka03a] K. Kawarabayashi, Two circuits through independent edges: Lovász-Woodall Conjecture, submitted.

[Ka03b] K. Kawarabayashi, Extremal problems for two circuits through independent edges: Lovász-Woodall Conjecture, in preparation.

[Ka03c] K. Kawarabayashi, Proof of Lovász-Woodall Conjecture, in preparation.

[Ka74] T. Kameda, Note on Halin's theorem on minimally connected graphs, *J. Combin. Theory Ser. B* 17 (1974), 1–4.

[Kr00] M. Kriesell, Contractible subgraphs in 3-connected graphs, *J. Combin. Theory Ser. B* 80 (2000), 32–48.

[Kr02] M. Kriesell, A survey on contractible edges in graphs of a prescribed vertex connectivity, *Graphs Combin.* 18 (2002), 1–30.

[LaMa70] D. G. Larman and P. Mani, On the existence of certain configurations within graphs and the 1-skeletons of polytopes, *Proc. London Math. Soc* 20 (1970), 144–160.

[Li72] D. R. Lick, Minimally n-line connected graphs, *J. Reine Angew. Math.* 252 (1972), 178–182.

[Lo74] L. Lovász, Problem 5, *Period. Math. Hung.* 4 (1974), 82.

[Lo76] L. Lovász, On some connectivity properties of eulerian graphs, *Acta Math. Acad. Sci. Hung.* 28 (1976), 129–138.

[Lo77] L. Lovász, A homology theory for spanning trees of a graph, *Acta Math. Acad. Sci. Hung.* 30 (1977), 241–251.

[Lo90] M. V. Lomonosov, Cycles through prescribed elements in a graph, *Algorithms Comb.* 9 (1990), 215–234.

[Lo93] L. Lovász, *Combinatorial Problems and Exercises*, 2nd Edition, North-Holland, Amsterdam, 1993.

[LoNePl78] L. Lovász, V. Neumann-Lara, and M. D. Plummer, Mengerian theorems for paths of bounded length, *Period. Math. Hungar.* 9 (1978), 269–276.

[Ma71] W. Mader, Minimale n-fach kantenzusammenhängende Graphen, *Math. Ann. (Basel)* 191 (1971), 21–28.

[Ma72a] W. Mader, Existenz n-fach zusammenhängender Teilgraphen in Graphen genügend grosser Kantendichte, *Abh. Math. Sem. Univ. Hamburg* 37 (1972), 86–97.

[Ma72b] W. Mader, Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen, *Arch. Math. (Basel)* 23 (1972), 219–224.

[Ma73] W. Mader, Grad und lokaler Zusammenhang in endlichen Graphen, *Math. Ann.* 205 (1973), 9–11.

[Ma74] W. Mader, Kreuzungsfreie a, b-Wege in endlichen Graphen, *Abh. Math. Sem. Univ. Hamburg* 42 (1974), 187–204.

[Ma77] W. Mader, Endlichkeitssätze für k-kritische Graphen, *Math. Ann.* 229 (1977), 143–153.

[Ma78a] W. Mader, A reduction method for edge-connectivity in graphs, *Ann. Discrete Math.* 3 (1978), 145–164.

[Ma78b] W. Mader, Über die Maximalzahl kantendisjunkter A-Wege, *Arch. Math. (Basel)* 30 (1978), 325–336.

[Ma78c] W. Mader, Über die Maximalzahl kreuzungsfreier H-Wege, *Arch. Math. (Basel)* 31 (1978/79), 387–402.

[Ma79] W. Mader, Connectivity and edge-connectivity in finite graphs, pp. 66–95 in *Surveys in Combinatorics, Proc. 7th Br. Comb. Conf.*, Cambridge 1979, Lond. Math. Soc. Lect. Note Ser. 38, 1979.

[Ma84] W. Mader, On k-critically n-connected graphs, pp. 389–398 in *Progress in graph theory (Waterloo, Ont., 1982)*, Academic Press, Toronto, ON, 1984.

[MaSl77] S. B. Maurer and P. J. Slater, On k-critical, n-connected graphs, *Discrete Math.* 20 (1977), 255–262.

[McCu84] W. D. McCuaig, A simple proof of Menger's Theorem, *J. Graph Theory* 8 (1984), 427–429.

[McOt94] W. D. McCuaig and K. Ota, Contractible triples in 3-connected graphs, *J. Combin. Theory Ser. B* 60 (1994), 308–314.

[Me27] K. Menger, Zur allgeminen Kurventheorie, *Fund. Math.* 10 (1927), 96–115.

[Oe03] O. R. Oellermann, Variations on Menger's Theorem, in L. W. Beineke and R. J. Wilson (Eds.), *Structural Graph Theory*, to appear.

[Oe96] O. R. Oellermann, Connectivity and edge-connectivity in graphs: A survey, *Congr. Numerantium* 116 (1996), 231–252.

[Ok84] H. Okamura, Paths and edge-connectivity in graphs, *J. Combin. Theory B* 37 (1984), 151–172.

[Ok85] H. Okamura, Paths and edge-connectivity in graphs, II, pp. 337–352 in J. Akiyama et al. (Eds.), *Number Theory and Combinatorics*, World Scientific Publishing, 1985.

[Ok87] H. Okamura, Paths and edge-connectivity in graphs, III: Six-terminal k paths, *Graphs Combin.* 3 (1987), 159–189.

[Ok88] H. Okamura, Paths in k-edge-connected graphs, *J. Combin. Theory Ser. B* 45 (1988), 345–355.

[Ok90a] H. Okamura, Every $4k$-edge-connected graph is weakly $3k$-linked, *Graphs Combin.* 6 (1990), 179–185.

[Ok90b] H. Okamura, Cycles containing three consecutive edges in $2k$-edge-connected graphs, pp. 549–553 in R. Bodendiek and R. Henn (Eds.), *Topics in Combinatorics and Graph Theory*, Physica-Verlag, Heidelberg, 1990.

[Ok95] H. Okamura, 2-reducible cycles containing three consecutive edges in $(2k+1)$-edge-connected graphs, *Graphs Combin.* 11 (1995), 141–170.

[RoSe85] N. Robertson and P. D. Seymour, Graph minors—a survey, pp. 153–171 in *Surveys in combinatorics 1985* (Glasgow, 1985), London Math. Soc. Lecture Note Ser. 103, Cambridge Univ. Press, Cambridge, 1985.

[Sa96] D. P. Sanders, On circuits through five edges, *Discrete Math.* 159 (1996), 199–215.

[Sl74] P. J. Slater, A classification of 4-connected graphs, *J. Combin. Theory* 17 (1974), 281–298.

[Su88] J. J. Su, Proof of Slater's conjecture on k-critical n-connected graphs, *Kexue Tongbao (English ed.)* 33 (1988), 1675–1678.

[Su97] X.-Y. Su, Some generalizations of Menger's theorem concerning arc-connected digraphs, *Discrete Math.* 175 (1997), 293–296.

[Th01] C. Thomassen, The Erdös-Pósa property for odd cycles in graphs of large connectivity, *Combinatorica* 21 (2001), 321–333.

[Th80a] C. Thomassen, 2-linked graphs, *European J. Combin.* 1 (1980), 371–378.

[Th80b] C. Thomassen, Planarity and duality of finite and infinite graphs, *J. Combin. Theory Ser. B* 29 (1980), 244–271.

[Th81] C. Thomassen, Non-separating cycles in k-connected graphs, *J. Graph Theory* 5 (1981), 351–354.

[Th88] C. Thomassen, Paths, circuits and subdivisions, pp. 97–131 in L. W. Beineke and R. J. Wilson (Eds.), *Selected Topics in Graph Theory III*, Academic Press, London, 1988.

[ThTo81] C. Thomassen and B. Toft, Non-separating induced cycles in graphs, *J. Combin. Theory B* 31 (1981), 199–224.

[Tu61] W. T. Tutte, A theory of 3-connected graphs, *Nederl. Akad. Wetench. Indag. Math.* 23 (1961), 441–455.

[Tu66] W. T. Tutte, *Connectivity in Graphs*, University of Toronto Press, London, 1966.

[WaMe67] M. E. Watkins and D. M. Mesner, Cycles and connectivity in graphs, *Canad. J. Math.* 19 (1967), 1319–1328.

[WaNa87] T. Watanabe and A. Nakamura, Edge-Connectivity Augmentation Problems, *J. Comput. System Sci.* 35 (1987), 96–144.

[Wh32] H. Whitney, Congruent graphs and the connectivity of graphs, *Amer. J. Math.* 54 (1932), 150–168.

[Wi92] A. Wigderson, The complexity of graph connectivity, pp. 112–113 in I. M. Havel and V. Koubek (Ed.), *Lecture Notes in Comput. Sci.*, 629, Proc. Mathematical Foundations of Computer Science 1992, Springer, Berlin, 1992.

[Wo77] D. R. Woodall, Circuits containing specified edges, *J. Combin. Theory Ser. B* 22 (1977), 274–278.

4.2 EULERIAN GRAPHS

Herbert Fleischner, Technical University of Vienna, Austria

4.2.1 Basic Definitions and Characterizations
4.2.2 Algorithms to Construct Eulerian Tours
4.2.3 Eulerian-Tour Enumeration and Other Counting Problems
4.2.4 Applications to General Graphs
4.2.5 Various Types of Eulerian Tours and Cycle Decompositions
4.2.6 Transforming Eulerian Tours
References

Introduction

Eulerian graph theory has its roots in the Königsberg Bridges Problem: Four landmasses are being connected by seven bridges as depicted in Figure 4.2.1(a).

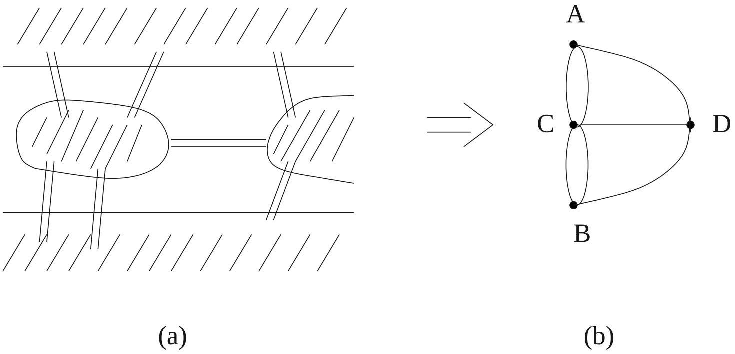


Figure 4.2.1

QUESTION: Starting at any of the four landmasses, is it possible to perform a walk such that every bridge is crossed once and only once, the walk ending at any of these four landmasses? L. Euler wrote an article on this problem in 1736 [Eu,1736]; hence the name eulerian graph. This paper can be viewed as the "birth certificate" for graph theory, in general. For an extensive treatment of eulerian graphs and related topics see [Fl90, Fl91].

4.2.1 Basic Definitions and Characterizations

NOTATION: Throughout this section, a graph, digraph, or mixed graph are denoted $G = (V, E)$, where V is the vertex-set of G and E is the edge-set of G, consisting of undirected edges, directed edges (arcs), or both, respectively.

TERMINOLOGY: Sometimes, for emphasis and to avoid confusion, the adjective "undirected" is used for "graph" or "edge".

DEFINITIONS

D1: An ***eulerian tour*** in a graph (or digraph) G is a closed walk that uses each edge (or arc) of G *exactly* once. An eulerian tour in a mixed graph is a closed trail that uses each edge and each arc exactly once.

D2: A graph, digraph, or mixed graph that has an eulerian tour is called ***eulerian***.

D3: An undirected graph is ***even*** if every vertex has even degree.

D4: In a digraph, a vertex v is ***balanced*** if the indegree and outdegree of v are equal. A digraph is balanced if all of its vertices are balanced.

TERMINOLOGY NOTE: In §4.3, the term *symmetric* is used instead of balanced when the indegree and outdegree of v are equal.

D5: A ***balanced orientation*** of a graph (or mixed graph) G is an assignment of a direction to each edge of the graph (or each undirected edge of the mixed graph) so that the resulting digraph is balanced.

D6: A ***cycle decomposition*** of a graph (digraph) G is a partition of the edge-set (arc-set) of G such that each partition set forms a cycle (directed cycle).

Some Basic Characterizations

FACTS

For details of the following facts, see, e.g., [To73, Mc84, Wo90, Fl89, Fl90].

F1: *The Classical Characterization* ([Eu1736], [Hi1873], [Ve12, Ve31])

Let G be a connected graph. The following are equivalent:

(a) G is eulerian.

(b) G is an even graph.

(c) G has a cycle decomposition.

F2: A graph is even if and only if it has a balanced orientation.

F3: A graph is even if and only if it has a decomposition into closed trails.

F4: A graph is even if and only if every edge belongs to an odd number of cycles.

F5: A graph is even if and only if it has an odd number of cycle decompositions.

F6: A connected graph $G = (V, E)$ is eulerian if and only if the number of subsets of E (including the empty set) that induce an acyclic subgraph of G is odd ([Sh79, Fl89, Fl90]).

F7: For a connected digraph D the following are equivalent.

(a) D is eulerian.

(b) D is a balanced digraph.

(c) D has a directed-cycle decomposition.

REMARKS

R1: For the classical characterization, Euler ([Eu,1736]) showed that Fact 1(a) implies Fact 1(b), while the converse is due to Hierholzer [Hi,1873]. The equivalence of Fact 1(b) and Fact 1(c) is due to Veblen ([Ve12, Ve31]).

R2: By Fact 1, the statements in Facts 2 through 6 can be viewed as alternative characterizations of eulerian graphs.

R3: Note that a connected eulerian digraph is strongly connected.

R4: There is no digraph or mixed graph analog for the characterization expressed in Fact 4.

Characterizations Based on Partition Cuts

DEFINITIONS

D7: Let G be a graph and let $X \subset V(G)$. The ***partition-cut*** associated with X, denoted $E(X, \overline{X})$, is the set of edges in G with one endpoint in X and one endpoint in $\overline{X} = V(G) - X$. A partition-cut in a digraph or mixed graph is analogously defined.

D8: An ***edge-cut***, ***arc-cut***, and ***mixed-cut*** are partition-cuts in a graph, digraph, and mixed graph G, respectively, associated with some $X \subset V(G)$.

D9: The ***out-arcs*** of an arc-cut (or mixed-cut) $E(X, \overline{X})$ is the subset of directed edges whose tail is in X and is denoted $E^+(X, \overline{X})$. The ***in-arcs*** of $E(X, \overline{X})$ is the subset of directed edges whose head is in X and is denoted $E^-(X, \overline{X})$.

D10: Let v be a vertex of a graph, digraph, or mixed graph G. The ***incidence set*** of v, denoted E_v, is the partition-cut $E(X, \overline{X})$, where $X = \{v\}$.

NOTATION: In a digraph, the out-arcs and in-arcs of the incidence set of v are denoted E_v^+ and E_v^-, respectively.

FACTS

F8: A graph G is even if and only if $|E(X, \overline{X})|$ is even for every $X \subset V(G)$.

F9: A connected digraph G is eulerian if and only if $|E^+(X, \overline{X})| = |E^-(X, \overline{X})|$ for every $X \subset V(G)$.

F10: Let G be a connected mixed graph. The following are equivalent:

(a) G is eulerian.

(b) $|E(X, \overline{X})| - \big||E^+(X, \overline{X})| - |E^-(X, \overline{X})|\big|$ is nonnegative and even for every $X \subset V(G)$.

(c) G has a cycle decomposition.

REMARKS

R5: While Fact 1(b) and its digraph analogue, Fact 7(b), are (*local*) degree conditions guaranteeing that a graph (undirected or directed) is eulerian, for mixed graphs one needs the *global* condition in Fact 10(b) (which reduces to Facts 8 and 9 for undirected graphs and digraphs).

R6: Although the condition in Fact 10 is impractical from an algorithmic point of view for producing an eulerian tour in a mixed graph G, such a tour can be obtained using *network-flow techniques* by first getting a balanced orientation D_G of G; then any eulerian tour of D_G corresponds to an eulerian tour in G [FoFu62].

4.2.2 Algorithms to Construct Eulerian Tours

We begin with two classical algorithms for constructing an eulerian tour. All three algorithms in this subsection are polynomial-time (see [Fl90]).

Algorithm 4.2.1: Hierholzer's Algorithm [Hi,1873]

Input: a connected graph G whose vertices all have even degree.
Output: an eulerian tour T.

```
Start at any vertex $v$, and construct a closed trail $T$ in $G$.
While there are edges of $G$ not already in trail $T$
    Choose any vertex $w$ in $T$ that is incident on an unused edge.
    Starting at vertex $w$, construct a closed trail $D$ of unused edges.
    Enlarge trail $T$ by splicing trail $D$ into $T$ at vertex $w$.
Return $T$.
```

COMPUTATIONAL NOTE: A modified *depth-first search* (see §10.1), in which every unused edge remains in the stack, can be used to construct the closed trails.

EXAMPLE

E1: The key step in Algorithm 4.2.1 is enlarging a closed trail by combining it with a second closed trail — the "detour." To illustrate, consider the closed trails, $T = \langle t_1, t_2, t_3, t_4 \rangle$ and $D = \langle d_1, d_2, d_3 \rangle$, in the graph shown in Figure 4.2.2. The closed trail that results when detour D is spliced into trail T at vertex w is given by $T' = \langle t_1, t_2, d_1, d_2, d_3, t_3, t_4 \rangle$. At the next iteration, the trail $\langle e_1, e_2, e_3 \rangle$ is spliced into trail T', resulting in an eulerian tour of the entire graph.

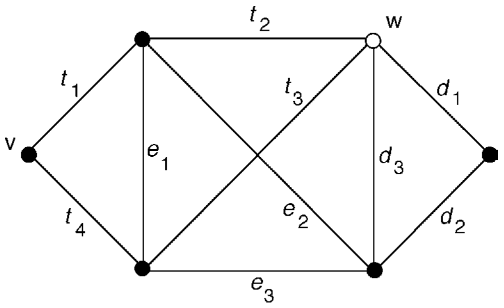


Figure 4.2.2

REMARKS

R7: The splicing operation in Hierholzer's algorithm is also called a *κ-absorption* and is discussed later in this section.

R8: The strategy in Fleury's algorithm, shown below, is to avoid, if possible, traversing a bridge in the subgraph induced on the set of untraversed edges. Fleury's algorithm also appeared in [Lu,1894].

Algorithm 4.2.2: Fleury's Algorithm [Fl,1883]

Input: Eulerian graph G with q edges and $v_0 \in V(G)$.
Output: Eulerian tour T_q.

```
Choose $e_1 = v_0 v_1 \in E_{v_0}$ arbitrarily.
Let $T_1 = \langle v_0, e_1, v_1 \rangle$
For $i = 1$ to $q - 1$
    Let $G_i = G - E(T_i)$.
    If $\deg_{G_i}(v_i) = 1$
        Let $e_{i+1} = v_i v_{i+1} \in E(G_i)$.
    Else
        Choose $e_{i+1} = v_i v_{i+1} \in E(G_i)$ that is not a bridge in $G_i$.
    Extend $T_i$ to $T_{i+1} = \langle v_0, e_1, v_1, \ldots, v_i, e_{i+1}, v_{i+1} \rangle$.
```

The Splitting and Detachment Operations

The *splitting* and *detachment* operations can serve as the basis for many of the characterizations, constructions, and decompositions discussed in this section.

DEFINITIONS

D11: Let G be a graph with vertex v such that $\deg(v) \geq 3$, and let e_a, e_b be incident on v and w_a, w_b, respectively. The graph $G_{a,b}$ obtained from G by introducing a new vertex $v_{a,b}$, adding new edges e'_a, e'_b joining $v_{a,b}$ and w_a, w_b, respectively, and deleting e_a, e_b is called the ***a-b split of*** G ***at*** v. The operation that produces $G_{a,b}$ is called the ***splitting operation*** (see Figure 4.2.3).

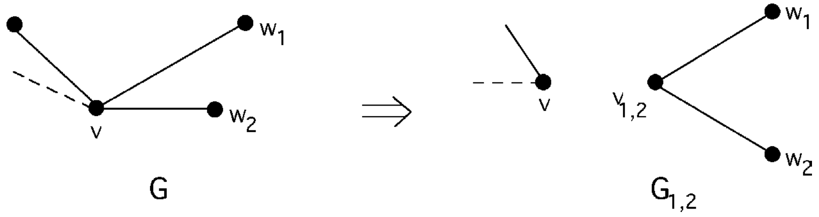


Figure 4.2.3 **The splitting operation producing the 1-2 split of G at v.**

D12: Let v be a vertex of a graph G with $\deg(v) \geq 2$, and let the edge subsets $E_1(v), E_2(v), \ldots, E_k(v)$, $k \geq 2$ be a partition of the incidence set E_v. Replace v with new vertices $v_1, v_2, \ldots, v_k$, and let v_i, $i = 1, 2, \ldots, k$, be incident on the edges of $E_i(v)$ (without altering any other incidence). The graph H thus obtained is called a ***detachment of*** G ***at*** v. This action is called a ***detachment operation at*** v (see Figure 4.2.4).

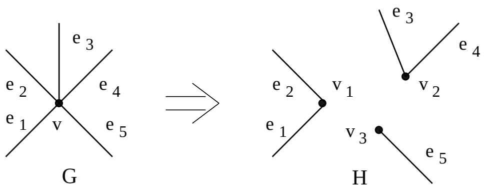


Figure 4.2.4 **Graph H is a detachment of G at v.**

D13: A graph H is a ***detachment of*** G if it results from a sequence of detachment operations performed at each of the vertices of some vertex subset $W \subseteq V(G)$. For a discussion of detachments of graphs, see [Na79, Na85a, Na85b].

FACTS

F11: *Splitting Lemma.* Let v be a vertex of a connected, bridgeless graph G with $\deg(v) \geq 4$, and let $e_1, e_2, e_3 \in E_v$.
(a) If v is not a cut-vertex then at least one of the splits $G_{1,2}$ or $G_{1,3}$ is connected and bridgeless.
(b) If v is a cut-vertex and e_1 and e_3 belong to different blocks, then $G_{1,3}$ is connected and bridgeless.

F12: Let v be a vertex in a 2-connected graph G (i.e., no cut-vertices) with $\deg(v) \geq 4$. If neither the 1-2 split $G_{1,2}$ at v nor the 1-3 split $G_{1,3}$ is 2-connected, then $G_{1,2}$ and $G_{1,3}$ have the same cut-vertex x and no other cut-vertices. In this case, both $G_{1,2}$ and $G_{1,3}$ are connected and bridgeless.

F13: A graph is connected if and only if there is a detachment of G that is a tree.

F14: A graph is eulerian if and only if it has a detachment that is a cycle.

REMARKS

R9: The Splitting Lemma (Fact 11) can serve as the basis for many of the results and algorithms mentioned in this section (see, e.g., [Fl90]). It can also be used to restrict, with no loss in generality, various other (solved as well as unsolved) graph theoretical problems to 3-regular graphs. For a short proof of the Splitting Lemma, see [Fl00].

R10: Definitions 11 through 13 and Facts 11 through 14 can be formulated for graphs with self-loops as well. In this case (but also later on) it makes sense to consider an edge e (not just self-loops) as composed of two *half-edges* incident on the respective endpoints of e. Correspondingly, one then considers the splitting operation as involving different half-edges and the sets E_v, $E_i(v)$ as being sets of half-edges.

R11: The splitting operation can be viewed as a special case of the detachment operation, where the partition of the incidence set E_v has exactly two cells with at least one cell containing exactly two edges.

Algorithm 4.2.3: Splitting Algorithm

Input: Eulerian graph G with q edges and $v_0 \in V(G)$.
Output: Eulerian tour T_q in the form of a detachment of G.

```
Initialize $H = G$.
Choose $e_1 = v_0 v_1 \in E_{v_0}$ arbitrarily.
Let $T_1 = \langle v_0, e_1, v_1 \rangle$
For $i = 1$ to $q$
    If $\deg_H(v_i) = 2$
        Let $e_{i+1} = v_i v_{i+1} \in E_{v_i}(H) - E(T_i)$.
    Else {apply splitting lemma}
        If $v_i$ is not a cut-vertex of $H$
            Choose $e_{i+1} = v_i v_{i+1} \in E_{v_i}(H) - E(T_i)$ arbitrarily.
        Else
            Choose $e_{i+1} = v_i v_{i+1}$ in a different block than $e_i$.
        $H := H_{i,(i+1)}$   {the $i$-$(i+1)$ split of $H$ at $v_i$}
    Extend $T_i$ to $T_{i+1} = \langle v_0, e_1, v_1, \ldots, v_i, e_{i+1}, v_{i+1} \rangle$.
```

REMARKS

R12: Algorithms 4.2.1–4.2.3 can easily be adapted to construct an eulerian tour in a digraph: all one needs to do is choose e_{i+1} such that v_i is its tail since v_i is the head of e_i.

R13: The difference between the Splitting Algorithm and Fleury's Algorithm lies exclusively in the fact that the intermediate trails T_i, $0 \le i < q$, are stored separately as edge sequences, say, by Fleury's Algorithm, while the Splitting Algorithm retains them as part of the graphs considered. In both cases, however, it is the Splitting Lemma which guarantees the correctness of these algorithms (see [Fl90]). Observe that all even graphs are necessarily bridgeless.

4.2.3 Eulerian-Tour Enumeration and Other Counting Problems

The BEST-Theorem gives an explicit, computationally good formula for the number of eulerian tours in an eulerian digraph. It rests on the *Matrix Tree Theorem* (Fact 15) and can be applied to (undirected) graphs by summing over all balanced orientations of G. The latter, however, grows exponentially large with the number of vertices. We also briefly mention deBruijn (di)graphs because of their relevance to DNA-sequencing and other questions. DeBruijn digraphs are discussed in §4.4.

DEFINITIONS

D14: An ***out-tree*** in a digraph is a tree having a root of indegree 0 and all other vertices of indegree 1, and an ***in-tree*** is an out-tree with edges reversed.

D15: Let D be a digraph, $\boldsymbol{A}(D)$ its adjacency matrix with entries $a_{i,j}$, and let λ_i be the number of loops at $v_i \in V(D) = \{v_1, \ldots, v_n\}$. The ***Kirchhoff matrix*** $\boldsymbol{A}^*(D)$ with entries $a^*_{i,j}$ is defined by setting

$$a^*_{i,j} = -a_{i,j} \text{ if } i \neq j,\ a^*_{ii} = id(v_i) - \lambda_i; \qquad 1 \le i, j \le n$$

D16: Let a set $A = \{a_1, \ldots, a_n\}$ be called an ***alphabet*** whose ***letters*** are the elements of A. A k-letter ***word*** over A is an ordered k-tuple whose components are letters. A ***k-deBruijn sequence*** over A is a cyclic sequence of letters from A such that every k-letter word over A appears exactly once in this cyclic sequence.

D17: Let $n \geq 2, k \geq 2$. The ***deBruijn graph*** $D_{n,k}$ has as its vertices the length-$(k-1)$ words over an n-letter alphabet A; thus, there are altogether n^{k-1} vertices. For each length-k word $a_{i_1}, \ldots, a_{i_k}$ in the alphabet A, there is an arc of $D_{n,k}$ that joins the vertex $a_{i_1}, \ldots, a_{i_{k-1}}$ to the vertex $a_{i_2}, \ldots, a_{i_k}$.

TERMINOLOGY: For a matrix A, A_{ij} denotes the (i,j)-th ***minor***, i.e., the matrix obtained by deleting the i-th row and j-th column from A.

FACTS

F15: *Matrix Tree Theorem.* Given a digraph $D, V(D) = \{v_1, \ldots, v_n\}$, and let $\boldsymbol{A}^* = \boldsymbol{A}^*(D)$ be its Kirchhoff matrix. The number of spanning out-trees of D rooted at v_i, is det $\boldsymbol{A}^*_{i,i}$.

F16: In an eulerian digraph D, the number of spanning in-trees rooted at v_i equals the number of spanning out-trees rooted at v_i.

F17: For an eulerian digraph D, det $\boldsymbol{A}^*_{i,i} =$ det $\boldsymbol{A}^*_{j,j}$, $1 \leq i, j \leq n$.

F18: *BEST-Theorem.* [EhBr51, TuSm41] Let D be an eulerian digraph, and let $v_i \in V(D)$, $a \in E^+_{v_i}$ be chosen arbitrarily. The number of eulerian tours starting at v_i with the traversal of a is

$$\det \boldsymbol{A}^*_{i,i} \prod_{j=1}^{n} (od(v_j) - 1)!$$

F19: For an eulerian graph G with p vertices and q edges, and chosen $e \in E(G)$, the number $\mathcal{O}_E(G)$ of balanced orientations of G containing a fixed orientation of e, satisfies

$$\left(\frac{3}{2}\right)^{q-p} \leq \mathcal{O}_E(G) \leq 2^{q-p}.$$

F20: The deBruijn graph $D_{n,k}$ is a n-regular digraph ($id(v) = od(v) = n$ for every $v \in V(D_{n,k})$) with n^{k-1} vertices.

F21: There is a 1-1-correspondence between the set of k-deBruijn sequences over an n-letter alphabet and the set of eulerian tours of the deBruijn graph $D_{n,k}$. Consequently, and as an application of the BEST-Theorem, the number of k-deBruijn sequences over an n-letter alphabet is

$$\frac{(n!)^{n^{k-1}}}{n^k}$$

EXAMPLE

E2: The deBruijn graphs $D_{2,3}$ and $D_{2,4}$ are shown in Figure 4.2.5 (see also [ChOe93, p. 220]).

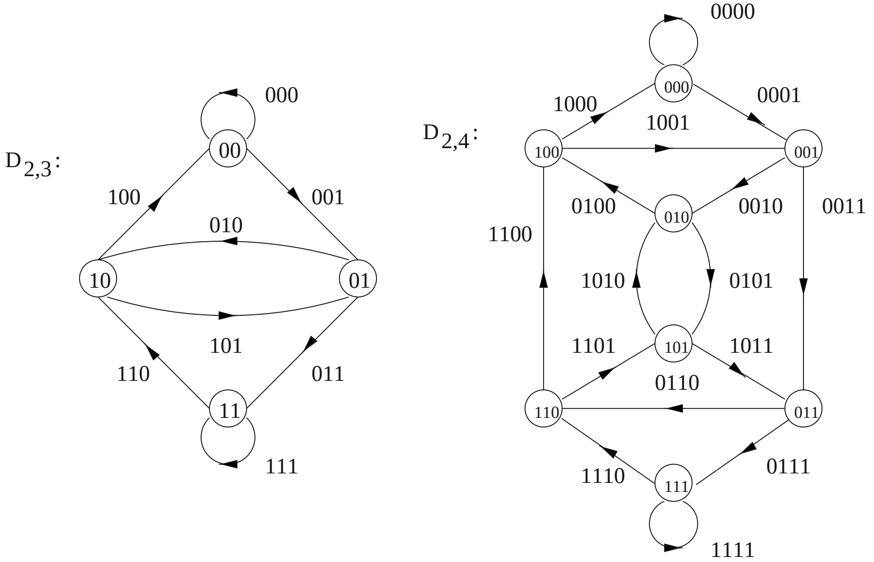


Figure 4.2.5 **The deBruijn digraphs $D_{2,3}$ and $D_{2,4}$**

REMARK

R14: DeBruijn graphs are of particular interest in the case $n = 2$, i.e., when the words are binary sequences. The study of these graphs $D_{2,k}$ are of particular interest in biochemistry when considering the problem of DNA sequencing. The same graphs are of interest, however, also in telecommunications when one is concerned with the question of ***network reliability***: apart from $D_{2,k}$, also Kautz graphs and hypercubes play an important role because these graphs perform well with respect to diameter and other parameters although the number of edges is relatively small in comparison to its number of vertices (see, e.g., [Xu02]).

4.2.4 Applications to General Graphs

In this subsection, we introduce some applications of eulerian graph theory to graph theory in general; some of these applications are also relevant in computer science and operations research, for example, the *Chinese Postman Problem* (§4.3). Interestingly, while certain analogues of results in eulerian graph theory hold equally well for general graphs, there are other quite natural analogues that lead to yet unsolved problems.

Covering Walks and Double Tracings

DEFINITIONS

D18: A ***covering walk*** (or ***postman tour***) in an arbitrary graph G is a closed walk containing every edge of G.

D19: A ***double tracing*** is a closed walk that traverses every edge exactly twice. A double tracing is ***bidirectional*** if every edge is used once in each of its two directions.

D20: A ***retract*** or ***retracing*** in a walk W is a section of the form v_{i-1}, e_i, v_i, e_{i+1}, v_{i+1} such that $e_i = e_{i+1}$ (and thus $v_{i+1} = v_{i-1}$). W is called ***retract-free*** if it has no retracts.

D21: A double tracing is called ***strong*** if it is both bidirectional and retract-free.

D22: The ***edge-connectivity*** of a connected graph G, denoted $\lambda(G)$, is the minimum number of edges whose removal can disconnect G. G is called ***k-edge-connected*** if $\lambda(G) \geq k$.

FACTS

F22: Let G be a graph with $2k$ vertices of odd degree, $k > 0$. Then G has a decomposition into k open trails whose initial and end vertices are of odd degree in G. Consequently, G has a decomposition into cycles and k paths; and if $k = 1$ and G is connected, then it has an eulerian trail.

F23: Every connected graph has a bidirectional double tracing. In a tree, every double tracing is bidirectional.

F24: [Sa77] A connected graph has a retract-free double tracing if and only if it has no end-vertices (vertices of degree 1).

F25: [Th87] If G is a graph without 1- and 3-valent vertices, then it has a strong double tracing. Consequently, every 4-edge-connected graph has a strong double tracing.

F26: [Tr66], [Th87] A connected 3-regular graph with $|V(G)| \equiv 0 \bmod 4$ has no strong double tracing.

F27: [Ve75] Let G be a connected graph, $E_0 \subseteq E(G)$. G has a double tracing using every $e \in E(G) - E_0$ twice in the same (not prescribed) direction and acting bidirectional on E_0, if and only if $G - E_0$ is an even graph. Observe that this implies Fact 23 (taking $E_0 = E(G)$).

REMARKS

R15: The condition for a double tracing to be bidirectional (Definition 19) applies to the case of self-loops if one views edges as composed of two half-edges, which allows a loop to be viewed as being also orientable in two ways.

R16: The double tracings quoted in Facts 23 and 25 can be obtained in polynomial time by reducing the respective problems to problems of finding eulerian tours satisfying certain restrictions, in eulerian digraphs derived from the given graphs by replacing every edge by two oppositely oriented arcs joining the same pair of vertices.

Maze Searching

In the context of this section, a *maze* may be viewed as a connected graph for which one has at each vertex, local information only. Tarry's algorithm is just one of several *maze-searching algorithms*. (See [Fl91] for a more extensive study.)

NOTATION: In the description of Algorithm 4.2.4, $e_{\text{in}}(v)$ denotes the edge that was traversed in visiting vertex v for the first time, and $E_{\text{left}}(v)$ denotes the set of edges that have been already traversed in leaving v.

```
Algorithm 4.2.4: Tarry's Algorithm [Ta,1895]
Input: a connected graph $G$.
Output: a bidirectional double tracing of $G$.
    Choose $v_0 \in V(G)$.
    Initialize $i = 0$ and $W = \langle v_0 \rangle$.
    While $(E_{v_i} - E_{\text{left}}(v_i) \neq \emptyset)$
        While $([E_{v_i} - E_{\text{left}}(v_i)] - \{e_{\text{in}}(v_i)\} \neq \emptyset)$
            Choose edge $e_i = v_i v_{i+1} \in [E_{v_i} - E_{\text{left}}(v_i)] - \{e_{\text{in}}(v_i)\}$.
            $W := W, \langle e_i, v_{i+1} \rangle$   { Extend $W$ to $v_{i+1}$ via edge $e_i$.}
            $i := i + 1$
        Let $e_i = v_i v_{i+1} = e_{\text{in}}(v_i)$
        $W := W, \langle e_i, v_{i+1} \rangle$
        $i := i + 1$
```

Covers, Double Covers, and Packings

DEFINITIONS

D23: A ***cycle cover*** of a graph G is a family S of cycles of G such that every edge of G belongs to at least one element of S.

D24: A cycle cover S is a ***cycle double cover*** (CDC) if every edge of G belongs to exactly two elements of S.

D25: A ***cycle packing*** in G is a set of edge disjoint cycles in G.

D26: A CDC S is called ***orientable*** if the elements of S can be cyclically oriented in such a way that every edge e is given opposite orientations in the two elements of S containing e.

CONJECTURES

Cycle Double Cover Conjecture (CDCC): Every bridgeless graph has a CDC.

Oriented Cycle Double Cover Conjecture: Every bridgeless graph has an oriented CDC.

Strong Cycle Double Cover Conjecture: Every bridgeless graph has a CDC containing a prescribed cycle of the graph.

Three Optimization Problems

DEFINITIONS

D27: Let G be a bridgeless, edge-weighted graph with weight function w: $E(G) \to R^+$. The ***weight of a cycle*** C in G, denoted $w(C)$, is given by $w(C) = \sum_{e \in E(C)} w(e)$. The weight of a cycle cover or cycle packing S is $w(S) = \sum_{C \in S} w(C)$.

D28: The ***Minimum-Weight Cycle-Cover Problem*** (MWCCP) is to find a cycle cover S in G such that $w(S)$ is minimum.

D29: The ***Maximum-Weight Cycle-Packing Problem*** (MWCPP) is to find a cycle packing S such that $w(S)$ is maximum.

D30: The ***Chinese Postman Problem*** is to find a minimum-weight covering walk W in G where $w(e)$ is counted as often as e is traversed by W. (See §4.3.)

FACTS

F28: [Fl86] Let G be a planar, bridgeless graph. Then G has an oriented CDC, and for any given cycle packing S, G has a CDC containing S as a subset. Thus, the strong Cycle Double Cover Conjecture is true for planar graphs.

F29: [FlGu85] The Undirected Chinese Postman Problem and the Maximum-Weight Cycle-Packing Problem are both solvable in polynomial time, and for planar, bridgeless graphs, the Minimum-Weight Cycle-Cover Problem can be solved in polynomial time.

F30: [FlGu85] Let G be an edge-weighted graph with weight function w. If W is a solution of the Undirected Chinese Postman Problem and S a solution of the Maximum-Weight Cycle-Packing Problem, then $w(S) = w(W) - 2w(E_d)$, where $E_d \subset E(G)$ is the set of those edges used twice in W, and $w(E_d) := \sum\limits_{e \in E_d} w(e)$.

F31: [FlGu85] For any planar, connected, bridgeless graph, if S is a solution of the Minimum-Weight Cycle-Cover Problem and W is a solution of the Undirected Chinese Postman Problem, then $w(S) = w(W)$.

F32: For any connected, bridgeless graph G with weight function w, if W is a solution of the Undirected Chinese Postman Problem and S is a solution of the Minimum-Weight Cycle-Cover Problem, then $w(S) \geq w(W)$. The Petersen graph (§1.2) shows that the inequality can be strict ($w(S) = 21$ and $w(W) = 20$, for $w \equiv 1$).

Nowhere-Zero Flows

DEFINITIONS

D31: Let $f\colon E(D) \to R$ be given for a digraph D. The function f is called a ***flow*** if for every $v \in V(D)$, $\sum\limits_{a \in E_v^+} f(a) = \sum\limits_{a \in E_v^-} f(a)$.

D32: Let $f\colon E(G) \to N$ be given for a graph G. Let D be an orientation of G with $a_e \in E(D)$ be the directed edge corresponding to $e \in E(G)$, and define $f'(a_e) := f(e)$. Then f is an ***integer flow*** in the graph G if f' is a flow in the digraph D.

D33: An integer flow f in G is ***nowhere-zero*** if $f(e) \neq 0$ for each edge $e \in E(G)$.

D34: A ***k-flow*** is an integer flow f such that $f(e) < k$ for each edge $e \in E(G)$.

CONJECTURE

Nowhere-Zero 5-Flow Conjecture (NZ5FC). Every bridgeless graph has a nowhere-zero 5-flow. [Tu54]

FACTS

F33: [Se81b] Every bridgeless graph has a nowhere-zero 6-flow.

F34: [Tu54] In a plane graph G, a (proper) k-face coloring of G corresponds to a nowhere-zero k-flow, and vice versa.

F35: A 3-regular graph G has a nowhere-zero 4-flow if and only if it is 3-edge-colorable, and it has a nowhere-zero 3-flow if and only if it is bipartite.

F36: To prove or disprove the NZ5FC and CDCC, one can assume without loss of generality that the graphs are 3-regular.

F37: [Se79] Let G be a bridgeless, planar graph, and let f: $E(G) \to Z^+$. Then the following two statements are equivalent:
(a) There exists a cycle cover S such that for every edge $e \in E(G)$,

$$e \text{ belongs to exactly } f(e) \text{ elements in } S$$

(b) For every edge-cut $E_0 \subseteq E(G)$,

$$\sum_{e \in E_0} f(e) \text{ is even and } \frac{1}{2} \sum_{e \in E_0} f(e) \geq \max\{f(e)\colon\ e \in E_0\}$$

REMARKS

R17: Double tracings in arbitrary connected graphs are the natural analogue to eulerian tours – Euler was already aware of that. Correspondingly, cycle double covers seem to be the natural analogue to cycle decompositions, yet their existence has been guaranteed so far only for certain classes of graphs, apart from the planar case. See [AlGoZh94], and [Zh97] for a thorough treatment of integer flows and cycle covers.

R18: Nowhere-zero flows can be viewed as eulerian tours in an eulerian multidigraph derived from an appropriate orientation of the given graph G, by replacing every arc a_e (corresponding to $e \in E(G)$) by $f(e)$ arcs with the same head and tail as a_e has.

4.2.5 Various Types of Eulerian Tours and Cycle Decompositions

DEFINITION

D35: Let G be an eulerian digraph and D_0 a subdigraph of G. If for every $v \in V(G)$, an eulerian trail T of G traverses every arc of D_0 incident from v, before it traverses any other arc incident from v, then T is called D_0***-favoring***.

FACTS

F38: [Ko56] Let G be a connected graph with vertex-set $V(G) = \{v_1, \ldots, v_n\}$ and having an even number of edges. Then G is eulerian if and only if G is the edge-disjoint union of graphs G_1, G_2 with $\deg_{G_1}(v_i) = \deg_{G_2}(v_i)$, $1 \leq i \leq n$; and if G is the union of two such graphs, then G has an eulerian tour in which the edges of G_1 and G_2 alternate.

F39: [Se81a, FlFr90] A planar even graph G has a decomposition into even cycles if and only if every block of G has an even number of edges.

F40: Let v be an arbitrary vertex of a strongly connected digraph G. Then there exists a spanning in-tree of G with root v.

F41: [EhBr51] Let D' be a spanning in-tree with root v in the eulerian digraph G, and let $D_0 = G - E(D')$. Then there exists a D_0-favoring eulerian tour of G starting and ending at v.

F42: [CaFl95] Let $\{e_1, \ldots, e_m\} \subseteq E(G)$ be an ordered set where G is eulerian. An eulerian tour T of the form $T = \ldots, e_1, \ldots, e_2, \ldots, e_m, \ldots$ exists if the edge-connectivity $\lambda(G) \geq m-1$; and if $\lambda(G) \geq 2m$, then one can even prescribe the direction in which these m edges are traversed by T.

REMARKS

R19: Fact 38 can be proved using the Splitting Lemma (Fact 11).

R20: Fact 39 is stated for planar graphs, but it can be extended to a more general class of graphs (see [Zh97]).

R21: D_0-favoring eulerian tours are studied in [FlWe89, Fl90]. However, in-trees are a special case of a more general class of digraphs D' for which there is a $(G - E(D'))$-favoring eulerian tour. We restricted Fact 41 to in-trees because of its relevance to enumerating eulerian tours in digraphs (see the BEST-Theorem [Fact 18]).

Incidence-Partition and Transition Systems

DEFINITIONS

D36: For each vertex v in a graph G, let $P(v) = \{E_1(v), \ldots, E_{k_v}(v)\}$, $k_v \geq 1$, be a partition of the incidence set E_v. Then $P(G) = \bigcup\limits_{v \in V} P(v)$ is called an ***incidence-partition system*** of G.

D37: A ***transition system*** of an even graph G, denoted $\tau(G)$, is an incidence-partition system $\tau(G) = \bigcup\limits_{v \in V} P(v)$ such that for every $v \in V(G)$, $|E_i(v)| = 2$ for every cell of the partition $P(v)$. Each cell $E_i(v)$ is called a ***transition***.

D38: An eulerian tour T and a cycle decomposition S give rise to transition systems, denoted τ_T and τ_S, respectively, in a natural way. Each transition in the ***eulerian-tour transition system*** τ_T is a pair of consecutive edges in the tour T. Similarly, each transition in the ***cycle-decomposition transition system*** τ_S is a pair of consecutive edges in a cycle $C \in S$.

TERMINOLOGY: A transition in τ_T and a transition in τ_S are referred to as a ***transition of*** T and a ***transition of*** S, respectively.

D39: Let $P(G)$ be an incidence-partition system of a graph G. An eulerian tour T is $P(G)$***-orthogonal*** (or ***orthogonal to*** $P(G)$) if no transition of T is a subset of any cell $E_i(v)$ of $P(G)$. $P(G)$-orthogonal cycle decompositions are defined analogously.

D40: A cycle decomposition S and an eulerian tour T are ***orthogonal*** if $\tau_S \cap \tau_T = \emptyset$.

TERMINOLOGY: The term "orthogonal" has been suggested by several authors as describing the underlying concept more accurately than the original term "compatible".

D41: An incidence-partition system $P(G)$ satisfies the ***cut condition*** if for every vertex subset X, the edge-cut $E(X, \overline{X})$ satisfies $|E(X, \overline{X}) \cap E_i(v)| \leq \frac{1}{2}|E(X, \overline{X})|$ for every cell $E_i(v)$ of $P(G)$.

FACTS

F43: [Ko68] A loopless eulerian graph G has an eulerian tour orthogonal to a given partition system $P(G)$ if and only if $P(G)$ satisfies the cut condition restricted to the edge-cuts E_v, $v \in V(G)$.

F44: [Fl80] Given a cycle decomposition S of the eulerian graph G with $\deg(v) > 2$ for every $v \in V(G)$, there exists an eulerian tour orthogonal to S.

F45: [Fl80] Let T be an eulerian tour of the eulerian graph G. If $\deg(v) \equiv 0 \bmod 4$ for every $v \in V(G)$, then there exists a cycle decomposition orthogonal to T.

F46: [FlFr90] Let G be a planar, even, loopless graph with incidence-partition system $P(G)$. Then G has a $P(G)$-orthogonal cycle decomposition if and only if $P(G)$ satisfies the cut condition.

F47: [Fl80] Let G be a planar eulerian graph and let T be an eulerian tour of G. If $\deg(v) > 2$ for every $v \in V(G)$, then G has a cycle decomposition orthogonal to T.

EXAMPLE

E3: The complete graph K_5 in Figure 4.2.6, with transition system $\tau(K_5) = \{\{i, i{+}1\}, \{i', (i+1)'\}$: $1 \leq i \leq 5$, setting $6 = 1\}$ has no $\tau(K_5)$-orthogonal cycle decomposition, which shows that Fact 46 cannot be generalized to arbitrary non-planar graphs.

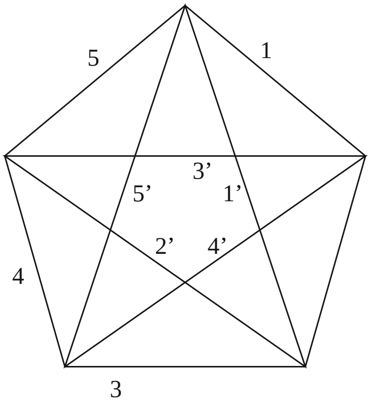


Figure 4.2.6 K_5 **having no** $\tau(K_5)$**-orthogonal cycle decomposition.**

REMARKS

R22: To produce a cycle decomposition S orthogonal to a given eulerian tour in a graph with $\deg(v) \equiv 0 \bmod 4$ for every $v \in V(G)$, one can apply a procedure developed by J. Petersen in his celebrated paper [Pe,1891]: Color the edges of T alternately 'blue' and 'red', and combine a cycle decomposition of the blue even graph with one of the red even graph.

R23: Fact 45 follows from Fact 38 by using the classical characterization (Fact 1).

R24: Fact 44 is basically a special case of Fact 43. We stated it separately because its converse (given an eulerian tour T, there exists a cycle decomposition orthogonal to T) is an open problem known as *Sabidussi's Compatibility Conjecture.* Its relevance to other open problems such as the Cycle Double Cover Conjecture and the Nowhere-Zero 5-Flow Conjecture is discussed in [FL84, Fl88, Fl01, Fl02].

R25: Facts 43 and 46 show that the existence of eulerian tours satisfying certain restrictions does not necessarily imply the existence of cycle decompositions satisfying the same restrictions: Fact 43 relates to *arbitrary* loopless graphs and uses the cut condition only locally, whereas in Fact 46, the full strength of the cut condition is invoked.

R26: While Facts 46 and 47 have been formulated for planar graphs only, they can be extended to a somewhat more general class of graphs (see [Zh97]).

R27: $P(G)$-orthogonal eulerian tours in digraphs have been studied in [Fl90]. Naturally, due to the appearance of arcs instead of edges, somewhat stronger conditions than the cut condition of Definition 41 are needed to prove the existence of $P(G)$-orthogonal eulerian tours.

Orderings of the Incidence Set, Non-Intersecting Tours, and A-Trails

DEFINITIONS

D42: Given a graph G and a vertex v, a fixed sequence $\langle e_1, e_2, \ldots, e_{\deg(v)}\rangle$ of the edges in the incidence set E_v is called a ***positive ordering of*** E_v and is denoted $O^+(v)$. If G is imbedded in some surface, one such $O^+(v)$ is given by the counterclockwise cyclic ordering of the edges incident on v.

D43: Let G be an even graph and v a vertex with $\deg(v) \geq 4$ and with a positive ordering of its incident set E_v given by $O^+(v) = \langle e_1, e_2, \ldots, e_{\deg(v)}\rangle$. A transition system $\tau(G)$ is ***non-intersecting with respect to*** $O^+(v)$ if for any $e_i, e_j, e_k, e_l \in E_v$ with $i < j < k < l$, $\{e_i, e_k\}$ and $\{e_j, e_l\}$ cannot both be transitions of $\tau(G)$. That is,

$$\{e_i, e_k\} \in \tau(G) \Rightarrow \{e_j, e_l\} \notin \tau(G)$$

D44: Let G be an even graph with a given positive ordering $O^+(v)$ for each $v \in V$. A transition system $\tau(G)$ is ***non-intersecting*** if $\tau(G)$ is non-intersecting with respect to $O^+(v)$ for every $v \in V$ with $\deg(v) \geq 4$. An eulerian tour T and a cycle decomposition S are non-intersecting if their corresponding transition systems, τ_T and τ_S, respectively, are non-intersecting.

D45: Let G be an eulerian graph with a given positive ordering $O^+(v)$ for each $v \in V$. An eulerian tour T is an ***A-trail*** if $\{e_i, e_j\} \in \tau_T$ implies $j = i + 1$ or $j = i - 1$ (modulo $\deg(v)$).

D46: An ***outerplanar graph*** is a graph with an imbedding in the plane such that every vertex appears on the boundary of the exterior face.

D47: A graph (imbedding) ***triangulates*** a surface if every region is 3-sided.

EXAMPLE

E4: An A-trail T in the octahedron, given by the sequence $1, 2, 3, \ldots, 11, 12$, is shown in Figure 4.2.7 below. A cycle decomposition orthogonal to T is given by the sets $\{2, 6, 10\}$, $\{4, 8, 12\}$, $\{1, 11, 9, 7, 5, 3\}$.

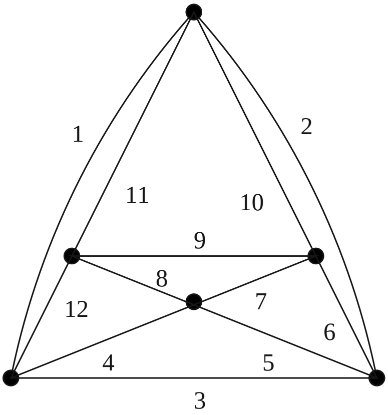


Figure 4.2.7 **An A-trail in the octahedron.**

FACTS

F48: Given an eulerian graph G and $O^+(v)$ for every $v \in V(G)$, a non-intersecting eulerian tour exists.

F49: In an eulerian graph with $\deg(v) \leq 4$ for every $v \in V(G)$, the concepts of non-intersecting eulerian tour and A-trail are equivalent.

F50: [AnFl95] The decision problem whether a given simple, planar, 3-connected eulerian graph has an A-trail is NP-complete.

F51: [AnFlRe98] Simple, outerplanar, eulerian graphs have A-trails; they can be constructed in polynomial time.

F52: Let G be a simple eulerian graph that triangulates the plane. Suppose that G has maximum degree $\delta_{\max}(G) \leq 8$ with at most one 8-valent vertex, which, if it exists, is adjacent to a 4-valent vertex. Then G has an A-trail.

REMARKS

R28: Facts 43, 44, and 48 can be proved by employing the Splitting Lemma (Fact 11). Consequently, algorithms for constructing eulerian tours and that are based on the Splitting Lemma, can be modified so as to yield $P(G)$-orthogonal eulerian tours or non-intersecting eulerian tours.

R29: There is a 1-1-correspondence between transition systems $\tau(G)$ of the even graph G and the decompositions of $E(G)$ into closed trails: traversing edges of G following the given transitions results in closed trails, one at a time; together they form a decomposition into closed trails. Likewise, each of these trails defines a subset of $\tau(G)$ (for a given $\tau(G)$), and since these trails are edge-disjoint, the union of the subsets is $\tau(G)$.

4.2.6 Transforming Eulerian Tours

The Kappa Transformations

The kappa transformations consist of various combinations of splitting, splicing, and reversing closed trails. They form the basis for constructing eulerian tours and for transforming one eulerian tour into another. For a detailed discussion, see, e.g., [Fl90].

DEFINITIONS

D48: The ***reverse of a trail*** $T = \langle v_0, e_1, v_1, \ldots, e_l, v_l\rangle$ is the trail

$$T^{-1} = \langle v_l, e_l, v_{l-1}, \ldots, e_1, v_0\rangle$$

D49: Let $T = \langle \ldots, e_i, v_i, e_{i+1}, \ldots, e_j, v_j, e_{j+1}, \ldots\rangle$ be an eulerian tour in a graph G such that $v_i = v_j$, and consequently, $\{e_i, e_{i+1}, e_j, e_{j+1}\} \subseteq E_{v_i}$. The closed subtrail $\langle v_i, e_{i+1}, \ldots, e_j, v_j\rangle$ is called a ***segment*** of tour T and is denoted $S_{i,j}$.

D50: A ***segment reversal*** (or ***κ-transformation***) is the replacement of one of the segments in an eulerian tour T by its reverse segment. The resulting eulerian tour is denoted $\kappa(T)$. Thus, if tour $T = \langle \ldots, e_i, S_{i,j}, e_{j+1}, \ldots\rangle$, then

$$\kappa(T) = \langle \ldots, e_i, S_{i,j}^{-1}, e_{j+1}, \ldots\rangle$$

D51: Let $T = \langle \ldots, e_i, S_{i,j}, e_{j+1}, \ldots\rangle$ be an eulerian tour of a graph G with segment $S_{i,j}$. The 2-cell partition of $E(G)$ consisting of the edge set of $S_{i,j}$ and the edge set of the ("rest of the way around") segment $S_{j,i} = \langle v_j, e_{j+1}, \ldots, e_i, v_i\rangle$ is called a ***κ-detachment*** and is denoted $\kappa'(T)$.

D52: Given a trail decomposition of $E(G)$ into closed trails $T_1, \ldots, T_k$, $k \geq 2$, choose trails T_i and T_j such that $v \in V(T_i) \cap V(T_j)$ for some vertex v. Let $e_{m,i}$, $e_{m+1,i} \in E_v$ be consecutive in T_i, and $e_{n,j}$, $e_{n+1,j} \in E_v$ consecutive in T_j (i.e., they are transitions of their respective trails). Thus, we may write $T_i = \langle \ldots, e_{m,i}, v, e_{m+1,i}, \ldots\rangle$ and $T_j = \langle v, e_{n,j}, \ldots, e_{n+1,j}, v\rangle$. A ***splice at*** v of trail T_j into trail T_i (or the ***κ-absorption*** at v) of T_j by T_i is either one of the closed trails:

$$\text{splice}(T_i, T_j, v) = \langle \ldots, e_{m,i}, T_j, e_{m+1,i} \ldots\rangle$$
$$\text{splice}(T_i, T_j^{-1}, v) = \langle \ldots, e_{m,i}, T_j^{-1}, e_{m+1,i} \ldots\rangle$$

NOTATION: Either one of the closed trails that result from a splice of T_j into T_i is denoted $\kappa''(\{T_i, T_j\})$.

D53: Let T be an eulerian tour in a graph G. An eulerian tour T' is obtained from T by a ***κ^*-transformation***, denoted $T' = \kappa^*(T)$, if there exists a κ-detachment $\kappa'(T) = \{S_{i,j}, S_{j,i}\}$ such that $T' = \kappa''\{S_{i,j}, S_{j,i}\}$. That is, $T' = \kappa^*(T) = \kappa'(\kappa''(T))$.

D54: Let T_1 and T_2 be two eulerian tours in a graph G. Tour T_2 is obtained from T_1 by a κ_1-transformation, denoted $T_2 = \kappa_1(T_1)$, if either $T_2 = \kappa(T_1)$ or $T_2 = \kappa^*(T_1)$.

D55: Two eulerian trails, T_1 and T_2, are considered different if their corresponding transition systems are different, i.e., if $\tau_{T_1} \neq \tau_{T_2}$.

REMARK

R30: The various transformations defined above carry over to eulerian digraphs with the added restriction that each transition at a vertex v must comprise an arc incident *to* v and an arc incident *from* v.

FACTS

F53: Let T_1 and T_2 be two different eulerian tours of an eulerian graph G (they exist unless G is a cycle). T_2 can be obtained from T_1 by a sequence of κ-transformations (see [AbKo80], [Sk83], [Fl90]).

F54: Let G be an eulerian graph with a partition system $P(G)$, and suppose that T_1 and T_2 are different $P(G)$-orthogonal eulerian tours. Then T_2 can be obtained from T_1 by a sequence of κ_1-transformations in such a way that any eulerian tour and any trail decomposition S with $|S| = 2$ arising in this sequence are $P(G)$-orthogonal.

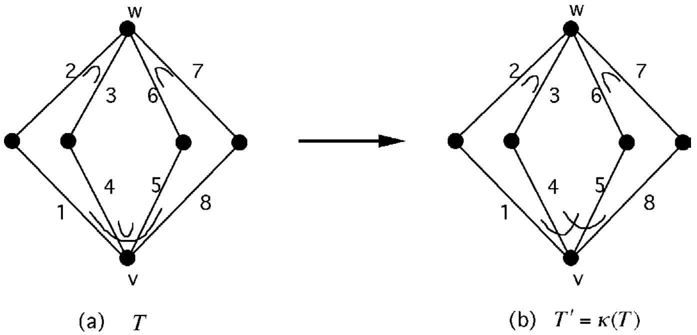
F55: Let G be an eulerian graph with a given positive ordering $O^+(v)$ for every $v \in V(G)$, and let T_1 and T_2 be different non-intersecting eulerian tours of G. Then T_2 can be obtained from T_1 by a sequence of κ_1-transformations in such a way that any eulerian tour and any trail decomposition S with $|S| = 2$ arising in this sequence are non-intersecting.

F56: Let T_1 and T_2 be two different eulerian tours in a digraph G. Then tour T_2 can be obtained from T_1 by a sequence of κ_1-transformations.

F57: In 4-regular plane graphs, A-trails (which are non-intersecting eulerian tours in this case) are in 1-1-correspondence with spanning trees in an (easily constructed) auxiliary graph. The κ_1-transformations correspond to the edge-addition and edge-deletion process in transforming one spanning tree into another spanning tree.

EXAMPLES

E5: The complete bipartite graph $K_{2,4}$ with eulerian tour $T = \langle 1, 2, 3, \ldots, 8\rangle$ (written as edge sequence) is shown in Figure 4.2.8(a). The transitions at v and w are marked with little arcs. Tour T is transformed into the eulerian tour $T' = \langle 1, 2, 3, 4, 8, 7, 6, 5\rangle$ by a κ-transformation (segment reversal) at v (see Figure 4.2.8(b)).



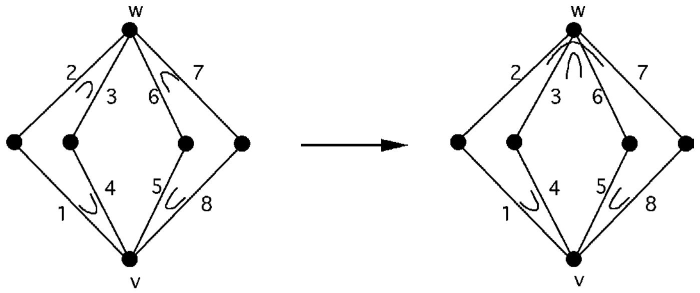
(a) T

(b) $T' = \kappa(T)$

Figure 4.2.8

E6: The tour T in Example 5 is a non-intersecting eulerian tour. By a κ-detachment at v, one obtains the non-intersecting trail decomposition $S = \{T_1, T_2\}$ with $T_1 = \langle 1, 2, 3, 4\rangle$ and $T_2 = \langle 5, 6, 7, 8\rangle$ (written as edge sequences) (Figure 4.2.9(a)). A κ-absorption at

w results in $T'' = \langle 1,2,7,8,5,6,3,4\rangle$, another non-intersecting eulerian tour (Figure 4.2.9(b)).



(a) $S = \{T_1, T_2\}$, $S = \kappa'(T)$ (b) $T'' = \kappa_1(T) = \kappa''(S)$

Figure 4.2.9

Splicing the Trails in a Trail Decomposition

We close the section with an eulerian-tour construction by A. Tucker that starts with a closed-trail decomposition and iteratively splices pairs of trails together (i.e., performs κ-absorptions) until there is only one trail left.

Algorithm 4.2.5: Tucker's Algorithm [Tu76]

```
Input: eulerian graph $G$.
Output: eulerian tour $T$.
    Produce a trail decomposition of $G$ by forming an arbitrary 2-regular
      detachment $H$ of $G$.
    Let $W = \{T_1, \ldots, T_k\}$ be the set of components of $H$.
    While $k \geq 2$
        Choose $T_i, T_j$ with $i \neq j$ such that $V(T_i) \cap V(T_j) \neq \emptyset$.
        Let $v \in T_i \cap T_j$.
        Let $T_{i,j} = \kappa''(T_i, T_j)$   {a $\kappa$-absorption at $v$}.
        $W := W \cup \{T_{i,j}\} - \{T_i, T_j\}$
        $k := k - 1$
```

References

[AbKo80] J. Abraham and A. Kotzig, Transformations of Euler tours, *Ann. Discrete Math.* **8** (1980), 65–69.

[AlGoZh94] B. Alspach, L. Goddyn, and C-Q. Zhang, Graphs with the circuit cover property, *Transactions Amer. Math. Soc.* **344, 1** (1994), 131–154.

[AnFl95] L. D. Andersen and H. Fleischner, The NP-completeness of finding A-trails in eulerian graphs and of finding spanning trees in hypergraphs, *Discrete Appl. Math.* **59** (1995), 203–214.

[AnFlRe98] L. D. Andersen, H. Fleischner, and S. Regner, Algorithms and outerplanar conditions for A-trails in plane eulerian graphs, *Discrete Appl. Math.* **85** (1998), 99–112.

[CaFl95] M-C. Cai and H. Fleischner, An eulerian trail traversing specified edges in given order, *J. Graph Theory* **19** (1995), 137–144.

[ChOe93] G. Chartrand and O. R. Oellermann, *Applied and Algorithmic Graph Theory*, Internat. Series in Pure and Applied Math., McGraw-Hill, 1993.

[EhBr51] T. van Aardenne-Ehrenfest and N. G. de Bruijn, Circuits and trees in oriented linear graphs, *Simon Stevin* **28** (1951), 203–217.

[Eu,1736] L. Euler, Solutio problematis ad geometriam situs pertinentis, *Commentarii Academiae Petropolitanae* **8** (1736), 1741, 128–140 = Opera omnia Ser. I, Vol. 7, 1–10.

[Fl,1883] Fleury, Deux problemes de geometrie de situation, *Journal de mathematiques elementaires* 1883, 257–261.

[Fl80] H. Fleischner, Eulersche Linien und Kreisüberdeckungen, die vorgegebene Durchgänge in den Kanten vermeiden, *J. Comb. Theory* **B29, 2** (1980), 145–167.

[Fl84] H. Fleischner, Cycle decompositions, 2-coverings, removable cycles, and the 4CD, in *Progress in Graph Theory*, J. A. Bondy and U. S. R. Murty (Eds.), Academic Press, 1984, 233–246.

[Fl86] H. Fleischner, Proof of the strong 2-cover conjecture for planar graphs, *J. Combin. Theory* **B40, 2** (1986), 229–230.

[Fl88] H. Fleischner, Some blood, sweat, but no tears in eulerian graph theory, *Congressus Numer.* **63** (1988), 8–48.

[Fl89] H. Fleischner, Elementary proofs of (relatively) recent characterizations of eulerian graphs, *First Montreal Conference on Combinatorics and Computer Science*, 1987, *Discrete Appl. Math.* **24** (1989), No. 1–3, 115–119.

[Fl90] H. Fleischner, *Eulerian Graphs and Related Topics, Part 1, Vol. 1*, Ann. Discrete Math **45** North-Holland, Amsterdam, 1990.

[Fl91] H. Fleischner, *Eulerian Graphs and Related Topics, Part 1, Vol. 2*, Ann. Discrete Math **50** North-Holland, Amsterdam, 1991.

[Fl00] H. Fleischner, *Traversing Graphs: The Eulerian and Hamiltonian Theme* in *ARC ROUTING: Theory, Solutions, and Applications*, M. Dror (Ed.), Kluwer Academic Publishers, 2000, 19–87.

[Fl01] H. Fleischner, (Some of) The many uses of eulerian graphs in graph theory (plus some applications), *Discrete Math.* **230** (2001), 23–43.

[Fl02] H. Fleischner, Bipartizing matchings and Sabidussi's compatibility conjecture, *Discrete Math.* **244** (2002), 77–82.

[FlFr90] H. Fleischner and A. Frank, On circuit decompositions of planar eulerian graphs, *J. Comb. Theory* **B50** (1990), 245–253.

[FlGu85] H. Fleischner and M. Guan, On the minimum weighted cycle covering problem for planar graphs, *Ars Combinatoria* **20** (1985), 61–67.

[FlWe89] H. Fleischner and E. Wenger, D_0-favouring eulerian trails in digraphs, *Arch. Math. (Brno)* **25** (1989), 55–60.

[FoFu62] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ, 1962.

[Hi,1873] C. Hierholzer, Über die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren, *Math Annalen* **VI** (1873), 30–32.

[Ko56] A. Kotzig, Euler lines and decompositions of a regular graph of even order into two factors of equal orders (in Slovak), *Mat.-Fyz. Časopis Slovensk. Akad.* **6** (1956), No. 3, 133–136.

[Ko68] A. Kotzig, Moves without forbidden transitions in a graph, *Mat.-Fyz. Časopis* **18** (1968), No. 1, 76–80.

[Lu,1894] M. É. Lucas, *Récréations Mathématiques IV*, Gauthiers–Villars et fils, Paris, 1894.

[Mc84] T. A. McKee, Recharacterizing eulerian: Intimations of new duality, *Discrete Math.* **51** (1984), 237–242.

[Na79] C. St. J. A. Nash-Williams, Acyclic detachments of graphs, in *Graph Theory and Combinatorics*, R. J. Wilson (Ed.), Proc. Conf. Open University; Milton Keynes, 1978, *Res. Notes in Math.* **34**, Pitman, San Francisco (1979), 87–97.

[Na85a] C. St. J. A. Nash-Williams, Detachment of graphs and generalized Euler trails, in *Surveys in Combinatorics*, I. Anderson (Ed.), 1985, *Math. Soc. Lecture Notes Ser.* **103**, Cambridge University Press, London, (1985), 137–151.

[Na85b] C. St. J. A. Nash-Williams, Connected detachments of graphs and generalized Euler trails, *J. London Math. Soc.* **31** (1985), No. 2, 17–29.

[Pe1891] J. Petersen, Die Theorie der regulären Graphs, *Acta Math.* **15** (1891), 193–220.

[Sa77] G. Sabidussi, Tracing graphs without backtracking, in *Methods of Operations Research XXV, Part 1*, Henn et al. (Eds.), First Symp. on Oper. Res., Univ. Heidelberg, Sept. 1–3, 1976.

[Se79] P. D. Seymour, Sums of circuits, in *Graph Theory and Related Topics*, J. A. Bondy and U. S. R. Murty (Eds.), Academic Press, New York 1979, 341–356.

[Se81a] P. D. Seymour, Even circuits in planar graphs, *J. Combin. Theory* **B31** (1981), 327–338.

[Se81b] P. D. Seymour, Nowhere-zero 6-flows, *J. Comb. Theory* **B30** (1981), 130–135.

[Sh79] H. Shank, Some parity results on binary vectors spaces, *Ars. Combin.* **8** (1979), 107–108.

[Sk83] D. K. Skilton, Eulerian chains and segment reversals, in *Graph Theory, Proc. First Southeast Graph Theory Colloq.*, K. M. Koh and H. P. Yap (Eds.) Singapore, May 1983, Lecture Notes in Math. **1073**, Springer, Berlin-New York, 1984, 228–235.

[Ta,1895] G. Tarry, Le problème des labyrinthes, *Nouv. Ann. Math.* 14 (1895), 187–190.

[Th85] C. Thomassen, Retracting-free double tracings of graphs, *Ars Combinatoria* **19** (1985), 63–68.

[To73] S. Toida, Properties of a Euler graph, *J. Franklin Inst.* **295** (1973), 343–345.

[Tr66] D. J. Troy, On traversing graphs, *Amer. Math. Monthly* **73** (1966), 497–499.

[Tuc76] A. Tucker, A new applicable proof of the Euler circuit theorem, *Amer. Math. Monthly* **83** (1976), 638–640.

[Tu54] W. T. Tutte, A contribution to the theory of chromatic polynomials, *Canad. J. Math.* **6** (1954), 80–91.

[TuSm41] W. T. Tutte and C. A. B. Smith, On unicursal paths in a network of degree 4, *Amer. Math. Monthly* **48** (1941), 233–237.

[Ve12] O. Veblen, An application of modular equations in analysis situs, *Ann. of Math* **(2)14** (1912/13), 86–94.

[Ve31] O. Veblen, Analysis situs, *Amer. Math. Soc. Colloq. Publ.* **5**, Part II (1931), 1–39.

[Ve75] P. D. Vestergaard, Doubly traversed Euler circuits, *Arch. Math.* **26** (1975), 222–224.

[Wo90] D. R. Woodall, A proof of McKee's eulerian-bipartite characterization, *Discrete Math.* **84** (1990), 217–220.

[Xu02] J. Xu, *Topological Structure and Analysis of Interconnection Networks*, Network Theory and Applications **7**, Kluwer Academic Publishers, Dordrecht, 2002.

[Zh97] C-Q. Zhang, *Integer Flows and Cycle Covers of Graphs*, Marcel Dekker Inc., New York, 1997.

4.3 CHINESE POSTMAN PROBLEMS

R. Gary Parker, Georgia Institute of Technology

4.3.1 The Basic Problem and Its Variations
4.3.2 Undirected Postman Problems
4.3.3 Directed Postman Problems
4.3.4 Mixed Postman Problems
References

Introduction

The *Chinese Postman Problem* (CPP) is one of the more celebrated problems in graph optimization. It acts as a useful model in an array of practical contexts such as refuse collection, snow removal, and mail delivery. The basic problem appears to have been first posed by the mathematician Guan (or Kwan Mei-Ko) in 1962 [Gu62]; an attribute that apparently resulted in its being dubbed "Chinese" by Jack Edmonds ([Ed65-a]).

4.3.1 The Basic Problem and Its Variations

DEFINITIONS

D1: A ***postman tour*** in a graph G is a closed walk that uses each edge of G *at least* once.

D2: Given a finite graph, $G = (V, E)$, with edges weighted as $w : E \to Z^+$, the ***Chinese Postman Problem*** seeks a minimum-weight postman tour.

TERMINOLOGY: The basic problem is sometimes simply referred to as a ***postman problem***.

D3: The ***undirected version of CPP (UCPP)*** assumes that the instance graph G is an undirected graph.

D4: The ***directed version of CPP (DCPP)*** assumes that the instance graph G is a digraph.

D5: The ***mixed version of CPP (MCPP)*** assumes that the instance graph G is a *mixed graph*, that is, some edges are directed, and some edges are undirected.

FACTS

F1: UCPP and DCPP are polynomially solvable. (See §4.3.2 and §4.3.3.)

F2: MCPP is NP-hard. (See §4.3.4.)

The Eulerian Case

DEFINITIONS

D6: An ***eulerian tour*** in a graph (or digraph) G is a closed walk that uses each edge (or arc) of G *exactly* once. An eulerian tour in a mixed graph is a closed walk that uses each edge and each arc exactly once.

D7: A graph, digraph, or mixed graph that has an eulerian tour is called ***eulerian***.

D8: A digraph G is ***strongly connected*** if for every two of its vertices, u and v, there is a directed walk from u to v and one from v to u.

D9: If $G = (V, E)$ is a digraph, a vertex $v \in V$ is ***symmetric*** if the indegree and outdegree of v are equal.

TERMINOLOGY: If every vertex in a digraph G is symmetric, then G is sometimes referred to as a ***symmetric digraph***. However, in other contexts, "symmetric digraph" is sometimes taken to mean $(x, y) \in E \rightarrow (y, x) \in E$ for all $x, y \in V$.

FACTS

F3: A connected graph G is eulerian if and only if every vertex of G has even degree.

F4: A strongly connected digraph G is eulerian if and only if G is symmetric.

F5: If the instance graph G (undirected, directed, or mixed) is eulerian, then CPP is solved by producing an eulerian tour.

REMARK

R1: A characterization of eulerian mixed graphs is given later in the subsection Mixed Postman Problems.

Variations of CPP

DEFINITIONS

D10: ***open postman tour***: The postman is required to start and end at distinct vertices of the graph (or digraph).

D11: ***not requiring a specified edge***: A specified edge is not *required* to be in an admissible tour but its inclusion is at least permitted.

D12: ***a specified edge cannot be duplicated***: A specified edge is required to be present in the postman tour but cannot be duplicated (i.e., cannot be traversed more than once).

D13: ***windy postman problem***: Instances of UCPP place no restriction on the direction of traversal along an edge. This does not suggest, however, that in a practical application, the postman need necessarily experience the same "cost" of traversal in both directions (suppose the edge-weight metric that is relevant is not distance but rather time). If one allows edge weights to differ depending upon which direction an edge is traversed, the problem becomes the *windy postman problem*.

D14: ***rural postman problem***: This variant, also motivated by practical settings, arises when rather than requiring that all edges or arcs be traversed at least once, only a given subset has to be used. This version derives its name from the apparent case of postal delivery in non-urban settings where, perhaps, the postman may have to traverse every street within a small town or village and then moves on to another one but can do so by selecting any of a number of connecting roads (edges) that exist to connect the towns.

D15: ***stacker crane problem***: This is the rural postman problem for mixed graphs.

FACTS

F6: The open postman problem remains polynomially solvable. If v_1 and v_2 are the pre-specified (distinct) vertices, then all that is required is to add an artificial vertex to G, say v_a, connect it to v_1 and v_2 by two (artificial) edges and assign the new edges a weight of M, where M is sufficiently large. Clearly, in the application of the algorithm, the artificial edges would never be part of a shortest path and hence, would never be duplicated. In the resultant $\overline{G}$, one would simply find an eulerian tour starting and stopping at v_a; removing v_a from this tour induces the desired *open eulerian walk* (or *semieulerian tour*) from v_1 to v_2.

F7: For the second and third variations above, where a specified edge is not required or cannot be duplicated, the problem remains polynomially solvable (*cf.* [EdJo73]).

F8: [Wi92] The windy postman problem is NP-hard although solvable in polynomial time if instances are eulerian.

F9: The rural postman problem is NP-hard on both graphs and digraphs even if all edges/arcs have the same weight (see [GaJo79]).

REMARK

R2: It is important to note that the list of extensions presented here is not exhaustive. Additional variations to these sorts of problems in general are often easy to create whether motivated by purely combinatorial interests or ones more pragmatic, stemming from a given practical setting. A good starting place for a sense of the breadth of cases, degree of analysis, and categorization of results is the rather expansive survey in [EiGeLa95-a] and [EiGeLa95-b]. Extensive coverage for the Chinese Postman Problem and its variations may also be found in [Fl91].

4.3.2 Undirected Postman Problems

The solution posed by Guan, though clever, was not fast in the universally adopted, complexity-theoretic sense (i.e., not polynomial in the size of the input graph). This flaw was pointed out by Edmonds ([Ed65-a]) who then proposed a polynomial algorithm for the problem.

DEFINITIONS

D16: A ***matching M*** in a graph G is a subset of edges no two of which have a common vertex. (Matchings are discussed in §11.3.)

D17: A matching is ***perfect*** if every vertex in G is incident to some edge in the matching.

Algorithm 4.3.1: Solution to UCPP

Input: Connected graph G with edges weighted by nonnegative integer values.
Output: Minimum-weight postman tour in G.

```
Let $V_O$ be the set of vertices with odd degree in $G$.
For each pair of vertices $x, y \in V_0$
    Find a shortest path $P$ in $G$ between $x$ and $y$.
Form a complete graph $K$ on the vertex set $V_O$ with edges weighted by
  the respective shortest-path lengths.
Find a minimum-weight perfect matching $M$ in $K$.
For each edge $e \in M$
    Duplicate the edges in $G$ of the shortest path $P$ corresponding to $e$.
Let $\overline{G}$ be the resulting supergraph.
Produce an eulerian tour in $\overline{G}$.
```

COMPUTATIONAL NOTE: The shortest path computation in Algorithm 4.3.1 is straightforward and fast. Producing a minimum-weight perfect matching in a graph, although complicated, can be accomplished in polynomial effort following the seminal work by Edmonds (*cf.* [Ed65-b],[Ed65-c]). The implementation of the step relative to traversal-finding in $\overline{G}$ is also easy. The following strategy by Fleury (*cf.* [Ka67]) can be applied recursively: Given a position in the walk, select the next edge arbitrarily so long as its removal would not disconnect the graph $\overline{G}$, *unless* this is the only choice. (Fleury's algorithm appears in §4.2.)

EXAMPLE

E1: Consider G on the left in Figure 4.3.1; weights are specified directly on the edges. The set V_O is given by $\{v_1, v_3, v_4, v_5\}$ and the stated shortest paths along with the path lengths result as follows:

Vertex Pair	Path	Length
v_1, v_3	v_1, e_1, v_2, e_2, v_3	3
v_1, v_4	$v_1, e_1, v_2, e_7, v_5, e_4, v_4$	5
v_1, v_5	v_1, e_1, v_2, e_7, v_5	3
v_3, v_4	v_3, e_3, v_4	3
v_3, v_5	v_3, e_2, v_2, e_7, v_5	2
v_4, v_5	v_4, e_4, v_5	2

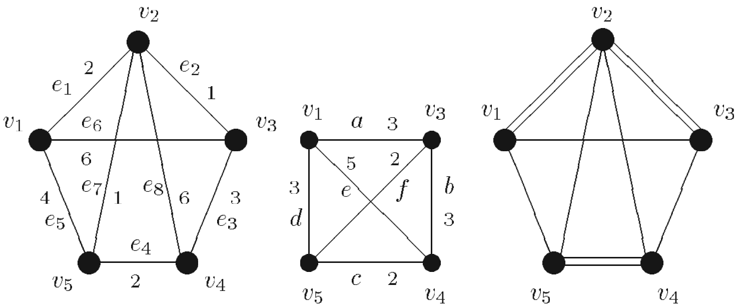


Figure 4.3.1 Application of Algorithm 4.3.1.

An optimal matching in the complete graph K_4, shown in the middle of Figure 4.3.1, consists of edges a and c, having total weight 5. These edges correspond to the paths

v_1, e_1, v_2, e_2, v_3 and v_4, e_4, v_5. The respective edges in these paths are duplicated in G producing the multigraph, $\overline{G}$, shown on the right in Figure 4.3.1. The latter is eulerian, and an eulerian tour (of total weight 30) is given by the walk below:

$$v_1, e_1, v_2, e_2, v_3, e_2, v_2, e_1, v_1, e_6, v_3, e_3, v_4, e_4, v_5, e_7, v_2, e_8, v_4, e_4, v_5, e_5, v_1$$

REMARKS

R3: The stipulation that edges be weighted by nonnegative integer values cannot be relaxed since otherwise negative weight closed walks result in G (by simply going back and forth on such an edge), which creates an intractability in the shortest path computation (see [GaJo79]).

R4: Implementation of the Fleury traversal strategy requires some attention due, largely, to the requirement to test the stated connectivity stipulation. An alternative that relaxes this complication was proposed by Edmonds and Johnson ([EdJo73]).

R5: Trivially, a necessary condition for a perfect matching to exist in a graph is that the graph possesses an even number of vertices. Since we may take G, in any interesting instance of UCPP, to be connected, there must be a path between every pair of vertices. Hence the complete graph specification is clear and since $|V_O|$ is even, it follows that the perfect matching step of the procedure is well-defined.

R6: Assuming a correct application of Algorithm 4.3.1, a postman would never traverse an edge more than twice in an optimal walk.

COMPUTATIONAL NOTE: If double-traversing occurred, the supergraph, $\overline{G}$, would have an edge from G duplicated more than once. But this would deny that $\overline{G}$ had been constructed correctly since two of the duplicated copies could be eliminated, leaving a connected graph with the same (even) degree parity everywhere and with smaller weight than $\overline{G}$.

4.3.3 Directed Postman Problems

The strategy for solving the DCPP is analogous to the one used for the undirected case. If the digraph is not symmetric, then a minimum-weight arc duplication produces a symmetric superdigraph. The number of copies of each arc is determined by solving a *circulation* problem.

FACTS

F10: Since easy to test, we may take G to be strongly connected.

F11: The multigraph $\overline{G}$ produced from a correct application of Algorithm 4.3.2 below is symmetric; obviously it remains strongly connected.

COMPUTATIONAL NOTE: The circulation problem in Algorithm 4.3.2 is easily solved by standard *network flow techniques* (Chapter 11).

```
Algorithm 4.3.2: Solution to DCPP

Input: Strongly connected digraph $G$ with arcs weighted by nonnegative integer
    values.
Output: Minimum-weight postman tour in $G$.

    If $G$ is symmetric
        Produce an eulerian tour in $G$.
    Else
        For each $k$, set $b_k = indegree(v_k) - outdegree(v_k)$.
        Solve the following circulation problem:

            minimize $\sum_{(v_i,v_j)\in E} w_{ij}x_{ij}$
            s.t. $\sum_{(v_k,v_j)\in E} x_{kj} - \sum_{(v_i,v_k)\in E} x_{ik} = b_k$ for $v_k \in V$
                 $x_{ij} \geq 0$

        For each $i, j$, add $x_{ij}$ copies of arc $(v_i, v_j)$ to $G$.
        Call the resulting multidigraph $\overline{G}$.
        Produce an eulerian tour in $\overline{G}$.
```

Producing an Eulerian Tour in a Symmetric (Multi)Digraph

Algorithm 4.3.2 above requires the traversal of an eulerian tour in the original digraph (if it is symmetric) or in a symmetric multidigraph. This can be accomplished by applying Algorithm 4.3.3 below (*cf.* [EhBr51]).

DEFINITION

D18: An ***intree*** is a connected, acyclic digraph where the outdegree of every vertex is at most 1.

```
Algorithm 4.3.3: Producing a Tour in an Eulerian Digraph

Input: Eulerian digraph $G$.
Output: Eulerian tour in $G$.

    Select any vertex in $G$ and denote it by $v^*$.
    Form an intree $T$ that spans $G$ and that is rooted at $v^*$.
    For each vertex $w$ in $G$, $w \neq v^*$,
        Label the out-arcs from $w$ randomly with consecutive integers
          subject to the restriction that the last (highest) label is given
          to the arc in intree $T$.
    Label the out-arcs from $v^*$ arbitrarily.
    Starting at vertex $v^*$, trace an eulerian tour in $G$ by always selecting
      the untraversed out-arc with the smallest label.
```

COMPUTATIONAL NOTE: An easy way to form an intree T of digraph G is to start with $T = \{v^*\}$ and proceed iteratively: select at each iteration an arc in G that is directed from a vertex in $V(G) - V(T)$ to a vertex in T; repeat until T spans G.

EXAMPLE

E2: Consider the instance digraph in the upper left of Figure 4.3.2. Specified next to each vertex is the respective value for b_k. Solving the explicit circulation problem defined in Algorithm 4.3.2 produces the following outcome: $x_{51} = x_{34} = 1$; $x_{45} = 2$; and $x_{ij} = 0$ elsewhere. Copies of the respective arcs are added forming the multigraph shown in the upper right of the figure. Applying Algorithm 4.3.3 and selecting (arbitrarily) vertex v_4 as a root, an intree T is constructed and shown at the bottom of Figure 4.3.2. The stated arc-labeling scheme is applied with labels affixed to the arcs in the multidigraph. Starting with vertex v_4 and proceeding in label order, produces an eulerian tour specified (unambiguously) by the following vertex sequence:

$$v_4, v_5, v_1, v_3, v_4, v_5, v_1, v_2, v_3, v_4, v_5, v_2, v_4$$

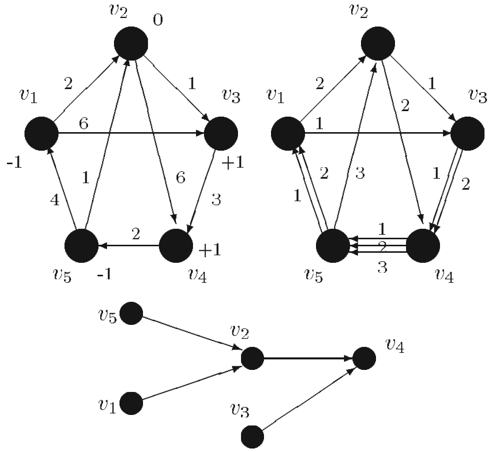


Figure 4.3.2 **Applications of Algorithms 4.3.2 and 4.3.3.**

REMARKS

R7: The network flow formulation in Algorithm 4.3.2 is due to Edmonds and Johnson ([EdJo73]).

R8: Trivially, a correct application of Algorithm 4.3.2 may require that an arc be duplicated several times.

R9: In applying Algorithm 4.3.3, the requirement that a tour be traced beginning with vertex v^* and proceeding in label order cannot be casually relaxed. For instance, if one starts from vertex v_3 on the labelled digraph in the upper right in Figure 4.3.2, any tour generated will violate the label ordering.

4.3.4 Mixed Postman Problems

FACTS

F12: The mixed postman problem, MCPP, is NP-hard; the reduction is from 3-SATISFIABILITY (*cf.* [Pa76]).

F13: MCPP remains NP-hard even on planar graphs with no vertex (total) degree exceeding 3 and with all edge weights the same (see [GaJo79].

Deciding if a Mixed Graph Is Eulerian

DEFINITIONS

D19: The ***total degree*** of a vertex v in a mixed graph G is the total number of arcs and undirected edges incident on v.

D20: A mixed graph is ***even*** if the total degree of each of its vertices is even.

D21: A vertex in a mixed graph is ***symmetric*** if its indegree and outdegree are equal.

TERMINOLOGY: A mixed graph is said to be ***symmetric*** if all of its vertices are symmetric.

D22: A mixed graph G satisfies the ***balance condition*** if for every $S \subseteq V(G)$, the difference between the number of arcs from S to $V(G)-S$ and the number of arcs from $V(G)-S$ to S is no greater than the number of undirected edges joining vertices in S and $V(G)-S$ (*cf.* [FoFu62]).

FACTS

F14: A (strongly) connected, mixed graph G is eulerian if and only if G is even and satisfies the balance condition.

F15: Mixed graphs that are even and symmetric are balanced.

EXAMPLE

E3: Clearly, the even-degree condition is necessary for a mixed graph to be eulerian while symmetry at each vertex is not. The graph in Figure 4.3.3 illustrates. An eulerian tour is specified by the vertex sequence $v_1, v_5, v_6, v_1, v_2, v_3, v_4, v_5, v_3, v_1$.

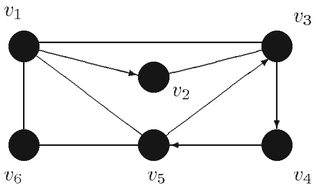


Figure 4.3.3 **An Eulerian Mixed Graph.**

COMPUTATIONAL NOTE: The (nontrivial) requirement in the mixed-graph case is to create a graph that satisfies the symmetry condition at each vertex or show that this is not possible. That is, we seek to orient some undirected edges in such a way that symmetry is created, albeit artificially. There is an easy network flow formulation that will do this or correctly conclude that no such orientation is possible.

Algorithm 4.3.4: Deciding if a Mixed Graph Is Eulerian

```
Input: an even and strongly connected mixed graph $G$.
Output: an orientation of some or all of the undirected edges of $G$ that is
        eulerian or a conclusion that no such orientation is possible.

    For each $k$, set $b_k = indegree(v_k) - outdegree(v_k)$.
    Replace each undirected edge in $G$ by a pair of oppositely oriented arcs.
    Let $U$ be the set of these new pairs of arcs.
    Solve the following network flow problem $P_s$:

        minimize $\sum_{(v_i,v_j)\in U} x_{ij}$
        s.t. $\sum_{\{(v_i,v_j)\in U | i=k\}} x_{ij} - \sum_{\{(v_j,v_i)\in U | i=k\}} x_{ji} = b_k$ for $v_k \in V$
             $0 \leq x_{ij} \leq 1$ for $(v_i, v_j) \in U$

    If $P_s$ has an admissible solution (i.e., $G$ is eulerian)
        For each undirected edge $\{v_i, v_j\}$
            If $x_{ij} = 1$
                Orient edge $\{v_i, v_j\}$ so that it is directed from $v_i$ to $v_j$.
            Else if $x_{ji} = 1$
                Orient edge $\{v_i, v_j\}$ so that it is directed from $v_j$ to $v_i$.
            Else
                Leave edge $\{v_i, v_j\}$ undirected.
    Else ($P_s$ has no admissible solution)
    Conclude that $G$ cannot be made eulerian through edge-orientation.
```

EXAMPLE

E4: The application of Algorithm 4.3.4 on the (mixed) graph in Figure 4.3.3 is illustrated in Figure 4.3.4. In the upper graph in the figure, the values b_k are written next to each vertex. The non-zero variables x_{23} and x_{31} induce the specified orientation for the original, undirected edges (v_2, v_3) and (v_1, v_3) as indicated by the lower graph in the figure. The existing eulerian tour can now be traced using Algorithm 4.3.3.

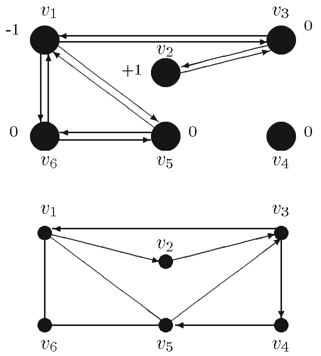


Figure 4.3.4 Application of Algorithm 4.3.4 on the graph in Figure 4.3.3.

REMARKS

R10: The second part of the balance condition in Definition 22 asks that for every subset of vertices, a lack of symmetry (a difference between total indegree and total outdegree of vertices in the subset) must be made up for by some or all of the undirected edges joining vertices in the subset to those outside.

R11: If the orientation produced by Algorithm 4.3.4 results in a graph with all edges directed, then an eulerian tour is produced by employing the strategy described previously in the case of eulerian digraphs (Algorithm 4.3.3). Alternatively, if Algorithm 4.3.4 outputs a graph with some undirected edges remaining, it is still possible to produce (with polynomial effort) a suitable tour, although care must be exercised, and even then, the ease of applicability of the procedures depends on whether the subgraph formed by the directed edges is connected and spanning (*cf.* [EdJo73]).

The Postman Problem for Mixed Graphs

Since MCPP is NP-hard in general, options are few. We may have to look for special cases that do submit to polynomial resolution, or we will simply have to be less ambitious and settle for *approximation algorithms*, i.e., fast procedures that cannot guarantee optimal solutions but that will produce ones that are, in some well-defined sense, reasonably close to optimal. Of course, for instances of manageable size, it might be feasible to resort to exact procedures. However, these approaches are inherently enumerative and will require effort that is exponential in the worst case (*cf.* [EiGeLa95-a], [EiGeLa95-b]).

DEFINITION

D23: An algorithm is an ***approximation algorithm*** for a given problem if given any instance of the problem, it finds at least a ***candidate*** solution for the instance.

REMARK

R12: If the instance for MCPP is at least even but perhaps not symmetric, we can apply Algorithm 4.3.4 in order to test if symmetry at each vertex can be created. If so, the instance is eulerian (by Facts 14 and 15), and we can proceed accordingly. Otherwise, it is not eulerian, and we have to determine if it can be made so through some duplication of edges and/or arcs.

FACT

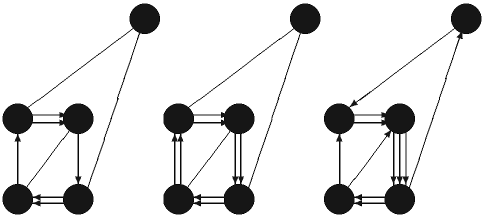
F16: A mixed graph G has a postman tour if and only if G is strongly connected.

COMPUTATIONAL NOTE: Testing for strong connectivity in mixed graphs is polynomial since undirected edges could be replaced by pairs of oppositely directed ones and then the (polynomial) algorithm for digraphs applies.

REFERENCE NOTE: The graphs that are employed in the remaining figures are either explicitly drawn from or are alluded to in an important paper by Frederickson ([Fr79]).

EXAMPLE

E5: To illustrate the problematical aspect unique to the mixed postman problem, consider the mixed graph in Figure 4.3.5, part a; all edges are assumed to have weight 1. It is easy to see that no orientation exists for undirected edges that would create symmetry. Now, duplication of two arcs creates symmetry as shown in part b of the figure; however, the resulting structure is not even so further duplication is required. On the other hand, the multigraph in part c also has only two arcs duplicated but is both symmetric *and* even; clearly, this graph is preferred. Unfortunately, it is not easy to distinguish, in any general way, its selection over the structure of part b.



(a) (b) (c)

Figure 4.3.5 Interaction between symmetry and even-degree.

COMPUTATIONAL NOTE: It is possible to deal with a few of these complications, albeit in somewhat ad hoc fashion, by employing various network flow formulations; relevant results are discussed in [EdJo73]. Important is to note that the prime contributor to the intractability of the general, mixed postman case is the ambiguity in effectively dealing with the interaction between symmetry and even-degree creation and/or preservation.

Approximation Algorithm ES

The following approximation algorithm combines an easy even-degree-creation phase followed by a more intricate, joint symmetry-producing/even-degree-preserving phase. The details are somewhat involved (*cf.* [EdJo73] and [Fr79]), and so in the statement of Algorithm 4.3.5, the step is simply referenced as "symmetric/even-parity."

Algorithm 4.3.5: Approximation Algorithm ES

Input: Strongly connected, mixed graph G with edges/arcs weighted by nonnegative integer values.
Output: Admissible postman tour.

```
    Apply the even-degree-creation component of UCPP to the underlying graph
     of mixed graph $G$.
    Restore orientation to edges as specified in $G$.
    Let $\overline{G}$ be the resulting supergraph.
    Operating on $\overline{G}$, apply the symmetric/even-parity construction.
    Let $\overline{\overline{G}}$ be the resulting graph.
    Produce an eulerian tour in $\overline{\overline{G}}$.
```

TERMINOLOGY: The approximation procedure stated by Algorithm 4.3.5 is sometimes referred to as the ***even-symmetric*** strategy, i.e., ***ES***.

REMARK

R13: Since it cannot guarantee an optimal solution, it is interesting to consider the limit (if any) to *how* poorly Algorithm ES could perform. In fact, this was answered by Frederickson.

FACT

F17: [Fr79] The ratio of the value of a postman solution produced by Algorithm ES to an optimal value cannot exceed 2. Importantly, the value of 2 is approachable as established by Example 6 below.

EXAMPLE

E6: Consider the mixed graph in Figure 4.3.6, part a, where edge weights are specified on the graph. The even-degree-creation phase of Algorithm ES duplicates the directed edges (considered undirected for the stated step) yielding the multigraph in part b. Operating on this graph to produce symmetry while preserving the even degree condition yields the structure in part c having total edge weight $4 + 12\epsilon$. The eulerian tour in this multigraph is not optimal however. Had one been less greedy in the even-degree-creation application, duplicating instead the edges with weight 2ϵ, the structure in part d of the figure would have resulted, yielding an optimal multigraph and hence, a correct tour directly. Its weight is $2 + 10\epsilon$.

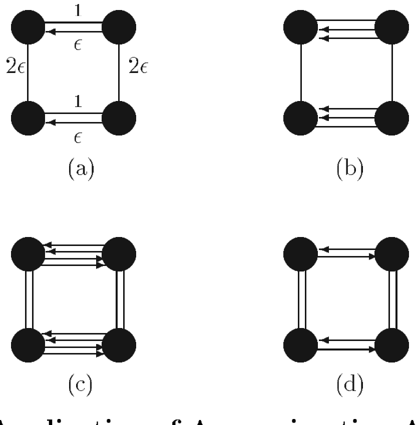


Figure 4.3.6 **Application of Approximation Algorithm ES.**

Approximate Algorithm SE

A natural alternative approximation is to reverse the strategy proposed by Algorithm ES, yielding the following, ***symmetric-even*** approach (***SE***) (*cf.* [Fr79]).

COMPUTATIONAL NOTE: The symmetry-creation construction of step 1 in Algorithm 4.3.6 is polynomial (*cf.* [Fr79]) and employs the same symmetry-creation component of the symmetry/even-parity step in Algorithm 4.3.5.

Algorithm 4.3.6: Approximation Algorithm SE

```
Input: Strongly connected, mixed graph $G$ with edges/arcs weighted by
       nonnegative integer values.
Output: Admissible postman tour.
    Create symmetry on mixed graph $G$, and denote resulting mixed graph $G^s$.
    Let $H$ be the subgraph induced on the undirected edges of $G^s$.
    Apply the even-degree-creation component of UCPP to $H$.
    Let $\overline{G}$ be the resulting even-degree, symmetric super(multi)graph of $G^s$.
    Produce an eulerian tour in $\overline{G}$.
```

EXAMPLE

E7: Consider the graph in Figure 4.3.7, part a. A correct application of Algorithm SE produces the multigraph in part b. However, the structure in part c is optimal.

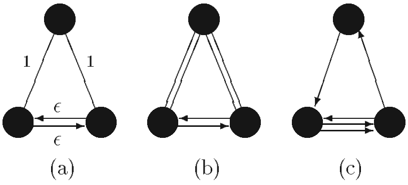


Figure 4.3.7 **Application of Approximation Algorithm SE.**

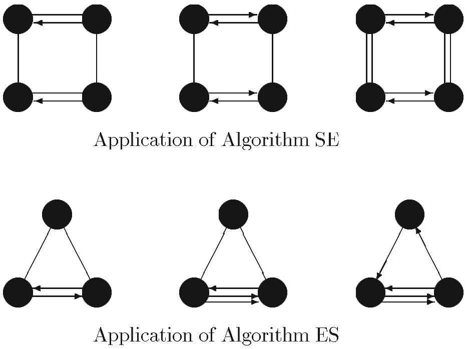
Some Performance Bounds

FACT

F18: [Fr79] The ratio of the length of an outcome from Algorithm SE to an optimal tour value will also never exceed 2. Example 7 provides evidence that this value is approachable as well.

EXAMPLE

E8: Proceeding from left to right in Figure 4.3.8 on the respective input instances, it is evident that Algorithm SE solves a worst-case instance for Algorithm ES, while the latter accomplishes the same outcome on a worst-case instance for Algorithm SE.


Application of Algorithm SE

Application of Algorithm ES

Figure 4.3.8 **Algorithms ES and SE on each other's worst-case instances.**

R14: In the worst-case sense, both approximation procedures Algorithm 4.3.5 (ES) and Algorithm 4.3.6 (SE) perform the same. However, Example 8 demonstrates an interesting phenomenon: if each approximation is applied to a worst-case instance of the other, the outcome is that Algorithm ES solves (to optimality) the worst-case instance for Algorithm SE, while the latter, when operating on the worst-case instance for Algorithm ES, produces the optimal outcome.

R15: The outcome of Example 8 motivates an obvious question which is stated loosely as follows: What if Algorithms ES and SE realize their respective worst-case behaviors on different classes of graphs? If this is the case, it is conceivable that they could be employed in a "composite" fashion where each strategy is applied separately and the best outcome is then selected. While certainly demanding additional work, the polynomiality of total effort required by the approach is unaffected. Preserving polynomiality is meaningful because if this composite strategy is applied, the outcome does indeed yield an improvement in guaranteed performance. The first result of this sort is also due to Frederickson ([Fr79]) where it was shown that applying the two stated heuristics and selecting the best result would never produce a tour having length that when compared to an optimal value yielded a ratio in excess of $\frac{5}{3}$.

R16: In the same paper ([Fr79]), Frederickson also proposed a separate composite strategy for planar instances. The bound on its performance was shown to be $\frac{3}{2}$.

R17: When the $\frac{5}{3}$ result in [Fr79] appeared, attempts to create an instance establishing realizability were not fruitful; the closest was a $\frac{3}{2}$-inducing instance shown in Figure 4.3.9 below. But a 1999 result, which employed a modification of the stated Frederickson approximation, was proposed in Raghavachari and Veerasamy (RaVe99]) with a performance ratio bounded by $\frac{3}{2}$. The instance in Figure 4.3.9 establishes tightness.

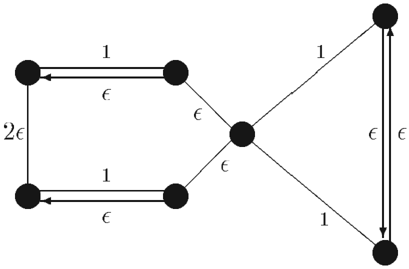


Figure 4.3.9 **A worst case for composite use of Algorithms ES and SE.**

R18: Following its attendant proof of intractability, MCPP is not likely to submit to any fast solution. However, as with any other provably NP-hard problem, this attribute does not preclude the existence of nor the value in pursuing special cases which might prove to be quickly solvable. This is certainly the case for MCPP.

COMPUTATIONAL NOTE: If input instances are confined to the class of *recursively structured graphs*, then it is possible to solve MCPP on members of this class and, in fact, by strategies requiring only linear-time effort (*cf.* [BoPaTo91], [BoPaTo92], and §10.4). Typical recursive graph classes include *trees*, *series-parallel graphs*, *Halin graphs*, *partial k-trees*, and *treewidth-k graphs*.

References

[BeBo74] E. L. Beltrami and L. D. Bodin, Network and Vehicle Routing for Municipal Waste Collection, *Networks*, 4 (1974), 65–94.

[BoPaTo91] R. B. Borie, R. Parker, and C. Tovey, Deterministic Decomposition of Recursive Graph Classes, *SIAM Journal of Discrete Mathematics*, 4 (1991), pp. 481–501.

[BoPaTo92] R. B. Borie, R. Parker and C. Tovey, Automatic Generation of Linear-Time Algorithms from Predicate Calculus Descriptions of Problems on Recursively Constructible Graph Families, *Algorithmica*, 7 (1992), pp. 555–581.

[Br80] P. Brucker, The Chinese Postman Problem for Mixed Networks, in *Proceedings of the International Workshop on Graphtheoretic Concepts in Computer Science, Lecture Notes in Computer Science 100*, Springer-Verlag, New York (1980), pp. 354–366.

[ChBeCaCoMo84] C. Christofides, E. Benavent, V. Campos, A. Corberan, and E. Mota, An Optimal Method for the Mixed Postman Problem, in *System Modeling and Optimization, Lecture Notes in Control and Information Science 59*, Springer, New York (1984).

[Ed65-a] J. Edmonds, The Chinese Postman Problem, *Operations Research*, 13, Suppl. 1 (1965), pp. 373.

[Ed65-b] J. Edmonds,Maximum Matching and a Polyhedron with 0,1 Vertices, *J. Research National Bureau of Standards*, 69B (1965), pp. 125–130.

[Ed65-c] J. Edmonds, Paths, Trees, and Flowers, *Canadian J. Mathematics*, 17 (1965), pp. 449–467.

[EdJo73] J. Edmonds and E. Johnson, Matching, Euler Tours, and the Chinese Postman, *Mathematical Programming*, 5 (1973), pp. 88–124.

[EhBr51] T. van Aardenne-Ehrenfest and N. G. de Bruin, Circuits and Trees in Oriented Linear Graphs, *Simon Stevin*, 28 (1951), pp. 203–217.

[EiGeLa95-a] A. Eiselt, M. Gendreau, and G. Laporte, Arc Routing Problems, Part I: The Chinese Postman Problem, *Operations Research*, 43 (1995), pp. 231–242.

[EiGeLa95-b] A. Eiselt, M. Gendreau, and G. Laporte, Arc Routing Problems, Part II: The Rural Postman Problem, *Operations Research*, 43 (1995), pp. 399–414.

[Fl91] H. Fleischner, Eulerian Graphs and Related Topics, Part 1, Vol. 2, *Ann. Discrete Math*, 50, North-Holland, Amsterdam, (1991).

[FoFu62] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, NJ (1962).

[Fr79] G. Frederickson, Approximation Algorithms for Some Postman Problems, *J. ACM*, 26 (1979), pp. 538–554.

[GaJo79] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., New York (1979).

[Gu62] M. Guan, Graphic Programming Using Even and Odd Points, *Chinese Mathematics*, 1 (1962), pp. 273–277.

[Gu84-a] M. Guan, A Survey of the Chinese Postman Problem, *J. Math. Res. and Expos.*, 4 (1984), pp. 113–119 (in Chinese).

[Gu84-b] M. Guan, On the Windy Postman Problem, *Discrete Applied Mathematics*, 9 (1984), pp. 41–46.

[KaKo79] C. Kappauf and G. Koehler, The Mixed Postman Problem, *Discrete Applied Mathematics*, 1 (1979), pp. 89–103.

[Ka67] A. Kaufmann, *Graphs, Dynamic Programming and Finite Games*, Academic Press, New York (1967).

[LiZh88] Y. Lin and Y. Zhao, A New Algorithm for the Directed Chinese Postman Problem, *Computers and Operations Research*, 15 (1988), pp. 577–584.

[Mi79] E. Mineka, The Chinese Postman Problem for Mixed Networks, *Management Science*, 25 (1979), pp. 643–648.

[NoPi96] Y. Norbert and J. C. Picard, An Optimal Algorithm for the Mixed Chinese Postman Problem, *Networks*, 27 (1996), pp. 95–108.

[Or74] C. S. Orloff, A Fundamental Problem in Vehicle Routing, *Networks*, 4 (1974), pp. 35–64.

[Pa76] C. H. Papadimitriou, On the Complexity of Edge Traversing, *J. ACM*, 23 (1976), pp. 544–554.

[Pe94] W. L. Pearn, Solvable Cases of the k-Person Chinese Postman Problem on Mixed Networks, *Operations Research Letters*, 16 (1994), pp. 241–244.

[PeLu95] W. L. Pearn and C. M. Lui, Algorithms for the Chinese Postman Problem on Mixed Networks, *Computers and Operations Research*, 22 (1995), pp. 479–489.

[RaVe99] B. Raghavachari and J. Veerasamy, A $\frac{3}{2}$-Approximation Algorithm for the Mixed Postman Problem, *SIAM J. Discrete Mathematics*, 12 (1999), pp. 425–433.

[Ra93] T. K. Ralphs, On the Mixed Chinese Postman Problem, *Operations Research Letters*, 14 (1993), pp. 123–127.

[Wi89] Z. Win, On the Windy Postman Problem on Eulerian Graphs, *Mathematical Programming*, 44 (1989), pp. 97–112.

4.4 DEBRUIJN GRAPHS AND SEQUENCES

A. K. Dewdney, University of Waterloo, Canada

4.4.1 DeBruijn Graph Basics
4.4.2 Generating deBruijn Sequences
4.4.3 Pseudorandom Numbers
4.4.4 A Genetics Application
References

Introduction

N. deBruijn solved the problem of finding a minimum-length binary string that contains as a (contiguous) substring every binary string of a prescribed length k. For this purpose, he prescribed a special directed graph, of in-degree 2 and out-degree 2, now called a deBruijn graph. In this section, we cover the basics of deBruijn graphs, two methods to generate deBruijn sequences, and applications to the generation of pseudorandom numbers and to genetics.

4.4.1 DeBruijn Graph Basics

DeBruijn Sequences

DEFINITIONS

D1: A ***deBruijn sequence of order*** k is a binary string of length $n = 2^k$ in which

- the last bit is considered to be adjacent to the first, and
- every possible binary k-tuple appears exactly once.

Two deBruijn sequences are considered to be the "same sequence" if one can be obtained from the other by a cyclic permutation.

D2: In a string s of length $m > k$, the ***successor of a substring*** t of length k is the k-bit substring t' that begins at the second bit of t. This is understood cyclically within s, so that if needed, the last bit of the successor substring t' is the first bit of string s.

D3: In a string s of length $m > k$, the ***k-tour*** is the sequence of substrings of length k, starting with the initial substring. Since this is understood cyclically within s, there are m substrings in the tour.

D4: A k-bit string b is said to be obtained from a k-bit string $a = a_1a_2a_3\cdots a_k$ by a ***(left) shift operation*** if $b_i = a_{i+1}$, for $i = 1, 2, \ldots, k-1$. The bit b_k may be arbitrary.

D5: A left shift $a_1a_2\ldots a_k \longrightarrow b_1b_2\ldots b_k$ is a ***cycle shift*** if $b_k = a_1$.

D6: A left shift $a_1a_2\ldots a_k \longrightarrow b_1b_2\ldots b_k$ is a ***deBruijn shift*** if $b_k \neq a_1$.

FACTS

F1: An obvious lower bound on the length of a deBruijn sequence of order k is 2^k, since there are 2^k different bitstrings of length k, and since each bit in a sequence starts only one k-bitstring.

F2: The successor of each k-bit substring t in a deBruijn sequence is either a cycle shift or a deBruijn shift of t.

EXAMPLES

E1: 00010111 is a deBruijn sequence of order 3. Its 3-tour is

$$000, 001, 010, 101, 011, 111, 110, 100$$

E2: 0000101101001111 is a deBruijn sequence of order 4.

DeBruijn Graphs

An intuitive approach to the problem of constructing a deBruijn sequence is to construct a graph in which a hamiltonian tour corresponds to such a sequence.

DEFINITIONS

D7: A ***deBruijn graph of order*** k, denoted by $G(k)$, is a directed graph with 2^k vertices, each labeled with a unique k-bit string. Vertex a is joined to vertex b by an arc if bitstring b is obtainable from bitstring a by either a cycle shift or a deBruijn shift. Additionally, each arc of $G(k)$ is designated as a ***cycle-shift arc*** or a ***deBruijn arc***, according to the shift operation it represents. Each arc is labeled by the first bit of the vertex at which it originates, followed by the label of the vertex at which it terminates.

D8: The ***cycle-shift 2-factor*** in a deBruijn graph is the 2-factor formed by all of its cycle-shift arcs.

D9: The ***deBruijn 2-factor*** in a deBruijn graph is the 2-factor formed by all of its deBruijn.

EXAMPLE

E3: Figure 4.4.1 below illustrates the deBruijn graph of order 3.

FACTS

F3: The cycle-shift arcs form a directed 2-factor, because the cycle-shift operation acts as a permutation on the bitstrings. Similarly, the deBruijn arcs form a directed 2-factor.

F4: Every vertex of a deBruijn graph has out-degree 2. The first bit of the label on one of the vertices to which it points is 0, and the first bit on the label of the other is 1.

F5: Every vertex of a deBruijn graph has in-degree 2.

F6: Every deBruijn graph is strongly connected.

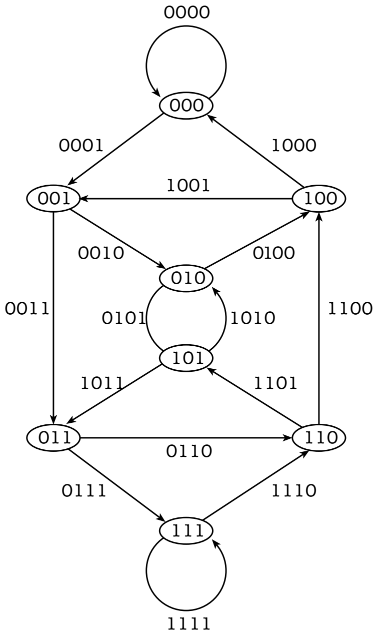


Figure 4.4.1 **A deBruijn graph of order 3.**

F7: Every deBruijn graph is hamiltonian.

F8: The hamiltonian (directed) circuits in the deBruijn graph $G(k)$ are in one-to-one correspondence with the deBruijn sequences of order k. The correspondence is realized by listing, in sequence, the first bit of each vertex encountered on a hamiltonian tour.

F9: **deBruijn's Theorem** [dB47] For each positive integer k there are $2^{2^{k-1}-k}$ deBruijn sequences of order k.

k	1	2	3	4	5	6	...
$2^{2^{k-1}-k}$	1	1	2	16	2048	67108864	...

REMARKS

R1: A hamiltonian circuit in a deBruijn graph can be constructed by splicing together the components of its deBruijn 2-factors. However, deBruijn's theorem depends on a more elegant way to construct deBruijn sequences.

R2: Since each component of the deBruijn 2-factor of the deBruijn graph $G(k)$ has cardinality at most k, it follows that the number of components of the deBruijn 2-factor grows exponentially in k.

4.4.2 Generating deBruijn Sequences

An efficient algorithm for constructing a deBruijn sequence of order k is based not on finding a hamiltonian circuit in the deBruijn graph of order k, but rather on the easier task of constructing an Eulerian tour in the deBruijn graph of order $k-1$. Another interesting method is strictly lexicographic.

FACTS

F10: [Go46]: A strongly connected directed graph in which every vertex has the same indegree as outdegree has an Eulerian tour.

F11: In a deBruijn graph $G(k)$, the k-sequence of arc labels encountered on every directed path of length k originating at a vertex v is the binary string that labels vertex v. (This is an immediate consequence of the specification of the arc labels in the definition of a deBruijn graph.)

F12: The sequence of arc labels encountered on an Eulerian tour of the deBruijn graph of order k is a deBruijn sequence of order $k + 1$.

EXAMPLE

E4: Figure 4.4.2 illustrates the construction of a deBruijn sequence of order 4 from the deBruijn graph of order 3.

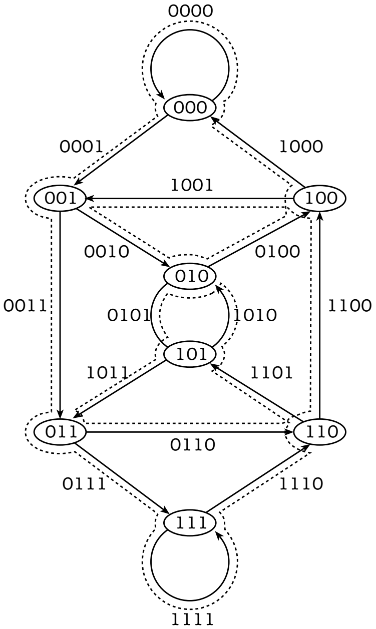


Figure 4.4.2 **An Eulerian circuit in $G(3)$.**

REMARKS

R3: The proof of Fact F12 is not difficult. Since an Eulerian tour of $G(k)$ visits each vertex twice, it follows from Fact F11 that each bitstring label occurs twice in the sequence of arc labels. By Fact F4, one occurrence is followed by a '0' and the other by a '1'.

ALGORITHM

A1: To construct a deBruijn sequence of order k, use Fleury's algorithm (quadratic time) to construct an Eulerian tour of the deBruijn graph $G(k - 1)$. Then record the sequence of arc labels on the Eulerian tour. (Fleury's algorithm appears in §4.2.)

Necklaces and Lyndon Words

Fredricksen and Kessler [FrKe77] have published a remarkable alternative method for constructing deBruijn sequences.

DEFINITIONS

D10: A ***rotation of a binary string*** is the result of an iteration of cycle shifts. (Rotation is clearly an equivalence relation.)

D11: An equivalence class under rotation of the binary strings of length n is called a ***necklace*** of order n.

D12: A ***Lyndon word*** of order n is a necklace of order n whose rotation class has n binary strings. A Lyndon necklace of length 1 is called ***trivial***. We take the lexicographically least element of the equivalence class as representative of the necklace.

FACTS

F13: A necklace representative is a Lyndon word if and only if it is aperiodic, i.e., cannot be written as the concatenation of two or more identical strings.

F14: By an elementary application of Burnside-Polya enumeration, the number of necklaces of order n is

$$\frac{1}{n}\sum_{k:k|n}\phi(k)\cdot 2^{\frac{n}{k}}$$

where $\phi(n)$ is the number of integers in the interval $[1, n]$ that are relatively prime to n.

F15: [FrKe77]: If the (lexicographically least) represenatives of all the nontrivial Lyndon words whose lengths divide n are arranged into lexicographic order and concatenated, with the terminal string 10 appended at the end, then the result is a deBruijn sequence of order n that is lexicographically minimum.

REMARK

R4: The number $N(n)$ of necklaces grows exponentially with n. While $N(5) = 8$, we have $N(10) = 108$, and $N(15) = 2192$.

EXAMPLES

E5: Figure 4.4.3 displays five equivalent strings of length 5.

01101

11010

10101

01011 ←

10110

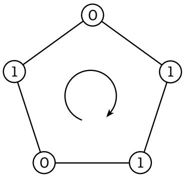


Figure 4.4.3 **A necklace and its representations.**

E6: The only nontrivial Lyndon word of length 2 is 01. We observe that 0110 is a deBruijn sequence of order 2.

E7: The only nontrivial Lyndon words of length 3 are 001 and 011. We observe that 00101110 is a deBruijn sequence of order 3.

E8: We now illustrate Fact F15 for $n = 4$. In lexicographic order, the nontrivial Lyndon words of lengths that divide 4 are

$$0001, 0011, 01, 0111$$

If we now concatenate these words in the order given, we obtain the lexicographically least deBruijn sequence

$$0001001101011110$$

of order 4.

4.4.3 Pseudorandom Numbers

For Monte Carlo applications, the numbers produced by ordinary pseudorandom number generators (e.g., congruential generators) are close enough to random not to affect the outcome of the study. But for certain applications called precision Monte Carlo simulation, special sequences must be used. DeBruijn sequences, which already appear somewhat random to the untrained eye, may be made more random by interchanging runs of zeros and ones.

DEFINITIONS

D13: A ***run*** in a binary sequence is a subsequence of identical bits, and a ***maximal run*** is a run that is not contained in any longer run.

D14: [Go67] The ***Golomb postulates of randomness*** for a periodic sequence X are as follows:

- The number of 1's in X differs from the number of 0's by at most unity.
- At least half the runs in X have length 1, at least one-quarter have length 2, at least one-eighth have length 3, etc.
- The bit in position i is correlated to the same degree with adjacent bits ($i + 1$ and $i - 1$) as it is with ones further away ($i + 100$ and $i - 100$).

D15: A ***run-permuted sequence*** is a pseudorandom sequence obtained by the following procedure:

0. Generate a random deBruijn sequence of order n.
1. Randomly permute the maximal runs of 0's.
2. Randomly permute the maximal runs of 1's.

D16: [Ch87] The ***randomness of an infinite binary sequence*** S is defined to be

$$\lim_{m\to\infty} \frac{s(m)}{m}$$

where $s(m)$ is the minimum number of states in a 2-symbol Turing machine that produces the first m bits of the sequence S.

FACTS

F16: Obviously, every deBruijn sequence can be generated as a run-permuted sequence.

F17: Any deBruijn sequence X of order n satisfies Golomb's first two postulates. First, the number of 1's exactly equals the number of 0's. Second, it is easily shown that over all possible binary subsequences of each length n, exactly half of the runs have that length.

F18: Interchanging (maximal) runs in permuting a deBruijn sequence does not change the number of runs of any length or kind. One therefore obtains a much larger class of sequences that are, by Golombs measure, just as random as the original deBruijn sequence from which the new sequences are generated.

F19: [Je91]: The class C_n of run-permuted sequences of order n contains a vanishingly small proportion of deBruijn sequences of order n as n increases.

4.4.4 A Genetics Application

Typically, the short DNA fragments observed in experiments are not sufficient to reconstruct the genome of an organism completely. Because of the time and expense of such experiments, it is desirable to minimize the remaining work. To this end, biologists algorithmically assemble as much of the genome as they can, thereby obtaining longer DNA fragments that are fewer in number. They then perform additional experiments at specific locales in the resulting sequences, in order to extend the reconstruction.

A phenomenon that complicates the stepwise reconstruction of the genome is the natural occurrence of multiple copies of the same substring in a number of DNA sequences acquired by experiment. To help resolve this difficulty, Pevzner, Tang and Waterman [PvTaWa01] have applied modified de Bruijn graphs, in which a repeated k-string in a given sequence s_i results in multiple vertices, and consequently, in multiple paths connecting certain pairs of vertices in the graph. Such a graph need not be connected.

DEFINITIONS

D17: A ***DNA sequence*** is any finite sequence of the letters A, C, G, T.

D18: For any set $S = \{s_1, s_2, \ldots, s_n\}$ of DNA sequences, we define the ***S-relative deBruijn graph of order*** k to have vertices corresponding to all k-substrings from the elements of S, one for each occurrence of a substring. Two such vertices u and v are adjacent if their substrings belong to the same DNA sequence s_i and the last $k-1$ letters of u coincide with the first $k-1$ letters of v.

REMARK

R5: Since Eulerian paths can be found very quickly in connected portions of the S-relative deBruijn graph, partial paths can be produced efficiently for the graph as a whole. These not only (in most cases) recapture the original sequences, but suggest where additional experiments need to be performed to choose between different possible paths through the S-relative deBruijn graph.

References

[COS] Information regarding necklaces, unlabeled necklaces, Lyndon words, deBruijn sequences. Available at www.theory.csc.uvic.ca/ cos/inf/neck/Necklaceinfo/html.

[Ch87] G. J. Chaitin, *Algorithmic Information Theory* Cambridge University Press, 1987.

[De46] N. G. deBruijn, A combinatorial problem, *Nederl. Akad. Wetensch., Proc.* 49 (1946), 758–764.

[FrKe77] H. Fredricksen and I. Kessler, Lexicographic compositions and deBruijn sequences, *J. Combin. Theory, Ser. A* 22 (1977), 17–30.

[Je01] C. J. A. Jensen, On the construction of run permuted sequences, pp. 196–203 in *Advances in Cryptology - Eurocrypt '90* (edited by L. B. Damgard), Springer-Verlag, 2001.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co, 1979.

[Go67] S. W. Golomb, *Shift Register Sequences*, Holden-Day, 1967.

[Go46] I. J. Good, Normal recurring decimals, *J. London Math. Soc.* 21 (1946), 167–172.

[GrGeMiLe99] A. J. F. Griffiths, W. M. Gelbart, J. H. Miller, and R. C. Lewontin, *Modern Genetic Analysis*, W. H. Freeman, 1999.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Ha67] M. Hall Jr., *Combinatorial Theory*, Blaisdell Publishing Co, 1967.

[MuPv02] Z. Mulyukov and P. A. Pvezner, Euler-PCR: Finishing experiments for repeat resolution, *Pacific Symposium on Biocomputing* 7 (2002), 199–210.

[PvTaWa01] P. A. Pvezner, H. Tang, and M. S. Waterman, An Eulerian path approach to DNA fragment assembly, *Proc. Natl. Acad. Sci.* 98 (2001).

4.5 HAMILTONIAN GRAPHS

Ronald J. Gould, Emory University

4.5.1 History
4.5.2 The Classic Attacks
4.5.3 Extending the Classics
4.5.4 More Than One Hamiltonian Cycle?
4.5.5 Random Graphs
4.5.6 Forbidden Subgraphs
References

4.5.1 History

Characterizing hamiltonian graphs is an NP-complete problem (see [GaJo79]), thus the hamiltonian problem is generally considered to be determining conditions under which a graph contains a hamiltonian cycle. Named for Sir William Rowan Hamilton, this problem traces its origins to the 1850s. Hamilton exhibited his *Icosian Game* at a meeting in Dublin in 1857. The game involved finding various paths and cycles, including spanning cycles, of the regular dodecahedron. The game was marketed by a wholesale dealer in 1859, but apparently was not a big hit. Perhaps the only profit was Hamilton's, as he sold the game to the dealer for 25 pounds.

Hamilton does not appear to be the first to have considered the question of spanning cycles. In a paper [Ki56] submitted in 1855, Thomas Penyngton Kirkman posed the question: Given the graph of a polyhedron, can one always find a circuit (cycle) that passes through each vertex once and only once. Thus, Kirkman actually asked a more general question than Hamilton. Unfortunately for Kirkman, the term hamiltonian cycle is much too ingrained to be changed now. For a more detailed account of this history see [BiLlWi86].

DEFINITIONS

D1: A graph G is ***hamiltonian*** if it contains a spanning cycle (***hamiltonian cycle***).

D2: A graph G is ***traceable*** if it contains a spanning path.

D3: A graph G is ***hamiltonian connected*** if any pair of vertices are the ends of a spanning path.

4.5.2 The Classic Attacks

There are certain fundamental results that deserve attention, both for their contribution to the overall theory and for their effect on the later development of the area.

The approach taken to developing sufficient conditions for a graph to be hamiltonian usually involved some sort of edge density condition; providing enough edges to ensure the existence of a hamiltonian cycle.

TERMINOLOGY: The ***order*** of a graph is the cardinality of its vertex set, and the ***size*** of a graph is the cardinality of its edge set.

Degrees

NOTATION: The mimimum degree of the vertices of a graph G is denoted $\delta_{min}(G)$, and the maximum degree is denoted $\delta_{max}(G)$.

DEFINITIONS

D4: We say a set $X \subseteq V(G)$ is ***independent*** if there are no edges between vertices in X. The cardinality of a largest independent set in G is called the ***independence number*** of G and is denoted $ind(G)$.

D5: The ***k-degree closure*** of G, denoted $C_k(G)$, is the graph obtained by recursively joining pairs of non-adjacent vertices whose degree sum is at least k, until no such pair remains.

D6: For a *balanced* bipartite graph $G = (X \cup Y, E)$ (i.e., $|X| = |Y|$), the ***bipartite degree closure*** is that graph obtained by joining any non-adjacent pair $x \in X$ and $y \in Y$ whose degree sum is at least $n + 1$.

NOTATION: The following notation has become standard in the area:

$$\sigma_k(G) = min\{\sum_i^k deg\ x_i\ \ |x_1, \ldots, x_k \text{ are independent}\}$$

FACTS

F1: [Di52] If G is a graph of order n such that $\delta_{min}(G) \geq n/2$, then G is hamiltonian.

F2: Let G be a graph of order n.

i. [Or60] If $\sigma_2(G) \geq n$, then G is hamiltonian, and if $\sigma_2(G) \geq n-1$, then G is traceable.

ii. [Or63] If $\sigma_2(G) \geq n + 1$, then G is hamiltonian connected.

EXAMPLE

E1: Consider two $K_{(p+1)/2}$ with one vertex from each identified (graph on left in Figure 4.5.1). This graph is not hamiltonian, but has order p, $\delta_{min}(G) = (p-1)/2$, and $\sigma_2(G) = p - 1$, illustrating the sharpness of Dirac's Theorem and Ore's Theorem (Fact 2 i.). The graph obtained by identifying a pair of vertices from two copies of $K_{(p+2)/2}$ is not hamiltonian connected, has $\delta_{min}(G) = p/2$ and $\sigma_2(G) = p$, showing Ore ii is sharp (graph on right in Figure 4.5.1).

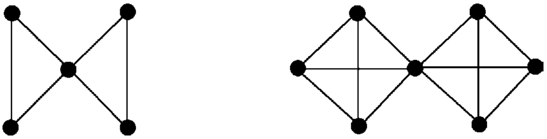


Figure 4.5.1 **Illustrating the sharpness of Dirac's and Ore's results.**

F3: [Ja80] Let G be a d-regular 2-connected graph of order n with $d \geq n/3$, then G is hamiltonian.

F4: [MoMo63] If $G = (X \cup Y, E)$ is a balanced bipartite graph of order $2n$ ($n \geq 2$) with $deg(u) + deg(v) \geq n + 1$ for each non-adjacent pair $u \in X$ and $v \in Y$, then G is hamiltonian.

F5: [BoCh76] Let G have order n. Then

i. $C_k(G)$ is well defined,

ii. G is hamiltonian if and only if $C_n(G)$ is hamiltonian,

iii. if $C_{n+1}(G)$ is a complete graph, then G is hamiltonian connected.

F6: [He91] A balanced bipartite graph is hamiltonian if and only if its bipartite closure is hamiltonian.

REMARK

R1: These closure results provide an interesting relaxation of the degree conditions. The closure is (hopefully) a denser graph, making it easier to find a hamiltonian cycle. However, the number of edges actually added in forming these closures can vary widely. It is easy to construct examples for all possible values from 0 to the total number of missing edges. Thus, we might receive no help in determining whether the original graph is hamiltonian, or we might conclude trivially that it is (when the closure is the complete graph).

Other Counts

DEFINITION

D7: The ***neighborhood*** of a vertex x in a graph G, denoted $N_G(x)$, is the set of all vertices adjacent to x in G. Similarly, $N_G(S)$ denotes the **neighborhood of the set** S and is the collection of all vertices adjacent to some vertex in S.

NOTATION: When the graph in which the neighborhood is defined is clear, the subscript is omitted.

D8: The ***(vertex-)connectivity*** of a connected graph G, denoted $\kappa_v(G)$, is the minimum number of vertices whose removal can either disconnect G or reduce it to a 1-vertex graph.

D9: A graph G is ***k-connected*** if $\kappa_v(G) \geq k$.

EXAMPLE

E2: The graph $G(r, p)$ is that graph with order p and vertex set $S \cup T \cup U$ where $|S| = |T| = r$ and $|U| = p - 2r$ and where two vertices are adjacent if either belongs to S or both belong to U. Hence, the subgraphs induced on S, T, and U are K_r, $\bar{K}_r$ and K_{p-2r}, respectively. Figure 4.5.2 below shows the graphs $G(1, 6)$ and $G(2, 5)$.

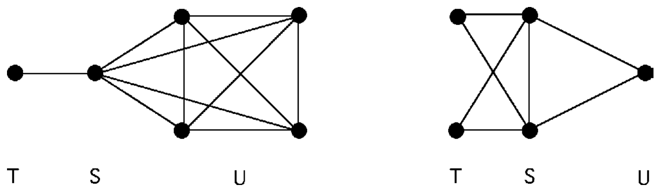


Figure 4.5.2 **Graphs $G(1,6)$ and $G(2,5)$.**

FACTS

F7: [Or63] If G is a graph of order n and size greater than $\binom{n-1}{2}+1$, then G is hamiltonian. Furthermore, the only non-hamiltonian graphs with size exactly $\binom{n-1}{2}+1$ are $G(1,n)$ and $G(2,5)$. In addition, if G has size at least $\binom{n-1}{3}$, then G is hamiltonian connected.

F8: [Fa84] If G is a 2-connected graph of order n such that

$$min\{max\{deg(u), deg(v)\} \mid d(u,v)=2\} \geq n/2$$

then G is hamiltonian.

F9: [BaBrVeLi89] If G is a 2-connected graph of order n and connectivity $\kappa_v(G)$ such that $\sigma_3(G) \geq n+\kappa_v(G)$, then G is hamiltonian.

F10: [ChEr72] Let G be a graph of order $n \geq 3$.

i. If $\kappa_v(G) \geq ind(G)-1$, then G is traceable.

ii. If $\kappa_v(G) \geq ind(G)$, then G is hamiltonian.

iii. If $\kappa_v(G) \geq ind(G)+1$, then G is hamiltonian connected.

F11: [Wo78] If for any nonempty $S \subseteq V$, $|N(S)| \geq \frac{|S|+n+3}{3}$, then G is hamiltonian.

F12: [Fr86] Let G be a k-connected graph of order n. If there exists some $t \leq k$ such that for every independent set S of vertices with cardinality t, we have $|N(S)| \geq \frac{t(n-1)}{t+1}$, then G is hamiltonian.

F13: [BrVe91], [FaGoJaLe92] If G is a 2-connected graph of order $n \geq 11$ such that $|N(S)| \geq n/2$ for every set S of two distinct vertices of G, then G is hamiltonian.

REMARK

R2: The Petersen graph is the only counterexample for $n \leq 10$.

Powers and Line Graphs

TERMINOLOGY: A ***circuit*** is a closed walk having no repeated edges (also called a closed trail).

DEFINITIONS

D10: The ***line graph*** $L(G)$ of a graph G is that graph whose vertices can be put into $1-1$ correspondence with the edges of G in such a way that two vertices of $L(G)$ are adjacent if and only if the corresponding edges of G are adjacent (have an endpoint in common).

D11: A circuit C such that every edge of G is incident on a vertex of C is called a ***dominating circuit***.

D12: We say that G contains a ***k-system that dominates*** if G contains a collection of k edge-disjoint circuits and stars, (here stars are K_{1,n_i}, $n_i \geq 3$), such that each edge of G is either contained in one of the circuits or stars, or is adjacent to one of the circuits.

D13: The ***k-th power*** G^k of a connected graph G is that graph with $V(G^k) = V(G)$ for which $uv \in E(G^k)$ if and only if $1 \leq d_G(u, v) \leq k$.

D14: A ***k-factor*** of a graph G is a k-regular spanning subgraph of G. In particular, a 2-factor is a (vertex-disjoint) union of cycles that covers $V(G)$.

FACTS

F14: [HaNW65] Let G be a graph without isolated vertices. Then $L(G)$ is hamiltonian if and only if $G \simeq K_{1,n}$, for some $n \geq 3$, or G contains a dominating circuit.

F15: [GoHy99] Let G be a graph with no isolated vertices. The graph $L(G)$ contains a 2-factor with k $(k \geq 1)$ cycles if and only if G contains a k-system that dominates.

F16: [ChWa73] If G is connected with $\delta_{min}(G) \geq 3$, then $L^2(G) = L(L(G))$ is hamiltonian.

F17: [Fl74] If G is a 2-connected graph, then G^2 is hamiltonian.

F18: If G is connected then G^3 is hamiltonian (in fact, hamiltonian-connected) (see [Be78]).

Planar Graphs

FACTS

F19: [Th83] Every 4-connected planar graph is hamiltonian connected (and hence, hamiltonian [Tut56]).

F20: [Gr68] Let G be a plane graph of order n with hamiltonian cycle C. If r_i denotes the number of i-sided regions interior to C and r'_i the number of i sided regions exterior to C, then $\sum_{i=3}^{n} (i-2)(r_i - r'_i) = 0$.

4.5.3 Extending the Classics

Adding Toughness

DEFINITION

D15: If every vertex cut-set S of G satisfies $t \cdot c(S) \leq |S|$, where $c(S)$ is the number of components of $G - S$, we say that G is ***t-tough***. The ***toughness*** of G is the maximum t such that G is t-tough.

FACTS

F21: [Ju78] Let G be a 1-tough graph of order $n \geq 11$ such that $\sigma_2(G) \geq n - 4$. Then G is hamiltonian, and this bound is sharp.

F22: [BaMoScVe90] Let G be a 2-tough graph of order n such that $\sigma_3(G) \geq n$. Then G is hamiltonian.

F23: [BrVe90] Let G be a 1-tough graph of order $n \geq 3$ with $\delta_{min}(G) \geq \frac{n+\kappa(G)-2}{3}$. Then G is hamiltonian.

REMARK

R3: Chvátal conjectured that there is a t_0 such that all t_0-tough graphs are hamiltonian. For years $t_0 = 2$ seemed possible. However, in [BaBrLiVe00], examples of $(9/4 - \epsilon)$-tough non-hamiltonian graphs, for arbitrary $\epsilon > 0$, were presented.

More Than Hamiltonian

DEFINITIONS

D16: A graph G of order n is ***pancyclic*** if it contains cycles of all lengths l, $3 \leq l \leq n$.

D17: A bipartite graph G of order n is ***bipancyclic*** if it contains cycles of all possible even lengths from 4 to n.

D18: A graph of order n is ***cycle extendable*** if any cycle C of length $m < n$ can be extended to a cycle of length $m+1$ containing all of $V(C)$. Further, if G is cycle extendable and every vertex is on a triangle, then G is called ***fully cycle extendable.***

D19: A graph is ***k-ordered*** (***hamiltonian***) if for every ordered sequence of k vertices there is a cycle (hamiltonian cycle) that encounters the vertices of the sequence in the given order.

FACTS

F24: [BrChFaGoLe97] If G is a graph of order n satisfying

$$(1)\ \delta_{min}(G) \geq n/2 \text{ and } n \geq 4k \quad \text{or} \quad (2)\ \sigma_2(G) \geq n \text{ and } n \geq 4k$$

then G contains a 2-factor with k cycles for each k, $1 \leq k \leq \lceil n/4 \rceil$, and this result is best possible.

EXAMPLE

E3: To see this result is best possible we need only consider the complete bipartite graph $K_{n/2,n/2}$. The smallest cycle in any 2-factor of this graph is a 4-cycle, hence the bounds on k are sharp.

FACTS

F25: [Bo77] If G is a hamiltonian graph of order n with $|E(G)| \geq \frac{n^2}{4}$, then either G is pancyclic or $G \simeq K_{n/2,n/2}$.

F26: [He90] If G has order $n \geq 3$ and $\sigma_2(G) \geq n$, then G is cycle extendable unless G belongs to one of two special classes. Also, if $\sigma_2(G) \geq (4n-5)/3$, then G is cycle extendable. Further, if $\delta_{min}(G) \geq (n+1)/2$, then G is fully cycle extendable.

F27: [He91] If $G = (X \cup Y, E)$ is a balanced bipartite graph of order $2n$ such that for any non-adjacent pair $x \in X$ and $y \in Y$ we have $deg(x) + deg(y) \geq n+1$, then G is bipancyclic.

F28: [He91] Let $n \geq 2m \geq 2$. If $G = (X \cup Y, E)$ is a balanced bipartite graph of order $2n$ satisfying $\delta_{min}(G) \geq m$ and $|E(G)| > n^2 - mn + m^2$, then G is bipancyclic.

F29: [KoSaSz96], [KoSaSz98] There exists a natural number n_0 such that if G has order n and $n \geq n_0$ and $\delta_{min}(G) \geq kn/(k+1)$, then G contains the k-th power of a hamiltonian cycle.

F30: [KiSaSe99] Let $k \geq 2$ be an integer and let G be a graph of order $n \geq 11k - 3$. If $\deg(u) \geq \lceil \frac{n}{2} \rceil + \lfloor \frac{k}{2} \rfloor - 1$ for every vertex u of G, then G is k-ordered hamiltonian.

F31: [FaGoKoLeScSa] Let k be an integer with $3 \leq k \leq n/2$ and let G be a graph of order n. If $\deg(u) + \deg(v) \geq n + (3k-9)/2$ for every pair u, v of non-adjacent vertices of G, then G is k-ordered hamiltonian.

REMARK

R4: Both of these last two bounds are sharp for the respective values of k. Unexpectedly, the Dirac-type bound does not follow from the Ore-type bound.

4.5.4 More Than One Hamiltonian Cycle?

A Second Hamiltonian Cycle

FACTS

F32: Every edge of a 3-regular graph is contained in an even number of hamiltonian cycles. Thus, every 3-regular hamiltonian graph contains a second and, in fact, a third hamiltonian cycle (see [Tu46]).

F33: [Th98] If G is hamiltonian and m-regular with $m \geq 300$, then G has a second hamiltonian cycle.

F34: [Th97] Let G be a graph with a hamiltonian cycle C. Let A be a vertex set in G such that A contains no two consecutive vertices of C and A is *dominating* in $G - E(C)$ (i.e., $N_{G-E(S)}(A) \geq V(G - E(C))$). Then G has a hamiltonian cycle C' such that $C' - A = C - A$ and there is a vertex v in A such that one of the two edges of C' incident on v is in C and the other is not in C.

F35: [HoSt00] For any real number $k \geq 1$, there exists a function $f(k)$ such that every hamiltonian graph G with $\delta_{max}(G) \geq f(k)$ has at least $\delta_{min}(G) - \lfloor \frac{\delta_{max}(G)}{k} \rfloor + 2$ hamiltonian cycles. In particular, every hamiltonian graph G with $\delta_{max}(G) \geq f(\delta_{max}(G)/\delta_{min}(G))$ has a second hamiltonian cycle.

F36: [Ma76], [GrMa76] There exist 4-regular, 4-connected planar graphs that do not have two edge-disjoint hamiltonian cycles.

F37: [Za76], [Ro89] There exist infinitely many examples of 5-connected planar graphs (both regular and nonregular) in which every pair of hamiltonian cycles have common edges.

REMARK

R5: Fact 32 is due to Smith. Thomason [Th78] extended Smith's result to all r-regular graphs where r is odd (in fact, to all graphs in which all vertices have odd degree). Thomassen extended this further (Fact 33).

Many Hamiltonian Cycles

FACTS

F38: [Th96] Let $C : x_1, y_1, x_2, y_2, \ldots, x_n, y_n, x_1$ be a hamiltonian cycle in a bipartite graph G.

(a) If all the vertices $y_1, \ldots, y_n$ have degree at least 3, then G has another hamiltonian cycle containing the edge x_1y_1.

(b) If all the vertices $y_1, \ldots, y_n$ have degree $d > 3$ and if $P_1, P_2, \ldots, P_q$ $(0 \leq q \leq d-3)$ are paths in C of length 2 of the form $y_{i-1}x_iy_i$, then G has at least $2^{q+1-d}(d-q)!$ hamiltonian cycles containing $P_1 \cup \ldots \cup P_q$.

F39: [FaRoSc85] Let k be a positive integer.

(a) If G is a graph of order $n \geq 60k^2$ such that $\sigma_2(G) \geq n+2k-2$, then G contains k edge-disjoint hamiltonian cycles.

(b) If G has order $n \geq 6k$ and size at least $\binom{n-1}{2} + 2k$, then G contains k edge-disjoint hamiltonian cycles.

F40: [Eg93] Let $n, k \geq 2$ be integers with $n \geq 44(k-1)$. If G is a graph of order n with $\sigma_2(G) \geq n$ and $\delta_{min}(G) \geq 4k-2$, then G contains k edge-disjoint hamiltonian cycles.

Uniquely Hamiltonian Graphs

DEFINITION

D20: A graph is ***uniquely hamiltonian*** if it contains exactly one hamiltonian cycle.

FACTS

F41: [EnSw80] There exist infinitely many uniquely hamiltonian graphs with minimum degree three.

F42: [JaWh89] Any uniquely hamiltonian graph contains a vertex of degree at most $(n+9)/4$ and if there is a unique 2-factor, then the graph contains a vertex of degree 2.

F43: [BoJa98] Every uniquely hamiltonian graph of order n has a vertex of degree at most $c\log_2(8n) + 3$, where $c = (2 - \log_2 3)^{-1} \approx 2.41$. Further, every uniquely hamiltonian plane graph has at least two vertices of degree less than four.

Products and Hamiltonian Decompositions

DEFINITIONS

D21: A ***hamiltonian decomposition*** is a partitioning of the edge set of G into hamiltonian cycles if G is $2d$-regular, or into hamiltonian cycles and a perfect matching if G is $(2d+1)$-regular.

D22: Each of the following four kinds of product graphs has vertex set $V(G_1) \times V(G_2)$.

The ***cartesian product*** $G = G_1 \times G_2$ has edge set

$$E(G) = \{(u_1, u_2)(v_1, v_2) \mid u_1 = v_1 \text{ and } u_2v_2 \in E(G_2) \text{ or } u_2 = v_2 \textit{ and } u_1v_1 \in E(G_1)\}$$

The ***direct product (or conjunction)*** $G = G_1 \cdot G_2$ has edge set

$$E(G) = \{(u_1, u_2)(v_1, v_2) \mid u_1v_1 \in E(G_1) \text{ and } u_2v_2 \in E(G_2)\}$$

The ***strong product*** $G = G_1 \otimes G_2$ has edge set

$$E(G) = \{(u_1, u_2)(v_1, v_2) \mid u_1 = v_1 \text{ and } u_2v_2 \in G_2, \text{ or}$$

$$u_2 = v_2 \text{ and } u_1v_1 \in E(G_1), \text{ or both } u_1v_1 \in E(G_1) \text{ and } u_2v_2 \in E(G_2)\}$$

The ***lexicographic product*** (sometimes called composition, tensor or wreath product) $G = G_1[G_2]$ has edge set

$$E(G) = \{(u_1, u_2)(v_1, v_2) \mid u_1v_1 \in E(G_1), \text{ or } u_1 = v_1 \text{ and } u_2v_2 \in E(G_2)\}$$

REMARK

R6: Jackson [Ja79] conjectured that every k-regular graph on at most $2k+1$ vertices is hamiltonian decomposable. Another natural question is: If G_1 and G_2 are hamiltonian decomposable, is the appropriate product of G_1 and G_2 also hamiltonian decomposable?

FACTS

F44: [St91] Let G_1 and G_2 be two graphs that are decomposable into s and t hamiltonian cycles, respectively, with $t \leq s$. Then $G_1 \times G_2$ is hamiltonian decomposable if one of the following holds:

(1) $s \leq 3t$

(2) $t \geq 3$

(3) the order of G_2 is even, or

(4) the order of G_1 is at least $6\lceil s/t \rceil - 3$.

F45: It is easy to see that if G_1 and G_2 are both bipartite, then the direct product $G_1 \cdot G_2$ is disconnected. Hence, the set of hamiltonian decomposable graphs is not closed under the direct product.

F46: [Bo90], [Zh89] Suppose both G_1 and G_2 are hamiltonian decomposable. If at least one of them has odd order, then $G_1 \cdot G_2$ is hamiltonian decomposable.

F47: [FaLi98] The set of hamiltonian decomposable graphs is closed under strong products, that is, if G_1 and G_2 are hamiltonian decomposable, then so is $G_1 \otimes G_2$.

F48: [BaSz81] The lexicographic product of two hamiltonian decomposable graphs is hamiltonian decomposable.

F49: [Kr97] (a) If G is 1-tough and contains a 2-factor and $|E(H)| \geq 2$ then $G[H]$ is hamiltonian.

(b) If G is 2-tough and $|E(H)| \geq 2$, then $G[H]$ is hamiltonian.

(c) If G is connected and $2k$-regular and $|V(H)| \geq k$, then $G[H]$ is hamiltonian.

(d) If G is $(2k+1)$-regular and connected and G has a 1-factor and $|V(H)| \geq k+1$, then $G[H]$ is hamiltonian.

(e) If G is connected and vertex transitive of degree k and $|V(H)| \geq k/2$, then $G[H]$ is hamiltonian.

(f) If G is connected and vertex transitive and $|E(H)| \geq 2$, then $G[H]$ is hamiltonian.

(g) If G is cubic and 2-edge connected and $|V(H)| \geq 2$, then $G[H]$ is hamiltonian. If G is 4-regular and connected and $|V(H)| \geq 2$, then $G[H]$ is hamiltonian.

4.5.5 Random Graphs

NOTATION: We shall use $Pr(X)$ to denote the probability of event X and we let $N = \binom{n}{2}$.

DEFINITIONS

D23: (***The edge density model***) Suppose that $0 \leq p \leq 1$. Let $G_{n,p}$ denote a graph on n vertices obtained by inserting any of the N possible edges with probability p.

D24: (***The fixed size model***) Suppose that $M = M(n)$ is a prescribed function of n which takes on values in the set of positive integers. Then there are $s = \binom{N}{M}$ different graphs with M edges possible on the vertex set $\{1, 2, \ldots, n\}$. We let $G_{n,M}$ denote one of these graphs chosen uniformly at random with probability $1/s$.

D25: A somewhat different approach is to consider a ***graph process*** as a sequence $(G_t)_{t=0}^{N}$ such that

1. each G_t is a graph on V,
2. G_t has t edges for $t = 0, 1, \ldots, N$,
3. $G_0 \subset G_1 \subset \ldots$.

D26: If Ω_n is a model of random graphs of order n, we say ***almost every graph*** in Ω_n has property Q if $Pr(Q) \to 1$ as $n \to \infty$. Note that this is equivalent to saying that the proportion of all labeled graphs of order n that have Q tends to 1 as $n \to \infty$.

D27: The ***k-in, l-out random digraph*** $D_{k-in,l-out}$ has n vertices, and for each vertex v, a subset of k arcs into v and l arcs out of v are chosen independently and uniformly at random. The union of these arc subsets is the arc-set of $D_{k-in,l-out}$.

FACTS

F50: [Po76], [Ko76] There exists a constant c such that almost every labeled graph of order n having at least $c\,n\log_2 n$ edges is hamiltonian.

F51: [Ko76], [KoSz83] Suppose $\omega(n) \to \infty$ as $n \to \infty$, and let $p = \frac{1}{n}\{\log_2 n + \log_2\log_2 n + \omega(n)\}$ and $M(n) = \lfloor \frac{n}{2}\{\log_2 n + \log_2\log_2 n + \omega(n)\}\rfloor$. Then almost every $G_{n,p}$ is hamiltonian and almost every $G_{n,M}$ is hamiltonian.

F52: [KoSz83] For $M(n) = n/2\,(\log_2 n + \log_2\log_2 n + c_n)$,

$$\lim_{n\to\infty} Pr(G_{n,M} \text{ is hamiltonian}) = \begin{cases} 0 & : \text{if } c_n \to -\infty \\ e^{-e^{-c}} & : \text{if } c_n \to c \\ 1 & : \text{if } c_n \to \infty \end{cases}$$

F53: [RoWo92, RoWo94] For every $r \geq 3$, almost all r-regular graphs are hamiltonian.

F54: [CoFr94] Almost all random digraphs $D_{3-in,3-out}$ are hamiltonian.

F55: [CoFr00] Almost all random digraphs $D_{2-in,2-out}$ are hamiltonian. In particular, this implies that G_{4-out}, the underlying graph of $D_{2-in,2-out}$, is hamiltonian. On the other hand, almost all $D_{1-in,2-out}$ and $D_{2-in,1-out}$ are not hamiltonian.

REMARKS

R7: In the probability space of all $N!$ graph processes (with equal probability), one can consider when a property "appears" (called the *hitting time*). Erdös and Spencer were the first to conjecture that with probability tending to 1, the very edge that increases the minimum degree to 2 also makes the graph hamiltonian. This was verified by Bollobás.

R8: It is natural to ask whether there exists a polynomial algorithm that, with probability tending to 1, finds a hamiltonian cycle in $G_{n,M(n)}$. Bollobás, Fenner and Frieze [BoFeFr85] constructed such an algorithm that is essentially best possible.

R9: Still open is the question of hamiltonicity for G_{3-out}.

4.5.6 Forbidden Subgraphs

DEFINITION

D28: A graph G is said to be $\{F_1, F_2, \ldots, F_k\}$***-free*** if G contains no induced subgraph isomorphic to any F_i, $1 \leq i \leq k$.

NOTATION: (a) The graph $N_{i,j,k}$ is a graph that consists of K_3 and three vertex-disjoint paths of lengths i, j, and k, with one path rooted at each of the three vertices of K_3.

(b) The graph L consists of two vertex-disjoint copies of K_3 and an edge joining them.

(c) The graph P_i is i-vertex path, and $K_{1,3}$ is the 4-vertex star (also called a *claw*).

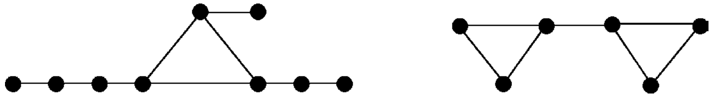


Figure 4.5.3 **The graphs $N_{1,2,3}$ and L.**

FACTS

F56: [DuGoJa81] If G is a $\{K_{1,3}, N_{1,1,1}\}$-free graph, then

(a) if G is 2-connected, then G is hamiltonian;

(b) if G is connected, then G is traceable.

F57: [BrDrKo00] There exists a linear-time algorithm for finding a hamiltonian cycle in a $\{K_{1,3}, N_{1,1,1}\}$-free graph.

F58: [BrVe90] If G is a 2-connected $\{K_{1,3}, P_6\}$-free graph, then G is hamiltonian.

F59: [GoJa82] If G is a 2-connected $\{K_{1,3}, N_{2,0,0}\}$-free graph, then G is hamiltonian.

F60: [Be91] If G is a 2-connected $\{K_{1,3}, N_{2,1,0}\}$-free graph, then G is hamiltonian.

F61: [FaGoRySc95] If G is a 2-connected $\{K_{1,3}, N_{3,0,0}\}$-free graph of order $n \geq 10$, then G is hamiltonian.

Other Forbidden Pairs

A natural question is: Are these the only such pairs? This was investigated in [Be91] for all graphs and in [FaGo97] for graphs of order 10 or more. We now summarize these combined results.

FACTS

F62: [Be91], [FaGo97] Let R and S be connected graphs $(R, S \neq P_3)$ and G a 2-connected graph of order n. Then G is $\{R, S\}$-free implies G is hamiltonian if and only if $R = K_{1,3}$ and S is one of the graphs $N_{1,1,1}$, P_6, $N_{2,1,0}$, $N_{2,0,0}$, (or $N_{3,0,0}$ when $n \geq 10$), or a connected induced subgraph of one of these graphs.

F63: [FaGo97] Let R, S be connected graphs $(R, S \neq P_3)$ and let G $(G \neq C_n)$ be a 2-connected graph of order $n \geq 10$. Then G is $\{R, S\}$-free implies G is pancyclic if and only if $R = K_{1,3}$ and S is one of P_4, P_5, P_6, $N_{1,0,0}$ or $N_{2,0,0}$.

F64: [GoLuPf] Let X and Y be connected graphs on at least three vertices such that $X, Y \neq P_3$ and $Y \neq K_{1,3}$. Then the following statements are equivalent:

(a) Every 3-connected $\{X, Y\}$-free graph G is pancyclic.

(b) $X = K_{1,3}$ and Y is a subgraph of one of the graphs from the family $\mathcal{F} = \{P_7, L, N_{4,0,0}, N_{3,1,0}, N_{2,2,0}, N_{2,1,1}\}$.

F65: [FaGo97] It is an easy observation that P_3 is the only nontrivial single graph that when forbidden implies G is hamiltonian.

Claw-Free Graphs

In each of the forbidden-pair results above, the claw $K_{1,3}$ is one of the two forbidden graphs. This led naturally to the question: Is the claw in every triple of forbidden subgraphs implying hamiltonicity? This was answered in the negative in [FaGoJaLe02], where all forbidden triples containing no $K_{1,t}$ with $t \geq 3$ for sufficiently large 2-connected graphs were given. Further, in [FaGoJa2] other forbidden triples implying the hamiltonicity of sufficiently large graphs were investigated. Brousek [Br02] gave the collection of all forbidden triples which include the claw that imply hamiltonicity for 2-connected graphs. In [FaGoJa], all possible remaining forbidden triples implying hamiltonicity for any graph were given.

DEFINITIONS

D29: For a vertex x such that the induced subgraph $G[N(x)]$ is connected, a ***local completion*** of G at x is the graph obtained by replacing $G[N(x)]$ by a complete subgraph on $V(N(x))$. (Observe that a local completion of a claw-free graph is claw-free.)

D30: The ***claw-free closure*** of a claw-free graph G, denoted $cl(G)$, is that graph obtained by repeatedly finding the local completion of a vertex x until it is no longer possible.

D31: The ***circumference*** of a graph G, denoted by $circum(G)$, is the length of a longest cycle in G.

FACTS

F66: [FaGo97] Let R, S be connected graphs ($R, S \neq P_3$) and G a 2-connected graph of order $n \geq 10$. Then G is $\{R, S\}$-free implies G is cycle extendable if and only if $R = K_{1,3}$ and S is one of K_3, P_4, $N_{1,0,0}$ or $N_{2,0,0}$.

F67: [Sh97] If G is a 3-connected $\{K_{1,3}, N_{1,1,1}\}$-free graph, then G is hamiltonian-connected.

F68: [Ry97] Let G be a claw-free graph. Then

(a) the closure $cl(G)$ is well-defined,

(b) there is a triangle-free graph H such that $cl(G) = H$,

(c) $circum(G) = circum(cl(G))$.

REMARKS

R10: The claw-free closure is different from the degree closure (see [BoCh76]) or any of several other closures that have recently been developed. For more information on closures, see [BrRySc00].

R11: By Fact 68, if G is claw-free, then G is hamiltonian if and only if $cl(G)$ is hamiltonian.

R12: The vast area of hamiltonian graphs contains far more than can be written here. For more details on hamiltonian graphs, the reader should see [Be78], [Bo78], [WiGa84], [Bo95], [CuGa96], [Go91] and [Go03].

References

[BaSz81] Z. Baranyai and Gy. R. Szasz, Hamiltonian decompositions of lexicographic product, *J. Combin. Theory Ser. B* 31 (1981), 253–261.

[BaBrLiVe00] D. Bauer, H. J. Broersma, R. Li and H. J. Veldman, Not every 2-tough graph is hamiltonian, *Discrete. Appl. Math.* 99 (2000), 317–321.

[BaBrVeLi89] D. Bauer, H. J. Broersma, H. J. Veldman and R. Li, A generalization of a result of Haggkvist and Nicoghossian, *J. Combin. Theory Ser. B* 47 (1989), No. 2, 237–243.

[BaMoScVe90] D. Bauer, A. Morgana, E. Schmeichel and H. J. Veldman, Long cycles in graphs with large degree sums, *Discrete Math.* 79 (1989/90), 59–70.

[BoCh76] J. A. Bondy and V. Chvátal, A method in graph theory, *Discrete Math.* 15 (1976), no. 2, 111–135.

[BrChFaGoLe97] S. Brandt, G. Chen, R. J. Faudree, R. J. Gould, and L. Lesniak, On the number of cycles in a 2-factor, *J. Graph Theory* 24 (1997), 165–173.

[Be91] P. Bedrossian, *Forbidden subgraphs and minimum degree conditions for hamiltonicity*, Ph.D. Thesis, Memphis State University, 1991.

[Be78] J.-C. Bermond, Hamiltonian graphs, in *Selected Topics in Graph Theory*, L. Beineke and R. Wilson, Eds., Academic Press, London, (1978).

[BiLlWi86] N. L. Biggs, E. K. Lloyd and R. J. Wilson, *Graph Theory 1736-1936*, Oxford University Press, Oxford, (1986).

[BoFeFr85] B. Bollobás, T. I. Fenner and A. M. Frieze, On matchings and hamiltonian cycles in random graphs, in *Proceedings of ACM Symposium on Theory of Computing*, New York, 1985, 430–439.

[Bo84] B. Bollobás, The evolution of sparse random graphs, *Trans. Am. Math. Soc.* 286 (1984), 257–274.

[BoJa98] J. A. Bondy and B. Jackson, Vertices of small degree in uniquely hamiltonian graphs, *J. Combin. Theory Ser. B* 74 (1998), 265–275.

[Bo77] J. A. Bondy, Pancyclic graphs, *J. Combin. Theory Ser. B* 11 (1977), 80–84.

[Bo78] J. A. Bondy, Hamilton cycles in graphs and digraphs, *Proceedings 9th S.E. Conf. on Combin., Graph Theory and Computing*, in Congr. Numer. XXI (1978), 3–28.

[Bo95] J. A. Bondy, Basic graph theory – paths and cycles, *Handbook of Combinatorics I*, Elsevier, Amsterdam, (1995), 5–110.

[Bo90] J. Bosak, *Decompositions of Graphs*, Kluwer Academic Publishers, (1990).

[Br02] J. Brousek, Forbidden triples for hamiltonicity, *Discrete Math.* 251 (2002), 71–76.

[BrDrKo00] A. Brandstädt, F. F. Dragan, and E. Köhler, Linear time algorithms for hamiltonian problems on (claw, net)-free graphs, *SIAM J. Comput.* 30 (2000), 1662–1677.

[BrRySc00] H. Broersma, Z. Ryjáček, and I. Schiermeyer, Closure concepts: a survey, *Graphs and Combinatorics* 16 (2000), 17–48.

[BrVe90] H. J. Broersma and H. J. Veldman, Restrictions on induced subgraphs ensuring hamiltonicity or pancyclicity of $K_{1,3}$-free graphs, *Contemporary Methods in Graph Theory*, R. Bodendiek, Ed., BI-Wiss.-Verl., Mannheim-Wien-Zurich, (1990), 181–194.

[BrVe91] H. J. Broersma and H. J. Veldman, Long dominating cycles and paths in graphs with large neighborhood unions, *J. Graph Theory* 15 (1991), 29–38.

[ChWa73] G. Chartrand and C. E. Wall, On the hamiltonian index of a graph, *Studia Sci. Math Hungar.* 8 (1973), 43–48.

[ChEr72] V. Chvátal and P. Erdös, A note on hamiltonian circuits, *Discrete Math.* 2 (1972), 111–113.

[CoFr94] C. Cooper and A. M. Frieze, Hamilton cycles in a class of random directed graphs, *J. Combin. Theory Ser. B* 62 (1994), 151–163.

[CoFr00] C. Cooper and A. M. Frieze, Hamilton cycles in random graphs and directed graphs, *Random Structures Algorithms* 16 (2000), 369–401.

[CuGa96] S. J. Curran and J. A. Gallian, Hamiltonian cycles and paths in cayley graphs and digraphs - a survey, *Discrete Math.* 156 (1996), 1–18.

[Di52] G. A. Dirac, Some theorems on abstract graphs, *Proc. London Math. Soc.* 2 (1952), 69–81.

[DuGoJa81] D. Duffus, R. J. Gould, and M. S. Jacobson, Forbidden subgraphs and the hamiltonian theme, *The Theory and Applications of Graphs*, ed. by G. Chartrand, J. Alavi, D. Goldsmith, L. Lesniak, and D. Lick (1981) 297–316.

[Eg93] Y. Egawa, Edge-disjoint hamiltonian cycles in graphs of Ore type, *SUT J. Math.* 29 (1993), 15–50.

[EnSw80] R. Entringer and H. Swart, Spanning cycles of nearly cubic graphs, *J. Combin. Theory Ser. B* 29 (1980), 303–309.

[Fa84] G. H. Fan, New sufficient condition for cycles in graphs, *J. Combin. Theory Ser B* 37 (1984), 221–227.

[FaLi98] C. Fan and J. Liu, Hamiltonian decompositions of strong products, *J. Graph Theory* 29 (1998), 45–55.

[FaGo97] R. J. Faudree and R.J. Gould, Characterizing forbidden pairs for hamiltonian properties, *Discrete Math.* 173 (1997) 45–60.

[FaGoJa-a] R. J. Faudree, R. J. Gould, and M. S. Jacobson, Forbidden triples implying hamiltonicity: for all graphs, Preprint.

[FaGoJa-b] R. J. Faudree, R. J. Gould, and M. S. Jacobson, Forbidden triples implying hamiltonicity: for sufficiently large graphs, Preprint.

[FaRoSc85] R. J. Faudree, C. Rousseau, R. Schelp, Edge disjoint hamiltonian cycles, *Graph theory with Applications to Algorithms and Computer Science* (Kalamazoo, Mich., 1984), Wiley, New York (1985), 231–249.

[FaGoJaLe92] R. J. Faudree, R. J. Gould, M. S. Jacobson, and L. Lesniak, On a generalization of Dirac's theorem, *Discrete Math.* 105 (1992), 61–71.

[FaGoJaLe02] R. J. Faudree, R.J. Gould, M. S. Jacobson, and L. Lesniak, Characterizing forbidden clawless triples implying hamiltonian graphs, *Discrete Math.* 249 (2002), 71–81.

[FaGoKoLeScSa03] R. J. Faudree, R. J. Gould, A. Kostochka, L. Lesniak, I. Schiermeyer, and A. Saito, Degree Conditions for k-ordered hamiltonian graphs, *J. Graph Theory* 42, No. 3, (2003), 199–210.

[FaGoRySc95] R. J. Faudree, R. J. Gould, Z. Ryjáček, and I. Schiermeyer, Forbidden subgraphs and pancyclicity, *Congr. Numer.* 109 (1995), 13–32.

[Fl74] H. Fleischner, The square of every two-connected graph is hamiltonian, *J. Combin. Theory Ser. B* 16 (1974), 29–34.

[Fr86] P. Fraisse, A new sufficient condition for graphs, *J. Graph Theory* 10 (1986), 405–409.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Co., New York (1979).

[GoLuPf] R. J. Gould, T. Łuczak, and F. Pfender, Pancyclicity of 3-connected graphs: pairs of forbidden subgraphs, Preprint.

[Go91] R. J. Gould, Updating the hamiltonian problem – a survey, *J. Graph Theory* 15 (1991), 121–157.

[Go03] R. J. Gould, Advances on the hamiltonian problem – a survey, *Graphs and Combinatorics* 19, No. 1 (2003), 7–52.

[GoHy99] R. J. Gould and E. Hynds, A note on cycles in 2-factors of line graphs, *Bull. of the I.C.A.* 26 (1999), 46–48.

[GoJa82] R. J. Gould and M. S. Jacobson, Forbidden subgraphs and hamiltonian properties of graphs, *Discrete Math.* 42 (1982), 189–196.

[Gr68] E. J. Grinberg, Plane homogeneous graphs of degree three without hamiltonian circuits, *Latvian Math. Yearbook* 4 (1968), 51–58.

[GrMa76] B. Grunbaum and J. Malkevitch, Pairs of edge-disjoint hamiltonian circuits, *Aequationes Math.* 14 (1976), 191–196.

[HaNW65] F. Harary and C. St. J. A. Nash-Williams, On eulerian and hamiltonian graphs and line graphs, *Canad. Math. Bull.* (1965), 701–710.

[He90] G. Hendry, Extending cycles in graphs, *Discrete Math.* 85 (1990), 59–72.

[He91] G. Hendry, Extending cycles in bipartite graphs, *J. Combin. Theory Ser. B* 51 (1991), 292–313.

[HoSt00] P. Horak and L. Stacho, A lower bound on the number of hamiltonian cycles, *Discrete Math.* 222 (2000), 275–280.

[Ja79] B. Jackson, Edge-disjoint hamiltonian cycles in regular graphs of large degree, *J. London Math. Soc.* (2) 19 (1979), 13–16.

[Ja80] B. Jackson, Hamilton cycles in regular 2-connected graphs, *J. Combin. Theory Ser B* 29 (1980), 27–46.

[JaWh89] B. Jackson and R. W. Whitty, A note concerning graphs with unique f-factors, *J. Graph Theory* 13 (1989), 577–580.

[Ju78] H. A. Jung, On maximal circuits in finite graphs, *Annals of Discrete Math.* 3 (1978), 129–144.

[KiSaSe99] H. S. Kierstead, G. N. Sárközy, and S. Selkow, On k-ordered hamiltonian graphs, *J. Graph Theory* 32 (1999), 17–25.

[Ki56] T. P. Kirkman, On the representation of polyedra, *Philosophical Transactions of the Royal Society* (London) 146 (1856), 413–418.

[KoSz83] J. Komlós and E. Szermerédi, Limit distribution for the existence of hamiltonian cycles in random graphs, *Discrete Math.* 43 (1983), 55–63.

[KoSaSz96] J. Komlós, G. N. Sárközy, and E. Szemerédi, On the square of a hamiltonian cycle in dense graphs, *Random structures and Algorithms* 9 (1996), 193–211.

[KoSaSz98] J. Komlós, G. N. Sárközy, and E. Szemerédi, Proof of the Seymour conjecture for large graphs, *Ann. Comb.* 2 (1998), n 43–60.

[Ko76] A. D. Korshunov, Solution of a problem of Erdös and Rényi on Hamilton cycles in non-oriented graphs, *Soviet Mat. Dokl.* 17 (1976), 760–764.

[Kr97] M. Kriesell, A note on hamiltonian cycles in lexicographic products, *J. Autom. Lang. Comb.* 2 (1997), 135–138.

[Ma76] P. Martin, Cycles hamiltoniens dans les graphes 4-reguliers 4-connexes, *Aequationes Math.* 14 (1976), 37–40.

[MoMo63] J. Moon and L. Moser, On hamiltonian cycles in bipartite graphs, *Isr. J. Math.* 1 (1963), 163–165.

[Or60] O. Ore, A note on hamiltonian circuits, *Amer. Math. Monthly* 67 (1960), 55.

[Or63] O. Ore, Hamiltonian connected graphs, *J. Math. Pures. Appl.* 42 (1963), 21–27.

[Po76] L. Pósa, Hamiltonian circuits in random graphs, *Discrete Math.* 14 (1976), 359–364.

[RoWo92] R. W. Robinson and N. C. Wormald, Almost all cubic graphs are hamiltonian, *Random Structures Algorithms* 3 (1992), 117–125.

[RoWo94] R. W. Robinson and N. C. Wormald, Almost all regular graphs are hamiltonian, *Random Structures Algorithms* 5 (1994), 363–374.

[Ro89] M. Rosenfeld, Pairs of edge disjoint hamiltonian circuits in 5-connected planar graphs, *Aequationes Math.* 38 (1989), 50–55.

[Ry97] Z. Ryjáček, On a closure concept in claw-free graphs, *J. Combin. Theory Ser B* 70 (1997), 217–224.

[Sh97] F. B. Shepherd, Hamiltonicity in claw-free graphs, *J. Combin. Theory Ser. B* 53 (1991) 173–194.

[St91] R. Stong, Hamilton decompositions of cartesian products of graphs, *Discrete Math.* 90 (1991), 169–190.

[Th78] A. G. Thomason, Hamiltonian cycles and uniquely edge colourable graphs, *Ann. Discrete Math.* 3 (1978), 259–268.

[Th83] C. Thomassen, A theorem on paths in planar graphs, *J. Graph Theory* 7 (1983), 169–176.

[Th96] C. Thomassen, On the number of hamiltonian cycles in bipartite graphs, *Combinatorics, Probability and Computing*, 5 (1996), 437–442.

[Th97] C. Thomassen, Chords of longest cycles in cubic graphs, *J. Combin. Theory Ser. B* 71 (1997), 211–214.

[Th98] C. Thomassen, Independent dominating sets and a second hamiltonian cycle in regular graphs, *J. Combin. Theory Ser. B* 72 (1998), 104–109.

[Tu46] W. T. Tutte, On hamiltonian circuits, *J. London Math. Soc.* 21 (1946), 98–101.

[Tu56] W. T. Tutte, A theorem on planar graphs, *Trans. Amer. Math. Soc.* 82 (1956), 99–116.

[WiGa84] D. Witte and J. A. Gallian, A survey – Hamiltonian cycles in Cayley graphs, *Discrete Math.* 51 (1984), 293–304.

[Wo78] D. R. Woodall, A sufficient condition for hamiltonian circuits, *J. Combin. Theory Ser. B* 25 (1978), 184–186.

[Za76] J. Zaks, Pairs of hamiltonian circuits in 5-connected planar graphs, *J. Combin. Theory, Ser. B* 21 (1976), 116–131.

[Zh89] M. Zhou, Decomposition of some product graphs into 1-factors and hamiltonian cycles, *Ars Combinatoria* 28 (1989), 258–268.

4.6 TRAVELING SALESMAN PROBLEMS

Gregory Gutin, Royal Holloway, University of London, UK

4.6.1 The Traveling Salesman Problem
4.6.2 Exact Algorithms
4.6.3 Construction Heuristics
4.6.4 Improvement Heuristics
4.6.5 The Generalized TSP
4.6.6 The Vehicle Routing Problem
References

Introduction

The Traveling Salesman Problem (TSP) is perhaps the most studied discrete optimization problem. Its popularity is due to the facts that TSP is easy to formulate, difficult to solve, and has a large number of applications. TSP has a number of variations and generalizations extensively studied in the literature [Pu02]. In this section, we consider TSP, the Generalized TSP and the Vehicle Routing Problem.

4.6.1 The Traveling Salesman Problem

K. Menger [Me32] was perhaps the first researcher to consider the Traveling Salesman Problem (TSP). He observed that the problem can be solved by examining all permutations one by one. Realizing that the complete enumeration of all permutations was not possible for graphs with a large number of vertices, he looked at the most natural *nearest neighbor* strategy and pointed out that this *heuristic*, in general, does not produce the shortest route. In fact, as we will see below, the nearest neighbor heuristic will generate the worst possible route for some problem instances. (For an interesting overview of TSP history, see [HoWo85].)

In applications, both the symmetric and asymmetric versions of the TSP are important.

Symmetric and Asymmetric TSP

DEFINITIONS

D1: ***Symmetric TSP*** (***STSP***):
Given a complete (undirected) graph K_n with weights on the edges, find a hamiltonian cycle in K_n of minimum (total) weight.

D2: ***Asymmetric TSP*** (***ATSP***):
Given a complete directed graph $\overset{\leftrightarrow}{K}_n$ with weights on the arcs, find a hamiltonian cycle in $\overset{\leftrightarrow}{K}_n$ of minimum weight.

D3: The ***Euclidean TSP*** is the special case of STSP in which the vertices are points in the Euclidean plane and the weight on each edge is the Euclidean distance between its endpoints.

D4: A hamiltonian cycle in K_n or $\overleftrightarrow{K}_n$ is called a ***tour***.

NOTATION: Throughout this section, the set $\{1, 2, \ldots, n\}$ denotes the vertices of K_n or $\overleftrightarrow{K}_n$ or any other n-vertex graph under discussion.

NOTATION: By ***TSP*** we refer to both STSP and ATSP simultaneously.

Matrix Representation of TSP

Every instance of TSP can be associated with the matrix of edge-weights of the corresponding complete graph. Such a matrix is symmetric for STSP and, in general, asymmetric for ATSP.

DEFINITIONS

D5: The ***distance*** (or ***weight***) ***matrix*** of an instance of STSP is the matrix $D = [d_{ij}]$, where d_{ij} is the weight of the edge between vertices i and j. The ***distance matrix*** of an instance of ATSP is the matrix $D = [d_{ij}]$, where d_{ij} is the weight of the arc directed from i to j. Accordingly, the diagonal entries d_{ii} are set to zero.

D6: An instance of TSP is said to satisfy the ***triangle inequality*** if $d_{ij} + d_{jk} \geq d_{ik}$ for all distinct vertices i, j, k.

EXAMPLES

E1: An instance of ATSP with distance matrix

$$D = \begin{pmatrix} 0 & 6 & 5 & 10 \\ 3 & 0 & 3 & 9 \\ 7 & 4 & 0 & 8 \\ 12 & 7 & 5 & 0 \end{pmatrix}$$

is shown in Figure 4.6.1. There are $3! = 6$ tours of total weight 29, 27, 30, 23, 27, and 22. The optimal tour of weight 22 is $\langle 1, 4, 3, 2, 1 \rangle$.

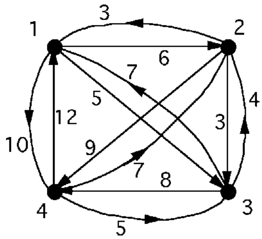


Figure 4.6.1 An instance of ASTP.

E2: An instance of STSP with distance matrix

$$D = \begin{pmatrix} 0 & 10 & 7 & 7 & 11 \\ 10 & 0 & 9 & 6 & 5 \\ 7 & 9 & 0 & 9 & 10 \\ 7 & 6 & 9 & 0 & 6 \\ 11 & 5 & 10 & 6 & 0 \end{pmatrix}$$

is shown in Figure 4.6.2. Since this graph has 5 vertices, there are $4!/2 = 12$ tours. The optimal tour is $\langle 1,3,2,5,4,1\rangle$ of weight 34.

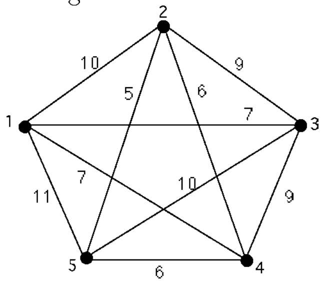


Figure 4.6.2 **An instance of SSTP.**

Algorithmic Complexity

FACTS

F1: The hamiltonian cycle problem on an n-vertex graph G can be transformed into STSP by converting G to an edge-weighted K_n as follows: assign weight 0 to each edge of G; and assign weight 1 to each edge in the edge-complement of G. A similar transformation can be used for digraphs and ATSP.

F2: Fact 1 implies that TSP is NP-hard, even if the triangle inequality holds.

By replacing the weights 0 by 1 and the weights 1 by nr in the transformation used in Fact 1, we obtain the following result.

F3: [SaGo76] For an arbitrary constant r, unless P=NP, there is no polynomial time algorithm that always produces a tour of total weight at most r times the optimal.

Exact and Approximate Algorithms

DEFINITIONS

D7: An ***exact algorithm*** is an algorithm that always produces an optimal solution.

D8: An ***approximate*** (or ***approximation***) ***algorithm*** is an algorithm that typically makes use of heuristics in reducing its computation but produces solutions that are not necessarily optimal.

NOTATION: Let $\mathcal{A}$ be an approximate algorithm for TSP and I a problem instance. Then $w_{\min}(I)$, $w_{\max}(I)$, $w_{\mathcal{A}}(I)$ denote the weights, respectively, of an optimal tour, a heaviest tour, and a tour produced by algorithm $\mathcal{A}$, for instance I.

D9: The ***Zemel measure*** [Ze81] of an algorithm $\mathcal{A}$, denoted $\rho_z(\mathcal{A})$, is the supremum of $(w_{\mathcal{A}}(I) - w_{\min}(I))/(w_{\max}(I) - w_{\min}(I))$, taken over all TSP instances I for which $w_{\max}(I) \neq w_{\min}(I)$.

FACT

F4: [HaKh01] There is a polynomial time approximate algorithm $\mathcal{A}$ for ATSP with $\rho_z(\mathcal{A}) \leq \frac{1}{2}$, and one for STSP with $\rho_z(\mathcal{A}) \leq \frac{1}{3}$.

The Euclidean TSP

Despite Fact 5 below, there was a feeling among some researchers that the Euclidean TSP is somewhat simpler than the general STSP. This was confirmed by Arora [Ar98] in 1996 (see Fact 6). Mitchell [Mi99] independently made a similar discovery a few months later (see [Ar02]).

FACTS

F5: [Pa77,GaGrJo76] Euclidean TSP is NP-hard.

F6: [Ar02] For every $\epsilon > 0$, there is a polynomial time algorithm $\mathcal{A}_\epsilon$ that, for any instance of the Euclidean TSP, finds a tour at most $1+\epsilon$ times longer than the optimal one.

F7: As of this writing, the fastest algorithm $\mathcal{A}_\epsilon$ has time complexity $O(n \log n + n/\text{poly}(\epsilon))$ [RaSm98].

COMPUTATIONAL NOTE: These $\mathcal{A}_\epsilon$ algorithms have been implemented, but, in their current form, they are not competitive with other TSP heuristics [Ar02].

F8: [Tr97] There exists a constant $r > 1$ such that, for the Euclidean TSP in $O(\log n)$-dimensional Euclidean space, the problem of finding a tour that is at most r times longer than the optimal tour is NP-hard.

REMARKS

R1: Arora's result (Fact 6) can be generalized to d-dimensional Euclidean space for any constant d. However, Fact 8 limits the scope of this generalization.

R2: Exact algorithms cannot be relied on for applications requiring very fast solutions (online, for example) or ones that involve huge problem instances. Although approximate algorithms forfeit the guarantee of optimality, with good heuristics they can normally produce solutions close to optimal. For many applications this is good enough, since often the data are inexact anyway.

R3: TSP heuristics can be roughly partitioned into two classes: *construction heuristics*, discussed in §4.5.3, and *improvement heuristics*, discussed in §4.5.4. More comprehensive overviews of TSP heuristics can be found in [GoSt85], [JoGuMcYeZhZv02], and [JoMc02].

4.6.2 Exact Algorithms

The NP-Hardness results mentioned in the previous subsection indicate that it is rather difficult to solve large instances of TSP to optimality. Nevertheless, there are computer codes that can solve many instances with thousands of vertices within days (on a single-processor computer) [ApBiChCo98]. For a discussion of TSP software implementing both exact algorithms and heuristics, see [LoPu02].

FACT

F9: The *brute-force* method of explicitly examining all possible TSP tours is impractible for even moderately sized problem instances because there are $\frac{(n-1)!}{2}$ different tours in K_n and $(n-1)!$ different tours in $\overset{\leftrightarrow}{K}_n$.

Integer Programming Approaches

Various methods have been suggested to solve TSP to optimality. They include *Lagrangian relaxation* ([BeLu00]), *dynamic programming* ([PaSt82]), and *branch-and-bound* and *branch-and-cut* (see [BaTo85], [FiLoTo02], and [Na02]). These are all well-known methods in integer programming ([Wo98]). The earliest (and still useful) integer programming formulation of ATSP is due to Dantzig, Fulkerson, and Johnson [DaFuJo54].

DANTZIG, FULKERSON, AND JOHNSON FORMULATION: Define zero-one variables x_{ij} by

$$x_{ij} = \begin{cases} 1, & \text{if the tour traverses arc } (i,j) \\ 0, & \text{otherwise} \end{cases}$$

Let d_{ij} be the weight on arc (i,j). Then ATSP can be expressed as:

$$\begin{aligned} \min z = & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \\ \text{subject to} & \sum_{i=1}^{n} x_{ij} = 1, \; j = 1, 2, \ldots, n \\ & \sum_{j=1}^{n} x_{ij} = 1, \; i = 1, 2, \ldots, n \\ & \sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \text{ for all } 0 < |S| < n \\ & x_{ij} = 0 \text{ or } 1, \quad i, j = 1, \ldots, n \end{aligned}$$

FACTS

F10: The first set of constraints ensures that a tour must come into vertex j exactly once, and the second set of constraints indicates that a tour must leave every vertex i exactly once. These two sets of constraints ensure that there are two arcs adjacent to each vertex, one in and one out. However, this does not prevent non-hamiltonian cycles. Instead of having one tour, the solution can consist of two of more vertex-disjoint cycles (called *sub-tours*).

F11: The third set of constraints, called *sub-tour elimination constraints*, requires that no proper subset of vertices, S, can have a total of $|S|$ arcs.

F12: The formulation without the third set of constraints is an integer programming formulation of the *Assignment Problem* that can be solved in time $O(n^3)$ [Wo98]. A solution of the Assignment Problem is a minimum-weight collection of vertex-disjoint cycles $C_1, \ldots, C_t$ spanning $\overset{\leftrightarrow}{K}_n$. If $t = 1$, then an optimal solution of ATSP has been obtained. Otherwise, one can consider two or more subproblems. For example, for a particular arc $a \in C_i$, one subproblem could add the constraint that arc a be in the solution, and a second subproblem could require that a not be in the solution. This simple idea gives a basis for branch-and-bound algorithms for ATSP.

4.6.3 Construction Heuristics

Approximate algorithms based on construction heuristics build a tour from scratch and stop when one is produced.

Greedy-Type Algorithms

The simplest and most obvious construction heuristic is *nearest neighbor*. The nearest neighbor greedy algorithm constructs a tour by always choosing as the next vertex to visit, one that is nearest to the last one visited.

Algorithm 4.6.1: Nearest Neighbor (NN)

Input: $n \times n$ distance matrix $[d_{ij}]$ and a fixed vertex i_1.
Output: TSP tour $\langle i_1, i_2, \ldots, i_n, i_1 \rangle$.

```
Initialize $S := \{1, 2, \ldots, n\} - \{i_1\}$.
For $k = 2, 3, \ldots, n$
    Choose $i_k$ such that $d_{i_{k-1}, i_k} = \min_{s \in S}\{d_{i_{k-1}, s}\}$.
    $S := S - \{i_k\}$.
```

A second greedy-type algorithm is based on the observation that a vertex-disjoint collection of paths in $\overleftrightarrow{K}_n$ (K_n) can be extended to a tour in $\overleftrightarrow{K}_n$ (K_n).

Algorithm 4.6.2: Greedy Heuristic (GR)

Input: $n \times n$ distance matrix $[d_{ij}]$.
Output: ATSP (STSP) tour as a set S of arcs (edges).

```
Set $S = \emptyset$ and $m = n(n-1)$ (for ATSP) or $m = n(n-1)/2$ (for STSP).
Sort the arcs (edges) $a_1, a_2, \ldots, a_m$ in non-decreasing order of weight.
For $i = 1, 2, \ldots, m$
    If $S \cup \{a_i\}$ is the arc (edge) set of a collection of vertex-disjoint paths
     or is the arc (edge) set of a tour,
        $S := S \cup \{a_i\}$
```

EXAMPLE

E3: We apply *NN* to the instance of ATSP from Example 1, whose digraph is repeated in Figure 4.6.3. Starting from vertex 1, Algorithm *NN* moves to vertex 3, then to vertex 2, and on to vertex 4. The resulting tour is $\langle 1, 3, 2, 4, 1 \rangle$ of weight 30 (which is the worst tour).

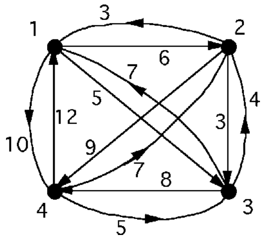


Figure 4.6.3 An instance of ASTP.

COMPUTATIONAL NOTE: Computational experiments in [JoGuMcYeZhZv02] indicate that, in fact, on most real-world problem instances of ATSP, *NN* performs better than *GR*; *GR* fails completely on one family of instances, where the average *GR*-tour is more than 2000% above the optimum. Computational experiments for STSP in [JoMc02] show that both *GR* and *NN* perform relatively well on Euclidean instances and perform poorly for general STSP. *GR* appears to perform better than *NN* for STSP.

Insertion Algorithms

Vertex insertion, another type of construction heuristic, applies to both Symmetric and Asymmetric TSP. For ATSP, the insertion algorithm begins with a cycle of length 2, and in each iteration, inserts a new vertex into the cycle. For STSP, the algorithm begins with a cycle of length 3. The algorithm descriptions and examples that follow are for ATSP, but with the obvious adjustments, they apply equally well to STSP.

DEFINITION

D10: Let $C = \langle i_1, i_2, \ldots, i_m, i_1 \rangle$ be the vertex sequence of a cycle in $\overset{\leftrightarrow}{K}_n$, and let v be a vertex not on C. For any arc (a, b) on cycle C, the ***insertion of vertex*** v ***at arc*** (a, b) is the operation of replacing arc (a, b) with the arcs (a, v) and (v, b) (see Figure 4.6.4). The resulting cycle is denoted $C(a, v, b)$. Thus, if $(a, b) = (i_k, i_{k+1})$, $1 \leq k \leq m - 1$, then $C(a, v, b) = \langle i_1, i_2, \ldots, a, v, b, \ldots i_m, i_1 \rangle$, and if $(a, b) = (i_m, i_1)$, then $C(a, v, b) = \langle i_1, i_2, \ldots i_{m-1}, a, v, b \rangle$.

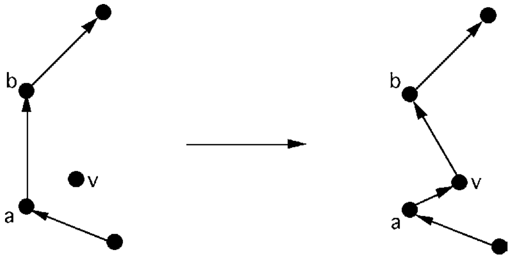


Figure 4.6.4 Insertion of vertex v at arc (a, b).

REMARK

R4: *Random vertex insertion*, *nearest vertex insertion*, and *farthest vertex insertion*, which are defined below, are three different versions of algorithm *VI*. Each is determined by how rule [*] chooses vertex i_s.

```
Algorithm 4.6.3:  Vertex Insertion (VI)

Input: $n \times n$ distance matrix $[d_{ij}]$.
Output: TSP tour $\langle i_1, i_2, \ldots, i_n, i_1 \rangle$.

    Let $i_1$ and $i_2$ be two vertices of $\overset{\leftrightarrow}{K}_n$, chosen by some rule.
    Initialize cycle $C = \langle i_1, i_2, i_1 \rangle$.
    For $s = 3, \ldots, n$
        Let $i_s$ be a vertex not on cycle $C$, chosen by some rule [*].
        Insert vertex $i_s$ at an arc $(a^*, b^*)$ of cycle $C = \langle i_1, i_2, \ldots, i_{s-1}, i_1 \rangle$
          such that the weight of $C(a^*, i_s, b^*)$ is minimum among the
          cycles $C(a, i_s, b)$ for all arcs $(a, b)$ in $C$.
        $C := C(a^*, i_s, b^*)$.
```

NOTATION: Given a vertex v and a cycle C in $\overset{\leftrightarrow}{K}_n$, $d(v, C)$ denotes the distance from v to C, that is, $d(v, C) = \min_{x \in V(C)} \{d_{vx}\}$.

DEFINITIONS

D11: The ***random vertex insertion (RVI)*** chooses vertex i_s randomly.

D12: The ***nearest vertex insertion (NVI)*** chooses vertex i_s so that its distance to cycle C is a minimum. That is, $d(i_s, C) = \min_{v \notin V(C)} \{d(v, C)\}$.

D13: The ***farthest vertex insertion (FVI)*** chooses vertex i_s so that its distance to cycle C is a maximum. That is, $d(i_s, C) = \max_{v \notin V(C)} \{d(v, C)\}$.

COMPUTATIONAL NOTE: The vertex insertion heuristics described above perform quite well for Euclidean TSP (see [JoMc02]). Computational experiments with *RVI* for ATSP in [GlGuYeZv01] show that *RVI* is good only for instances close to Euclidean.

Minimum Spanning Tree Heuristics

There are many more construction heuristics for TSP and especially for STSP, see [JoGuMcYeZhZv02,JoMc02]. For STSP, the heuristics that are often given in the literature include the *Double Minimum Spanning Tree (DMST)* and *Christofides* heuristics (see, e.g., Algorithms 6.4.2 and 6.4.3 in [GrYe99]). We will describe only the Christofides heuristic as it is significantly better than *DMST* from both theoretical and experimental points of view.

DEFINITION

D14: The operations to eliminate repeated vertices in the eulerian tour W in order to obtain an STSP tour are called ***shortcuts***.

COMPUTATIONAL NOTE: Implementing the ordinary shortcuts described in Algorithm *CH* below already produces a relatively good heuristic for the Euclidean TSP [JoMc02]. However, so-called greedy shortcuts (see [JoMc02]) result in a modification of *CH*, which seems to be one of the best construction heuristics for the Euclidean TSP.

```
Algorithm 4.6.4: Christofides Heuristic (CH)
Input: $n \times n$ distance matrix $[d_{ij}]$.
Output: STSP tour $\langle i_1, i_2, \ldots, i_n, i_1 \rangle$.
    Find a minimum spanning tree $T$ in $K_n$.
    Find a minimum-weight perfect matching $M$ in the subgraph of $K_n$
     induced on the odd-degree vertices of $T$.
    Create an eulerian subgraph $H$ of $K_n$ with edge set $E(H) = E(T) \cup M$.
    Construct an eulerian tour $W$ of $H$.
    Let $W = \langle i_1, i_2, \ldots, i_m, i_1 \rangle$ (written as a sequence of vertices).
    For $s = 3, \ldots, n$
        If $i_s = i_t$ for some $t < s$, then delete $i_{t+1}, \ldots, i_s$ from $W$.
```

FACT

F13: [JoPa85] Algorithm *CH* can be implemented to run in time $O(n^3)$.

Worst Case Analysis of Heuristics

While computational experiments are important in the evaluation of heuristics, they cannot cover all possible families of instances of TSP, and, in particular, they normally do not cover the most difficult instances. Moreover, certain applications may produce families of instances that are much harder than those normally used in computational experiments. For example, such instances can arise when Generalized TSP (discussed in subsection 4.5.5) is transformed into TSP. Thus, theoretical analysis of the worst possible cases is also important in evaluating and comparing TSP heuristics.

FACTS

F14: [GuYeZv02-a] For every $n \geq 3$, there is an instance of ATSP and an instance of STSP with n vertices satisfying the triangle inequality on which both *NN* and *GR* compute the unique worst possible tour.

F15: [Ru73] Let H be a tour produced by *RVI* for an instance I_n of STSP with $n \geq 3$ vertices. Then H is not worse than at least $(n-2)!$ tours when n is odd and $(n-2)!/2$ tours when n is even (including H itself).

F16: [GuYeZv02-a] For every $n \geq 2$, $n \neq 6$ and every instance of ATSP with n vertices, *RVI* computes a tour T that is not worse than at least $(n-2)!$ tours, including T itself.

F17: For STSP with triangle inequality, the *DMST* algorithm always produces a tour no more than twice as long as the optimal one, while *HC* produces tours never worse than 1.5 times the optimum (see [JoPa85]). However, there are instances for which *DMST* produces the unique worst possible tour, and there are instances for which *CH* produces a tour worse than all but at most $\lceil n/2 \rceil!$ tours [PuMaKa03].

REMARKS

R5: A simplified proof of Fact 14 can be found in [GuYeZv02-b]. It is based on a proof of a much more general result for the greedy algorithm in combinatorial optimization (see [GuYe02-a]).

R6: A proof of Fact 16 is similar to the proof of Fact 15, but is based on a different result that was first proved for n odd by V. I. Sarvanov [Sa76], and for n even by Gutin and Yeo [GuYe02-b]. The proofs use decompositions of K_n and $\overset{\leftrightarrow}{K}_n$, into hamiltonian cycles, which exist for K_n if and only if n is odd (see, e.g., [Ha69]), and for $\overset{\leftrightarrow}{K}_n$ if and only if $n \neq 4$ or 6 [Ti80].

4.6.4 Improvement Heuristics

Approximate algorithms based on improvement heuristics start from a tour (normally obtained using a construction heuristic) and iteratively improve it by changing some parts of it at each iteration. The best known tour improvement procedures are based on *edge exchange*, in which a tour is improved by replacing k its edges (arcs) with k edges (arcs) not in the solution.

COMPUTATIONAL NOTE: For many combinatorial optimization problems, well-known metaheuristics including *tabu search*, *simulated annealing* and *genetic algorithms* provide the best tools for producing good quality approximate solutions. This has not been the case for TSP, for which variations of the edge-exchange algorithms of Lin and Kernighan (*Lin-Kernighan local search*) are still state-of-the-art ([JoGuMcYeZhZv02], [JoMc02]). They are typically much faster than the exact algorithms, yet often produce solutions very close to the optimal one. Interested readers can find a detailed description of the Lin-Kernighan local search and its generalizations in [ReGl02]. Although the Lin-Kernighan local search can be applied only to STSP, ATSP can be transformed into STSP (see, e.g., [JoGuMcYeZhZv02]).

DEFINITIONS

D15: For STSP, the ***2-opt*** algorithm starts from an initial tour T and tries to improve T by replacing two of its non-adjacent edges with two other edges to form another tour (see Figure 4.6.5). Once an improvement is obtained, it becomes the new T. The procedure is repeated as long as an improvement is possible (or a time limit is exceeded).

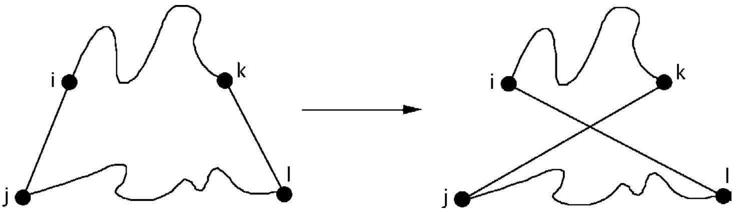


Figure 4.6.5 **Edges $\{i,j\}$ and $\{k,l\}$ are replaced by $\{i,l\}$ and $\{j,k\}$.**

D16: For $k \geq 3$, the ***k-opt*** algorithm is the same as 2-opt except that k edges (arcs) are replaced at each iteration.

D17: The ***best improvement k-opt*** is similar to the k-opt defined above; the difference is that every set of k arcs (edges) is tried for deletion from T and all possibilities of adding k arcs (edges) are considered before the *best* replacement of T is retained (as a replacement for T). The procedure is repeated.

FACT

F18: [PuMaKa03] Although best improvement 2-opt can take exponential time to find a local optimum, any (possibly sub-optimal) tour obtained after $O(n^3 \log n)$ iterations is not worse than $\frac{1}{n-1}$ of all STSP tours. Similar statements hold for the best improvement k-opt, $k \geq 3$.

Exponential Neighborhoods

Best improvement k-opt considers $\Theta(n^k)$ tours that can be obtained from a tour T by replacing edge-exchanges involving exactly k edges (arcs). Thus, it considers only a polynomial number of tours in the *neighborhood* of T. For TSP, one can construct various neighborhoods with an exponential number of tours in which the best tour can be found in polynomial time. In particular, there exist TSP neighborhoods of size $2^{\Theta(n \log n)}$ $(2^{\Theta(n)})$, where the best tour can be found in time $O(n^2)$ $(O(n))$. These neighborhoods are discussed in [AhErOrPu02], [DeWo00], and [GuYeZv02-b].

COMPUTATIONAL NOTE: While having seemingly strong theoretical properties, TSP exponential-neighborhood, local-search algorithms have not shown strong computation properties so far. Perhaps, this is due to the fact that it is not the size of the neighborhood that matters, but the total number of tours of TSP that are worse than the best tour in the neighborhood. This may explain why computational experiments show that some exponential-neighborhood, local-search heuristics are worse than the (seemingly much weaker) 3-opt.

4.6.5 The Generalized TSP

The *Generalized TSP* has numerous applications and is one of the most studied extensions of TSP [FiSaTo02].

DEFINITIONS

D18: The ***Generalized Asymmetric Traveling Salesman Problem (GATSP)***: Given a weighted complete digraph $\overleftrightarrow{K}_n$ and a partition $V_1, \ldots, V_k$ of its vertices, find a minimum-weight cycle containing exactly one (at least one) vertex from each set V_i, $i = 1, \ldots, k$.

D19: The sets V_i are called ***clusters***, and a cycle containing exactly one (at least one) vertex from each cluster is called a ***tour***.

D20: The ***Generalized Symmetric Traveling Salesman Problem (GSTSP)*** is formulated similarly with $\overleftrightarrow{K}_n$ replaced by K_n.

REMARK

R7: Observe that the requirements 'at least one' and 'exactly one' in GATSP and GSTSP coincide when the triangle inequality holds. The 'exactly one' versions of GATSP and GSTSP have received the most attention in the literature, and only these versions are discussed here.

Transforming Generalized TSP to TSP

One of the ways to solve instances of the Generalized TSP is to transform them into TSP instances. The most efficient transformations from GATSP to ATSP and from GSTSP to STSP appear to be the ones given in [NoBe93] and [LaSe99], respectively.

FACTS

F19: In the transformation of [NoBe93], from GATSP into ATSP, the number of vertices remains the same. Weights are modified so that an optimal ATSP tour must visit all the vertices that belong to the same cluster in the original problem before moving on to the next cluster. This is achieved by adding a large positive constant M to the weight of each inter-cluster arc. If the constant is large enough, an optimal tour will contain exactly k such heavy arcs, thus ensuring that no cluster is visited more than once.

F20: In the transformation of [LaSe99], from a GSTSP instance into an STSP instance, we first add a sufficiently large positive constant to every edge-weight, if needed, to ensure that all edge-weights are nonnegative. Then we consider each cluster V_i of cardinality at least 2. For each vertex v_i in such a cluster, we create a copy v_i'. In each such cluster, we form a hamiltonian cycle $C = \langle v_1, v_1', \ldots, v_t, v_t', v_1 \rangle$ and assign weight $-M$ to every edge of the form $v_i v_i'$ and weight $-2M$ to the rest of the edges in cycle C, where M is any constant larger than the sum of n heaviest edges in the GSTSP instance. The weights of the remaining edges within the clusters and between the clusters are inherited from the corresponding weights of the GSTSP instance. Clearly, an optimal tour T of the resulting STSP instance will use all edges of weight $-2M$, all but one of the $(-M)$-weight edges from each cluster of cardinality at least 2, and edges between the clusters. By contracting every vertex v_i and its copy v_i' in T, we obtain an optimal tour of the GSTSP instance.

F21: For the transformations in [LaSe99] and [NoBe93], there is a bijection between *optimal* tours in the original problem and those in the transformed one, making the transformations suitable for exact algorithms.

Exact Algorithms

FACTS

F22: Computational experiments ([BeGuPeYeZv03] and [LaSe99]) have shown that the transformations given in [NoBe93] and [LaSe99] can be used to solve to optimality small to moderate instances of Generalized TSP. However, even small instances require substantial computation because of the corresponding TSP instances' very large positive (negative) weights on some of its arcs (edges).

F23: A successful branch-and-cut algorithm for GSTSP is described and analyzed in [FiSaTo02], and a Lagrangian-based approach for GATSP is given in [NoBe91].

The next result appears to be a major stumbling block for using a standard branch-and-bound for ATSP adapted for GATSP.

F24: [GuYe03] Let $D = (V, A)$ be a digraph and let $V_1, V_2, \ldots, V_k$ be a partition of V. The problem of checking whether D has 1-regular subdigraph containing exactly one vertex from each $V_1, V_2, \ldots, V_k$ is NP-complete even if $|V_i| \leq 2$ for every $i = 1, 2, \ldots, k$. (A digraph H is 1-regular if the indegree and outdegree of every vertex in H equal 1.)

Approximate Algorithms

Although several TSP heuristics easily extend to the Generalized TSP, it seems better to first transform a Generalized TSP instance into a TSP instance, and then use heuristics on the TSP instance.

FACTS

F25: For the transformations in [LaSe99] and [NoBe93], not every tour in the transformed problem corresponds to a tour in the original problem, making the transformations less suitable for approximate algorithms.

F26: Two other transformations, described in detail in [FiSaTo02] and [BeGuPeYeZv03], are used in producing approximate (or exact) solutions. They apply to both GATSP and GSTSP. The description below is for an instance of GATSP with k clusters, $V_1, V_2, \ldots, V_k$.

TRANSFORMATION:
(1) Let G be a complete directed graph on k vertices, $v_1, \ldots, v_k$, such that the weight of arc (v_i, v_j) in G equals some function ϕ of the weights of the arcs in $\overset{\leftrightarrow}{K}_n$ from cluster V_i to cluster V_j. (In [FiSaTo02], the function ϕ is the minimum of these arc-weights, and in [BeGuPeYeZv03], ϕ is the average weight.)

(2) Let $\langle v_{i_1}, v_{i_2}, \ldots, v_{i_k}, v_{i_1}\rangle$ be an approximate (or exact) solution for the ATSP instance on G.

(3) Find a minimum-weight cycle C having exactly one vertex from each cluster and traversing the clusters in the order $V_{i_1}, V_{i_2}, \ldots, V_{i_k}$ by solving shortest path problems in $|V_1|$ acyclic digraphs.

COMPUTATIONAL NOTE: Computational experiments for GATSP in [BeGuPeYeZv03] indicate that the two different weight-functions ϕ produce solutions of comparable quality. However, theoretical analysis in [BeGuPeYeZv03] shows that the average-weight function appears to have better worst-case performance than the minimum-weight one.

4.6.6 The Vehicle Routing Problem

The *Vehicle Routing Problem (VRP)* was introduced by Dantzig and Ramser [DaRa59]. This problem (including its versions with additional constraints) seems to be the most applicable of all generalizations of TSP. Vehicle routing is the generic name given to a large family of problems involving the distribution of goods, information, services or people. A particularly important special case of VRP is that of minimizing the total distance traveled by a fleet of vehicles that deliver goods ordered by customers. The vehicles are assumed to have equal capacity Q, and their delivery tours start and end at a central depot.

DEFINITIONS

D21: Given a weighted complete directed or undirected graph on vertices $\{0, 1, \ldots, n\}$, a demand $d_i \geq 0$ for $i = 1, 2, \ldots, n$, and two parameters Q and k, a ***CVRP tour*** is a collection of k cycles $C_1, C_2, \ldots, C_k$, which contain all the vertices, pairwise intersect only in vertex 0, and satisfy $\sum_{i \in V(C_j)} d_i \leq Q$ for each $j = 1, 2, \ldots, k$.

D22: The ***Capacitated Vehicle Routing Problem (CVRP)***: Given a weighted complete directed or undirected graph on $n+1$ vertices, a demand $d_i \geq 0$ for $i = 1, 2, \ldots, n$, and two parameters, Q and k, find a CVRP tour for which the total weight of the cycles is minimum.

REMARKS

R8: Often, practitioners and researchers consider additional complicating constraints: the total weight of each cycle is limited, each customer (vertex) i must be visited within a prescribed *time window*, vehicles are allowed to have different capacities, routes of different vehicles cannot cross, etc. [PoKaWa99].

R9: In most research papers, the symmetric CVRP (on K_{n+1}) is considered. Nevertheless, the asymmetric (i.e., 'directed') CVRP version is also of interest [Vi96].

Exact Algorithms

FACTS

F27: The most efficient exact algorithms for symmetric CVRP are those based on branch-and-cut ([BlHo00], [NaRi01], [RaKoPuTr]).

F28: For the asymmetric version of CVRP it seems that the state-of-the-art exact algorithms still use branch-and-bound [ToVi01,ToVi02].

F29: Since CVRP has aspects of both TSP and *Bin Packing*, *set-covering* methods can sometimes be applied to CVRP and its generalizations with great success [BrSi01].

COMPUTATIONAL NOTE: The exact algorithms appear to be less powerful for CVRP than they are for TSP. Although they are able to solve some instances with 100 or more vertices, the exact algorithms were unable to solve an instance of symmetric CVRP with as few as 51 vertices [RaKoPuTr,ToVi02]. Often, practical versions of CVRP have various complicating constraints that cannot be tackled by exact algorithms. Thus, heuristics are of great importance for CVRP.

Heuristics for CVRP

CVRP heuristics fall roughly into two categories: those that produce a CVRP tour relatively quickly; and those that try to produce a near-optimal solution, using a substantial amount of computing if necessary. The latter kind are mostly metaheuristic-based algorithms. *Tabu search* seems to provide a good tradeoff between the quality of solution and running time ([ErOrSt], [GeLaPo01], [Ta93], [ToVi98]). Fast CVRP heuristics are of great importance, supplying quick and flexible solutions, good starting tours for metaheurisic-based algorithms, and upper bounds for exact algorithms. We close this section with brief descriptions of three classes of fast CVRP heuristics: *savings heuristics*, *insertion heuristics*, and *two-phase heuristics*.

REMARK

R10: The descriptions that follow are for the asymmetric CVRP, but they also apply to the symmetric CVRP with 'digraph' replaced by 'graph' and 'arc' replaced by 'edge'.

Savings Heuristics

The Clarke-Wright savings heuristic is perhaps the earliest [ClWr64] and best known heuristic for the VRP. Here, we describe a generic savings heuristic, whose concrete implementations may be found in [AlGa91], [ClWr64], and [LaSe01].

NOTATION: (a) For a vertex subset S, $t(S)$ denotes (an approximation of) the weight of an optimal TSP tour of the induced subdigraph on S.

(b) The total demand of a vertex subset S is $d(S) = \sum_{i \in S} d_i$.

DEFINITIONS

D23: A ***merge*** of cycles C_1 and C_2, denoted merge(C_1, C_2), is a cycle whose vertex set equals $V(C_1) \cup V(C_2)$. The resulting cycle is determined by some prescribed rule. Cycles C_1 and C_2 can be merged only if the total demand of their vertices does not exceed capacity Q (i.e., $d(V(C_1) \cup V(C_2)) \leq Q$).

D24: Given cycles C_1 and C_2, the ***saving*** of merge(C_1, C_2), denoted $s(C_1, C_2)$, is given by $s(C_1, C_2) = t(V(C_1)) + t(V(C_2)) - t(V(C_1) \cup V(C_2))$.

D25: Let $R = C_1, C_2, \ldots, C_m$ be a collection of m cycles of $\overset{\leftrightarrow}{K}_n$ whose pairwise intersections are vertex 0. The ***savings digraph***, $D(R)$, is the weighted digraph on m vertices, labeled $C_1, C_2, \ldots, C_m$, such that arc (C_i, C_j) exists if $d(V(C_1) \cup V(C_2)) \leq Q$, and the weight assigned to arc (C_i, C_j) is the saving $s(C_i, C_j)$.

REMARKS

R11: In the *Clarke-Wright savings algorithm* ([ClWr64],[LaSe01]), the weight of cycle C is used as an estimate of $t(V(C))$. To obtain the exact value of $t(V(C))$, one would have to solve a TSP on the induced subdigraph on $V(C)$, which may be too costly computationally.

R12: The simplest way to merge cycles C_1 and C_2 is the one used in the Clarke-Wright algorithm. If $(i, 0)$ is the arc in C_1 that enters vertex 0, and $(0, j)$ is the arc in C_2 that leaves vertex 0, then these two arcs are deleted and arc (i, j) is added to complete the new cycle. (See Figure 4.6.6.)

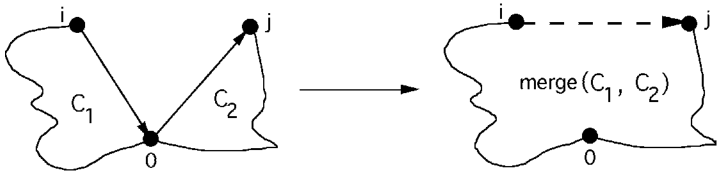


Figure 4.6.6 **A Clarke-Wright merge of cycles C_1 and C_2.**

REMARKS

R13: The easiest way to construct M is to simply choose a pair (C_a, C_b) with maximum saving $s(C_a, C_b)$ as the only arc in M. In some versions in which cycle-merging occurs in parallel, M is built in a greedy manner [LaSe01].

```
Algorithm 4.6.5: Savings Heuristic (SH)

Input: distance matrix $[d_{ij}]$, demands $d_i$, $i = 1, \dots, n$,
        capacity $Q$, and number of vehicles $k$.
Output: CVRP tour $R = \{C_1, \dots, C_k\}$.

    Initialize cycle count $m = n$ and cycles $C_i = \langle 0, i, 0 \rangle$, $i = 1, \dots, m$.
    Initialize $R = \{C_1, \dots, C_m\}$.
    While $m > k$
        Construct savings digraph $D(R)$.
        Construct a matching $M$ in $D(R)$ with $|M| \le m - k$.
        For each arc $(C_i, C_j)$ in $M$,
            $R := (R - \{C_i, C_j\}) \cup$ merge$(C_i, C_j)$
            $m := m - 1$
```

R14: Often, after R is produced by algorithm *SH*, each of the cycles in R is improved by some TSP improvement heuristic (see Section 4.6.4). For example, in [LaSe01], a CVRP tour found by the Clarke-Wright algorithm is improved by applying the best improvement 3-opt to each of its cycles.

Insertion Heuristics

In CVRP iteration algorithms [LaSe01], we start from k cycles of the form $C_p = \langle 0, i_p, 0 \rangle$. The vertices outside of the cycles are inserted one by one in sequential or parallel manner. The word 'parallel' here means that a vertex is inserted in one of the current cycles C_p for which the insertion is most beneficial. In the sequential mode, we start constructing a new cycle only when the previous one cannot be increased because of the capacity constraints. The cost of insertion of a vertex i into a current cycle C_p can be measured by $t(V(C) \cup \{i\}) - t(V(C))$.

REMARKS

R15: An example of such an algorithm is the Christofides-Mingozzi-Toth insertion heuristic [ChMiTo79].

R16: Fisher and Jaikumar [FiJa81] suggested inserting all vertices at the same time. They apply the *Generalized Assignment Problem* to find 'optimal' insertions that do not violate the capacity constraints.

Two-phase Heuristics

The basic idea of *two-phase heuristics* is to partition vertices $\{1, 2, \dots, n\}$ into k clusters $V_1, \dots, V_k$ and solve TSP for each of the graphs induced by $V_i \cup \{0\}$, $i = 1, 2, \dots, k$. Wren and Holliday [WrHo72] suggested a sweeping technique for the Euclidean CVRP, in which the depot 0 and vertices i, $1 \le i \le n$ are points on the Euclidean plane, and d_{ij} is the Euclidean distance between i and j.

We introduce a polar coordinate system, in which one of the n vertices, say i, is chosen as the reference point, with polar coordinates $(0, d_{0i})$. Then every vertex $j \neq i$ has coordinates (ϕ_j, d_{0j}), where ϕ_j is the angle between the rays from 0 to i and from 0 to j. The sweeping algorithm in its simplest form is as follows.

Algorithm 4.6.6: Sweeping Heuristic

```
Input: distance matrix $[d_{ij}]$, polar angles $\phi_i$ for $1 \le i \le n$,
        demands $d_i$, $i = 1, \ldots, n$, capacity $Q$, and number of vehicles $k$.
Output: CVRP tour consisting of cycles $C_j$, $j = 1, \ldots, k$.

    Sort the vertices $i_1, i_2, \ldots, i_n$ such that $\phi_{i_s} \le \phi_{i_{s+1}}$, $s = 1, \ldots, n-1$.
    Initialize $S_j = \emptyset$, $j = 1, \ldots, k$.
    Set $j = 1$.
    For $s = 1, \ldots, n$
        If $d(S_j \cup \{i_s\}) > Q$
            $j := j + 1$
        $S_j := S_j \cup \{i_s\}$
    For $j = 1, \ldots, k$
        Let $C_j$ be a TSP tour for the subgraph induced on $S_j \cup \{0\}$.
```

REMARK

R17: An extension of this approach to the general CVRP is described in [BrSi95]. Another example of a two-phase heuristic is a truncated branch-and-bound provided in [CHMiTo79].

References

[AhErOrPu02] R. K. Ahuja, O. Ergun, J. B. Orlin and A. P. Punnen, A survey of very large-scale neighborhood search techniques, *Discrete Appl. Math.* 123 (2002), 75–102.

[AlGa91] K. Altinkemer and B. Gavish, Parallel savings based heuristics for the delivery problem, *Oper. Res.* 39 (1991), 456–469.

[ApBiChCo98] D. Applegate, R. E. Bixby, V. Chvátal, and W. Cook, On the Solution of Traveling Salesman Problem, *Doc. Math., J. DMV, Extra Vol. ICM Berlin 1998, III* (1998) 645–656. The 12/15/1999 release of the `Concorde` code is currently available from `http://www.math.princeton.edu/tsp/concorde.html`

[Ar98] S. Arora, Polynomial-time approximation schemes for Euclidean TSP and other geometric problems, *JACM* 45 (1998), 753–782. Preliminary version in Proc. 37th IEEE Symp. Found. Comput. Sci., 1996, and Proc. 38th IEEE Symp. Found. Comput. Sci., 1997.

[Ar02] S. Arora, Approximation algorithms for geometric TSP, In *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[BaTo85] E. Balas and P. Toth, Branch and Bound Methods, In *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Eds.), Wiley, 1985.

[BeLu00] A. Belloni and A. Lucena, A relax and cut algorithm for the traveling salesman problem, *A talk at Intern. Symp. Math. Prog., Atlanta*, 2000.

[BeGuPeYeZv03] D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo, and A. Zverovitch, Transformations of Generalized ATSP into ATSP: experimental and theoretical study, *Oper. Res. Lett.* 31 (2003), 357–365.

[BlHo00] U. Blasum and W. Hochstättler, Application of the brach and cut method to the vehicle routing problem, *Zentrum für Angewandte Informatik*, Köln, Tech. Report zrp2000–386.

[BrSi95] J. Bramel and D. Simchi-Levi, A location based heuristic for general routing problems, *Oper. Res.* 43 (1995), 649–660.

[BrSi01] J. Bramel and D. Simchi-Levi, Set-Covering Based Algorithms for the Capacitated VRP, in *The Vehicle Routing Problem* (P. Toth and D. Vigo, Eds.), SIAM, 2001.

[ClWr64] G. Clarke and J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Oper. Res.* 12 (1964), 568–581.

[ChMiTo79] N. Christofides, A. Mingozzi, and P. Toth, The Vehicle Routing Problem, In *Combinatorial Optimization* (A. Mingozzi, P. Toth, and C. Sandi, Eds.), Wiley, 1979.

[DaFuJo54] G. B. Dantzig, D. R. Fulkerson and S. M. Johnson, Solution of large scale traveling salesman problem, *Oper. Res.* 2 (1954), 393–410.

[DaRa59] G. B. Dantzig and R. H. Ramser, The truck dispatching problem, *Manag. Sci.* 6 (1959), 80–91.

[DeWo00] V. G. Deineko and G. J. Woeginger, A study of exponential neighbourhoods for the traveling salesman problem and the quadratic assignment problem, *Math. Program., Ser. A* 87 (2000), 519–542.

[ErOrSt] O. Ergun, J. B. Orlin, and A. Steele-Feldman, Creating very large scale neighborhoods out of smaller ones by compounding moves: a study on the vehicle routing problem, Manuscript, 2002.

[FiLoTo02] M. Fischetti, A. Lodi, and P. Toth, Exact Methods for the Asymmetric Traveling Salesman Problem, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[FiSaTo02] M. Fischetti, J. J. Salazar-González, and P. Toth, The Generalized Traveling Salesman and Orienteering Problem, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[FiJa81] M. L. Fisher and R. Jaikumar, A generalized assignment heuristic for vehicle routing, *Networks* 11 (1981) 109–124.

[GaGrJo76] M. R. Garey, R. L. Graham, and D. S. Johnson, Some NP-complete geometric problems, in *Proc. 8th ACM Symp. Theory Comput.* (1976), 10–22.

[GeLaPo01] M. Gendreau, G. Laporte, and J. Y. Potvin, Metaheuristics for the Capacitated VRP, In *The Vehicle Routing Problem* (P. Toth and D. Vigo, Eds.), SIAM, 2001.

[GlGuYeZv01] F. Glover, G. Gutin, A. Yeo, and A. Zverovich, Construction heuristics for the asymmetric TSP, *Eur. J. Oper. Res.* 129 (2001), 555–568.

[GoSt85] B. L. Golden and W. R. Stewart, Empirical Analysis of Heuristics, in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. L. Lawler, J. K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Eds.), Wiley, 1985.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[GuYe02-a] G. Gutin and A. Yeo, Anti-matroids, *Oper. Res. Lett.* 30 (2002) 97–99.

[GuYe02-b] G. Gutin and A. Yeo, Polynomial approximation algorithms for the TSP and the QAP with factorial domination number, *Discrete Appl. Math.* 119 (2002), 107–116.

[GuYe03] G. Gutin and A. Yeo, Assignment Problem based algorithms are impractical for the Generalized TSP, *Ausralasian J. Combinatorics* 27 (2003), 149–154.

[GuYeZv02-a] G. Gutin, A. Yeo, and A. Zverovich, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP, *Discrete Appl. Math.* 117 (2002), 81–86.

[GuYeZv02-b] G. Gutin, A. Yeo, and A. Zverovitch, Exponential Neighborhoods and Domination Analysis for the TSP, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[HaKh01] R. Hassin and S. Khuller, z-Approximations, *J. Algorithms* 41 (2001), 429–442.

[Ha69] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[HoWo85] A. J. Hoffman and P. Wolfe, History, in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. L. Lawler, J. K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Eds.), Wiley, 1985.

[JoGuMcYeZhZv02] D. S. Johnson, G. Gutin, L. A. McGeoch, A. Yeo, W. Zhang, and A. Zverovitch, Experimental Analysis of Heuristics for ATSP, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, Dordrecht, 2002.

[JoMc02] D. S. Johnson and L. A. McGeoch, Experimental Analysis of Heuristics for STSP, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[JoPa85] D. S. Johnson and C. H. Papadimitriou, Performance guarantees for heuristics, in *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization* (E. L. Lawler, J. K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Eds.), Wiley, 1985.

[LaSe99] G. Laporte and F. Semet, Computational evaluation of a transformation procedure for the symmetric generalized traveling salesman problem, *INFOR* 37 (1999), 114–120.

[LaSe01] G. Laporte and F. Semet, Classical Heuristics for the Capacitated VRP, in *The Vehicle Routing Problem* (P. Toth and D. Vigo, Eds.), SIAM, 2001.

[LoPu02] A. Lodi and A. P. Punnen, TSP Software. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[Me32] K. Menger, Das botenproblem, *Ergebnisse Eines Mathematischen Kolloquiums* 2 (1932), 11–12.

[Mi99] J. C. B. Mitchell, Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial time approximation scheme for geometric TSP, k-MST, and related problem, *SIAM J. Comput.* 28 (1999), 1298–1309.

[Na02] D. Naddef, Polyhedral Theory and Branch-and-Cut Algorithms for the Symmetric TSP, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[NaRi01] D. Naddef and G. Rinaldi, Branch-and-Cut Algorithms for the Capacitated VRP, in *The Vehicle Routing Problem* (P. Toth and D. Vigo, Eds.), SIAM, 2001.

[NoBe91] C. E. Noon and J. C. Bean, A Lagrangian based approach for the asymmetric generalized traveling salesman problem. *Oper. Res.* 39 (1991), 623–632.

[NoBe93] C. E. Noon and J. C. Bean, An efficient transformation of the generalized traveling salesman problem, *INFOR* 31 (1993), 39–44.

[Pa77] C. H. Papadimitriou, The Euclidean traveling salesman problem is NP-complete, *Theoret. Comput. Sci.* 4 (1977), 237–244.

[PaSt82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982.

[PoKaWa99] A. Poot, G. Kant, and A. P. M. Wagelmans, A savings based method for real-life vehicle routing problems, Erasmus University Rotterdam, Econometric Institute Report Ei 9938/A.

[Pu02] A. P. Punnen, The Traveling Salesman Problem: Aplications, Formulations and Variations, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[PuMaKa03] A. P. Punnen, F. Margot, and S. Kabadi, TSP heuristics: domination analysis and complexity, *Algorithmica* 35 (2003), 111–127.

[RaKoPuTr] T. K. Ralphs, L. Kopman, W. R. Pulletblank, and L. E. Trotter, On the capacitated vehicle routing problem, to appear in *Math. Program.*

[RaSm98] S. Rao and W. Smith, Approximating geometric graphs via "spanners" and "banyans", *Proc. 30th Ann. ACM Symp. Theory Comput.* (1998), 540–550.

[ReGl02] C. Rego and F. Glover, Local Search and Metaheuristics, in *The Traveling Salesman Problem and its Variations* (G. Gutin and A. P. Punnen, Eds.), Kluwer, 2002.

[Ru73] V. I. Rublineckii, Estimates of the accuracy of procedures in the Traveling Salesman Problem, *Numerical Mathematics and Computer Technology* no. 4 (1973), 18–23 (in Russian).

[SaGo76] S. Sahni and T. Gonzalez, P-complete approximation problems, *JACM* 23 (1976), 555–565.

[Sa76] V. I. Sarvanov, On the minimization of a linear form on a set of all n-elements cycles, *Vestsi Akad. Navuk BSSR, Ser. Fiz.-Mat. Navuk* no. 4 (1976), 17–21 (in Russian).

[Ta93] E. Taillard, Parallel iterative search methods for vehicle routing problem, *Networks* 23 (1993), 661–673.

[Ti80] T. W. Tillson, A Hamiltonian decomposition of K^*_{2m}, $2m \geq 8$, *J. Combin. Theory, Ser. B* 29 (1980), 68–74.

[ToVi01] P. Toth and D. Vigo, Branch-and-Bound Algorithms for the Capacitated VRP, In *The Vehicle Routing Problem* (P. Toth and D. Vigo, Eds.), SIAM, 2001.

[ToVi02] P. Toth and D. Vigo, Models, relaxations and exact approaches for the capacitated vehicle routing problem, *Discrete Appl. Math.* 123 (2002), 487–512.

[Tr97] L. Trevisan, When Hamming meets Euclid: the approximability of geometric TSP and MST, *Proc. 29th ACM Symp. Theory Comput.* (1997), 21–39.

[Vi96] D. Vigo, A heuristic algorithm for the asymmetric capacitated vehicle routing problem, *Europ. J. Oper. Res.* 89 (1996), 108–126.

[Wo98] L. A. Wolsey, *Integer Programming*, Wiley, 1998.

[WrHo72] A. Wren and A. Holliday, Computer scheduling of vehicles from one or more depots to a number of delivery points, *Oper. Res. Quart.* 23 (1972), 333–344.

[Ze81] E. Zemel, Measuring the quality of approximate solutions to zero-one programming problems, *Math. Oper. Res.* 6 (1981), 319–332.

4.7 FURTHER TOPICS IN CONNECTIVITY

Josep Fàbrega, Technical University of Catalonia, Spain
Miguel Angel Fiol, Technical University of Catalonia, Spain

4.7.1 High Connectivity
4.7.2 Bounded Connectivity
4.7.3 Symmetry and Regularity
4.7.4 Generalizations of the Connectivity Parameters
References

Introduction

Continuing the study of connectivity, initiated in §4.1 of the *Handbook*, we survey here some (sufficient) conditions under which a graph or digraph has a given connectivity or edge-connectivity. First, we describe results concerning maximal (vertex- or edge-) connectivity. Next, we deal with conditions for having (usually lower) bounds for the connectivity parameters. Finally, some other general connectivity measures, such as one instance of the so-called "conditional connectivity", are considered.

For unexplained terminology concerning connectivity, see §4.1.

4.7.1 High Connectivity

Since connectivity has to do with "connection", intuitively we can expect to find high connectivity when the "edge density" is large. Different situations in which this seems to be the case are:

(a) Vertices with high degrees.

(b) Small diameter (for given girth).

(c) Small number of vertices (for given degree and girth).

(d) Large number of vertices (for given degree and diameter).

The results in this subsection give several conditions of the above types, under which maximum vertex- or edge-connectivity is attained.

Minimum Degree and Diameter

NOTATION: Let $G = (V, E)$ be a graph with order n, minimum degree δ, edge-connectivity λ, and (vertex-)connectivity κ. In some other sections of the *Handbook*, the notation δ_{min}, δ_{max}, κ_e, and κ_v are used instead of δ, Δ, λ, and κ, respectively.

DEFINITIONS

D1: The ***girth*** of a graph with a cycle is the length of its shortest cycle. An acyclic graph has infinite girth.

D2: The ***diameter*** of graph G is $\max_{u,v \in V} \{dist_G(u,v)\}$.

D3: The ***clique number*** of a graph G, denoted $\omega(G)$, is the maximum number of vertices in a complete subgraph of G.

D4: A (di)graph G is ***p-partite*** if its vertex-set can be partitioned into p independent (or stable) sets.

FACTS

F1: [Ch66] If $\delta \geq \lfloor n/2 \rfloor$, then the edge-connectivity λ equals the minimum degree δ.

F2: [Le74] If for any non-adjacent vertices u and v, $deg(u) + deg(v) \geq n - 1$, then $\lambda = \delta$.

F3: [Pl75] If G is a graph with diameter $D = 2$, then $\lambda = \delta$.

F4: [Vo88] If G is bipartite and $\delta \geq \lfloor n/4 \rfloor + 1$, then $\lambda = \delta$.

F5: [Vo89] If G is p-partite ($p \geq 2$) and $n \leq 2\lfloor \frac{p}{p-1}\delta \rfloor - 1$, then $\lambda = \delta$.

F6: [ToVo93] If G is p-partite ($p \geq 2$) and $\delta \geq n \, \frac{2p-3}{2p-1}$, then $\kappa = \delta$.

F7: [DaVo95] If G has clique number $\omega \leq p$ and $n \leq 2\lfloor \frac{p}{p-1}\delta \rfloor - 1$, then $\lambda = \delta$.

REMARKS

R1: As is easily shown, Fact 3 $\Rightarrow$ Fact 2 $\Rightarrow$ Fact 1.

R2: Fact 4 is a slight improvement of Fact 5 for $p = 2$.

R3: In addition to Fact 7, the authors in [DaVo95] gave other sufficient conditions for equality of edge-connectivity and minimum degree of graphs, which mostly generalize conditions in [PlZn89].

Degree Sequence

NOTATION: For the next group of results, G is an n-vertex graph with degree sequence $d_1 \geq d_2 \geq \cdots \geq d_n = \delta$. For a vertex u, $N(u)$ denotes the set of vertices adjacent to u.

FACTS

F8: [GoWh78] If the vertex set of G can be partitioned into $\lfloor n/2 \rfloor$ pairs of vertices (u_i, v_i) (and, if n is odd, one "unpaired" vertex w) such that $deg(u_i) + deg(v_i) \geq n$, $i = 1, 2, \ldots, \lfloor n/2 \rfloor$, then the edge-connectivity $\lambda = \delta$.

F9: [GoEn79] If each vertex u of minimum degree satisfies

$$\sum_{v \in N(u)} deg(v) \geq \begin{cases} \lfloor n/2 \rfloor^2 - \lfloor n/2 \rfloor, & \text{for even } n \text{ or odd } n \leq 15 \\ \\ \lfloor n/2 \rfloor^2 - 7, & \text{for odd } n \geq 15 \end{cases}$$

then $\lambda = \delta$.

F10: [Bo79] Let G be a graph with order $n \geq 2$. If its degree sequence $d_1 \geq d_2 \geq \cdots \geq d_n = \delta$ satisfies $\sum_{i=1}^{k}(d_i + d_{n-i}) \geq kn - 1$ for all k with $1 \leq k \leq \min\{\lfloor n/2 \rfloor - 1, \delta\}$, then $\lambda = \delta$.

F11: [DaVo97] If $\delta \geq \lfloor n/2 \rfloor$ or if $\delta \leq \lfloor n/2 \rfloor - 1$ and $\sum_{i=1}^{k}(d_i + d_{n+i-\delta-1}) \geq k(n-2) + 2\delta - 1$ for some k with $1 \leq k \leq \delta$, then $\lambda = \delta$.

F12: [Vo03] Suppose that G has order $n \geq 6$ with clique number $\omega \leq p$. Let $\nu = 1$ when n is even and $\nu = 0$ when n is odd. If $\delta \geq \lfloor n/2 \rfloor$ or if $\delta \leq \lfloor n/2 \rfloor - 1$ and $\sum_{i=1}^{\delta+1} d_{n+1-i} \geq (\delta+1)\frac{p-1}{p}\frac{n+1+\nu}{2} - \frac{2\delta+2}{p(n-3+\nu)}$, then $\lambda = \delta$.

REMARKS

R4: Note that Fact 8 implies Fact 1 only when n is even. Fact 9 also implies Fact 1. Moreover, as shown by the examples in [PlZn89], Fact 9 is independent of Fact 2 and Fact 3.

R5: Fact 10 implies Fact 1 when n is even, but in general, as shown in [PlZn89], it is independent of Facts 1, 2, 3, and 9.

R6: Fact 11 is even valid for digraphs, and a theorem of Xu [Xu94] follows easily (see Fact 22). As easily shown, Fact 11 implies Fact 10.

R7: Fact 12 generalizes results in [Vo88], [Vo89], as well as Fact 7. Furthermore, as shown in [HeVo03b], the conditions in Fact 12 also guarantee maximum local edge-connectivity for all pairs u and v of vertices in G; that is, $\lambda(u{-}v) = \min\{deg(u), deg(v)\}$.

Distance

DEFINITIONS

D5: The ***distance*** $dist_G(U_1, U_2)$ between two given subsets $U_1, U_2 \subset V(G)$ is the minimum of the distances $dist_G(u_1, u_2)$ for all vertices $u_1 \in U_1$ and $u_2 \in U_2$. (When there is no ambiguity, we omit the subscript G.)

D6: The ***line graph*** LG of a graph G has vertices representing the edges of G, and two vertices are adjacent if and only if the corresponding edges are adjacent (that is, they have one endpoint in common).

FACTS

F13: Let u_1v_1 and u_2v_2 be edges in a graph G, and let $U_i = \{u_i, v_i\}$, $i = 1, 2$. Then, the distance between the corresponding vertices of LG satisfies $d_{LG}(u_1v_1, u_2v_2) = d_G(U_1, U_2) + 1$ and thus, the diameters of LG and G satisfy $D(LG) \leq D(G) + 1$.

F14: [PlZn89] Let G be a connected graph such that every pair of vertex subsets U_1, U_2 of cardinality two satisfies $dist(U_1, U_2) \leq 2$. Then G is maximally edge-connected ($\lambda = \delta$).

F15: [BaCaFaFi96] Let G be a graph with minimum degree δ and line graph LG. Then,

(a) If LG has diameter at most three, then G is maximally edge-connected ($\lambda = \delta$).

(b) If LG has diameter two, then G is maximally connected ($\kappa = \delta$).

REMARKS

R8: The sufficient condition given in Fact 14 relaxes slightly the one given in Fact 3. Furthermore, it suffices to require such a condition on the 2-element subsets that are the endpoints of some edge, as shown in Fact 15(a).

R9: From the above remark, Fact 15(a) generalizes both Fact 14 and Fact 3 (Plesnik's result).

Super Edge-Connectivity

Here we consider a stronger measure of edge-connectivity.

DEFINITION

D7: A maximally edge-connected graph is ***super-λ*** if every minimum edge-disconnecting set is ***trivial***; that is, consists of the edges incident on a vertex of minimum degree.

EXAMPLE

E1: Figure 4.7.1 shows a 3-regular maximally edge-connected graph that is not super-λ. The set $\{e, f, g\}$ is a non-trivial minimum edge-disconnecting set.

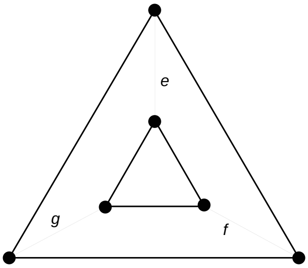


Figure 4.7.1 G **is maximally edge-connected but not super-**λ**.**

FACTS

F16: [Le74] Let $G \neq K_{n/2} \times K_2$. If for any non-adjacent vertices u and v, $deg(u) + deg(v) \geq n$, then G is super-λ.

F17: If for any non-adjacent vertices u and v, $deg(u) + deg(v) \geq n + 1$, then G is super-λ.

F18: [Ke72] If $\delta \geq \lfloor n/2 \rfloor + 1$, then G is super-λ.

F19: [Fi92] If G has diameter two and contains no complete subgraph H on δ vertices with $\deg_G(v) = \delta$ for all $v \in V(H)$, then G is super-λ.

F20: [So92] Let G be a graph with maximum degree Δ. If $n > 2\delta + \Delta - 1$, then G is super-λ.

REMARKS

R10: Facts 17 and 18, which are analogues of Facts 2 and 1, are direct consequences of Fact 16.

R11: Fact 19 can be seen as a refinement of Fact 3 (where only the diameter condition is required) and has Fact 20 as a corollary.

Digraphs

As mentioned in §4.1, since the connectivity parameters of a graph G equal those of its *symmetric digraph* $\overset{\leftrightarrow}{G}$ (obtained by replacing each edge of G by a digon), many of the previous results can be generalized to the directed case.

DEFINITIONS

D8: The ***vertex-connectivity*** of a digraph G, denoted $\kappa(G)$, is the minimum size of a vertex subset whose deletion results in a non-strongly connected digraph.

D9: The ***edge-connectivity*** of a digraph G, denoted $\lambda(G)$, is the minimum size of an edge subset whose deletion results in a non-strongly connected digraph.

NOTATION: (a) For a vertex $u \in V(G)$, $deg^+(u)$ denotes the *out-degree*, the number of vertices adjacent *from* vertex u, and $deg^-(u)$ denotes the *in-degree*, the number of vertices adjacent *to* vertex u; $\delta(u) = \min\{deg^+(u), deg^-(u)\}$.
(b) $\delta^+ = \min_{u \in V}\{deg^+(u)\}$ and $\delta^- = \min_{u \in V}\{deg^-(u)\}$.
(c) $\delta = \min_{u \in V} \delta(u) = \min\{\delta^+, \delta^-\}$.

Similar notations with Δ stand for maximum degrees.

NOTATION: For vertices $u, v \in V(G)$, $\lambda(u - v)$ denotes the maximum number edge-disjoint u-v directed paths.

FACTS

F21: [Jo72] If G is a digraph with diameter $D = 2$, then $\lambda = \delta$.

F22: [Xu94] Let G be a digraph of order n. If there are $\lfloor n/2 \rfloor$ pairs of (different) vertices (u_i, v_i) such that $\delta(u_i) + \delta(v_i) \geq n$, $i = 1, 2, \ldots, \lfloor n/2 \rfloor$, then $\lambda = \delta$.

F23: [HeVo03b] Let G be a digraph with diameter at most two. Then, $\lambda(u{-}v) = \min\{deg^+(u), deg^-(v)\}$ for all pairs u and v of vertices in G.

F24: [HeVo03a] Let G be a strongly connected digraph with edge-connectivity λ and minimum degree δ. If for all maximal pairs of vertex sets X and Y at distance 3 there exists an isolated vertex in the induced subgraph on $X \cup Y$, then $\lambda = \delta$.

F25: [HeVo03b] Let G be a p-partite digraph of order n and minimum degree δ with $p \geq 2$. If $n \leq 2\lfloor (p\delta)/(p-1) \rfloor - 1$, then $\lambda(u{-}v) = \min\{deg^+(u), deg^-(v)\}$ for all pairs u and v of vertices in G.

F26: [HeVo03b] Let G be a bipartite digraph of order n and minimum degree $\delta \geq 2$ with the bipartition $V' \cup V''$. If $deg(x) + deg(y) \geq (n+1)/2$ for each pair of vertices $x, y \in V'$ and each pair of vertices $x, y \in V''$, then $\lambda(u{-}v) = \min\{deg^+(u), deg^-(v)\}$ for all pairs u and v of vertices in G.

REMARKS

R12: Notice that Plesník's result (Fact 3) is, in fact, a consequence of the older result of Jolivet (Fact 21). Similarly, Fact 22 generalizes Fact 8.

R13: Fact 22 was improved by Dankelmann and Volkmann in two subsequent papers [DaVo97,DaVo00], where the bipartite case was also considered.

R14: A restatement of Fact 23 states that a digraph with diameter two has maximum local edge-connectivity. Moreover, this obviously implies Jolivet's result (Fact 21) and the corresponding local connectivity result for undirected graphs, proved in [FrOeSw00].

R15: A consequence of Fact 24 is the directed version of Fact 14.

Oriented Graphs

DEFINITIONS

D10: A digraph is ***super-λ*** if every minimum edge-disconnecting set consists of the edges directed to or from a vertex with minimum degree. A digraph is ***super-κ*** if every minimum disconnecting set consists of the vertices adjacent to or from a vertex with minimum degree.

D11: An ***oriented graph*** G (also called an ***antisymmetric digraph***) is a digraph such that between any two vertices u, v, there is at most one (directed) edge ((u, v) or (v, u)).

EXAMPLE

E2: Figure 4.7.2 shows a 2-regular maximally connected digraph G that is not super-κ. If $F = \{x, y\}$, then $G - F$ is not strongly connected (for instance, there is no [directed] path in $G - F$ from u to v) and F is non-trivial (it does not consist of the vertices adjacent to or from a vertex with minimum degree).

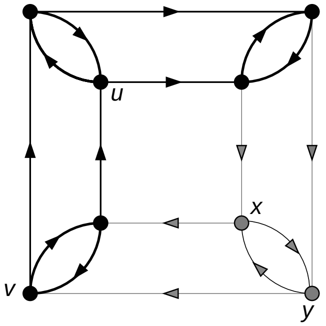


Figure 4.7.2 G **is maximally connected but not super-κ.**

FACTS

F27: [AyFr70] Let G be an oriented graph with n vertices and minimum degree δ. If $\delta \geq \lfloor (n+2)/4 \rfloor$, then $\lambda = \delta$.

F28: [Fi92] If G is an oriented graph with n vertices and minimum degree $\delta \geq \lfloor n/4 \rfloor + 1$, then G is super-λ.

F29: [Fi92] If G is an oriented graph with diameter two, then G is super-λ.

REMARKS

R16: Facts 27 and 28 are analogues of Fact 1, whereas Fact 29, similar to Fact 21, is a consequence of Fact 19.

R17: In fact, the sufficient conditions given in [AyFr70] and [Fi92] (Facts 27 and 28) were $\delta^+ + \delta^- \geq \lfloor n/2 \rfloor$ and $\delta^+ + \delta^- \geq \lfloor n/2 \rfloor + 1$, respectively. Furthermore, it is easily shown that Facts 28 and 29 do not imply each other.

R18: Higher connectivity in *tournaments*, which are oriented complete graphs, is discussed in §3.3 of the *Handbook*.

Semigirth

To generalize Jolivet's result (Fact 21) and give new results on superconnectivity, it is relevant to consider a new parameter related to the path structure of the digraph. In our context, this parameter plays a role similar (and is tightly related) to the girth of a graph.

DEFINITIONS

D12: [FaFi89,FiFaEs90] For a given digraph $G = (V, E)$ with diameter D, the ***semigirth***, denoted $\ell(G)$, is the greatest integer ℓ between 1 and D such that for any $u, v \in V$,

(a) if $dist(u, v) < \ell$, the shortest u-v directed walk is unique and there are no u-v directed walks of length $dist(u, v) + 1$.

(b) if $dist(u, v) = \ell$, there is only one shortest u-v directed walk.

D13: A digraph G is a ***generalized p-cycle*** when it has its vertex set partitioned in p parts cyclically ordered, and vertices in one part are adjacent only to vertices in the next part. Thus, a generalized 2-cycle is the same as a bipartite digraph.

EXAMPLE

E3: Figure 4.7.3 shows a 2-regular digraph for which the semigirth ℓ is equal to its diameter, namely, $\ell = D = 3$.

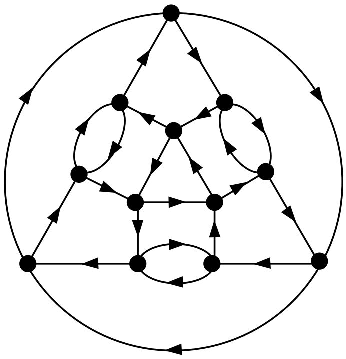


Figure 4.7.3 **Semigirth $\ell = D = 3$.**

FACTS

F30: [FaFi89] Let G be a digraph with minimum degree $\delta > 1$, diameter D, semigirth ℓ, and connectivities κ and λ.

(a) If $D \leq 2\ell$, then $\lambda = \delta$.

(b) If $D \leq 2\ell - 1$, then G is super-λ and $\kappa = \delta$.

(c) If $D \leq 2\ell - 2$, then G is super-κ.

F31: [FaFi96a,PeBaGo01] Let G be a generalized p-cycle ($p \geq 2$).

(a) If $D \leq 2\ell + p - 1$, then $\lambda = \delta$.

(b) If $D \leq 2\ell + p - 2$, then G is super-λ and $\kappa = \delta$.

(c) If $D \leq 2\ell + p - 3$, then G is super-κ.

F32: Any bipartite digraph with diameter three is maximally edge-connected.

REMARKS

R19: The main idea in the proof of the results in Fact 30 is that semigirth ℓ measures how far away one can move from or to a given subset F of vertices. For instance, in proving (a), it is shown that if $|F| < \delta$, in any connected component of $G - F$ there are vertices u, v such that $dist(u, F), dist(F, v) \geq \ell$. Hence, any shortest path of length at most $2\ell - 1$ cannot contain a vertex of F. As a conclusion, F cannot be a disconnecting set.

R20: Since any digraph G has semigirth $\ell \geq 1$, Fact 21 is included in Fact 30(a).

R21: Fact 32 is the analogue for bipartite digraphs of Jolivet's result (Fact 21). In fact, for a bipartite (di)graph, the condition $\delta \geq \lfloor n/4 \rfloor + 1$ implies $D \leq 3$, so that Fact 32 can be also seen as a generalization of Fact 4.

Line Digraphs

DEFINITION

D14: The ***line digraph*** LG of a digraph G, denoted LG, has $V(LG) \equiv E(G)$, and a vertex (u, v) is adjacent to a vertex (w, z) if $v = w$ (that is, the *head* of edge (u, v) is the *tail* of edge (w, z) in digraph G). The ***k-iterated line digraph***, L^kG, is defined recursively by $L^kG = LL^{k-1}G$.

FACTS

F33: The order of LG equals the size of G, $|V(LG)| = |E(G)|$, and their minimum degrees coincide, $\delta(LG) = \delta(G) = \delta$. Moreover, $\kappa(LG) = \lambda(G)$.

F34: If G is d-regular, $d > 1$, has order n, diameter D and semigirth ℓ, then L^kG is also d-regular, has $d^k n$ vertices, diameter $D(L^kG) = D(G) + k$ and semigirth $\ell(L^kG) = \ell(G) + k$. See the papers [Ai67],[FaFi89],[FiYeAl84],[ReKuHoLe82].

F35: [FaFi89] Let G be a digraph with minimum degree $\delta > 1$, diameter D and semigirth ℓ.

(a) If $k \geq D - 2\ell$, then $L^k(G)$ is maximally edge-connected.

(b) If $k \geq D - 2\ell + 1$, then $L^k(G)$ is super-λ and maximally connected.

(c) If $k \geq D - 2\ell + 2$, then $L^k(G)$ is super-κ.

REMARK

R22: As shown in Fact 35, the interest of considering k-iterated line digraphs stems from the fact that if k is large enough, Fact 34 guarantees that the conditions of Fact 30 hold.

Girth

For a given girth, high density/connectivity graphs occur when they have a reduced diameter, and also when they have a small number of vertices.

DEFINITION

D15: The same definition for the semigirth (Definition 12) applies for an undirected graph G (considering undirected walks). In this case, it turns out that the semigirth $\ell = \ell(G) = \ell(\overleftrightarrow{G})$ equals $\lfloor (g-1)/2 \rfloor$ where $g = g(G)$ stands for the girth of G.

FACTS

F36: [SoNaIm85,SoNaImPe87,FaFi89] Let G be a graph with minimum degree $\delta > 1$, diameter D, girth g, and connectivities κ and λ.

(a) If $D \leq \begin{cases} g-1, & g \text{ odd,} \\ g-2, & g \text{ even,} \end{cases}$ then $\lambda = \delta$.

(b) If $D \leq \begin{cases} g-2, & g \text{ odd,} \\ g-3, & g \text{ even,} \end{cases}$ then G is super-λ and $\kappa = \delta$.

(c) If $D \leq \begin{cases} g-3, & g \text{ odd,} \\ g-4, & g \text{ even,} \end{cases}$ then G is super-κ.

F37: [BaCaFaFi96,CaFa99] Let G be a graph with minimum degree $\delta > 1$, girth g, and connectivities κ and λ. Let LG be the line graph of G, with diameter $D(LG)$. Then,

(a) If $D(LG) \leq \begin{cases} g, & g \text{ odd,} \\ g-1, & g \text{ even,} \end{cases}$ then $\lambda = \delta$.

(b) If $D(LG) \leq \begin{cases} g-1, & g \text{ odd,} \\ g-2, & g \text{ even,} \end{cases}$ then G is super-λ and $\kappa = \delta$.

(c) If $D(LG) \leq \begin{cases} g-2, & g \text{ odd,} \\ g-2, & g \text{ even,} \end{cases}$ then G is super-κ.

F38: [FaFi96a] Any bipartite graph with diameter three is maximally edge-connected.

REMARKS

R23: Fact 36 is a simple consequence of Definition 15 and Fact 30.

R24: Fact 38 is the undirected version of Fact 32, which can be seen as Plesník's analogue for the bipartite case.

Cages

DEFINITIONS

D16: A (k, g)***-cage*** is a k-regular graph with girth g having the least possible number of vertices.

D17: A 3-connected graph $G = (V, E)$ is said to be ***quasi 4-connected*** if for every vertex-cut $F \subset V$ such that $|F| = 3$, F is the neighborhood of a vertex of degree 3 and $G - F$ has exactly two components.

EXAMPLE

E4: The ***Heawood graph***, shown in Figure 4.7.4, is a $(3, 6)$-cage with order 14 and diameter 3.

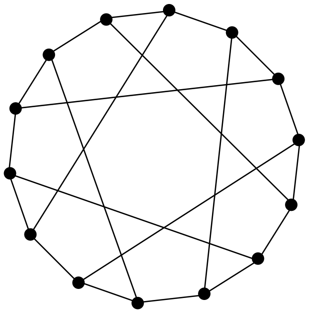


Figure 4.7.4 The Heawood graph.

FACTS

F39: [FuHuRo97,XuWaWa02] All (k, g)-cages are 2-connected and, for $k \leq 4$, every (k, g)-cage is maximally connected.

F40: [DaRo99,JiMu98] Every (k, g)-cage with $k \geq 3$ is 3-connected.

F41: [MaBa03?,MaPeBa02] Every $(3, g)$-cage is superconnected, edge-superconnected, and quasi 4-connected.

F42: [MaBaPeFa03?] Every (k, g)-cage with $k \geq 4$ and $g \geq 10$ is 4-connected.

F43: [WaXuWa03] Every (k, g)-cage with $k \geq 3$ and odd girth g is maximally edge-connected.

F44: [MaBa03?] Every (k, g)-cage with $k \geq 3$ and odd girth g is super-λ.

F45: [MaBaPe03?] $(k, 6)$- and $(k, 8)$-cages are maximally connected.

CONJECTURE

[FuHuRo97] Every (k, g)-cage is maximally connected.

Large Digraphs

The following results support the intuitive idea that dense (di)graphs have high connectedness.

DEFINITION

D18: For a digraph with maximum degree Δ and diameter D, the ***Moore bound***, denoted $n(\Delta, D)$, is given by $n(\Delta, D) = 1 + \Delta + \Delta^2 + \ldots + \Delta^D$.

FACTS

F46: An n-vertex digraph with maximum degree Δ and diameter D has $n \leq n(\Delta, D)$.

F47: [Wa67] The order of a (di)graph with connectivity $\kappa > 1$ and diameter D satisfies $n \geq \kappa(D-1)+2$.

F48: [ImSoOk85]

(a) If $\lambda < \delta$, then $n \leq \lambda \cdot [n(\Delta, D-2) + \Delta + 1]$.

(b) If $\kappa < \delta$, then $n \leq \kappa \cdot [n(\Delta, D-1) + \Delta]$.

F49:

(a) If $n > (\delta - 1)[n(\Delta, D-2) + \Delta + 1]$, then $\lambda = \delta$.

(b) If $n > (\delta - 1)\,[n(\Delta, D-1) + \Delta]$, then $\kappa = \delta$.

F50: [Fi93]

(a) If $\lambda < \delta$, then $n \leq \lambda \cdot [n(\Delta, D-2) + 1] + \Delta$.

(b) If $\kappa < \delta$, then $n \leq \kappa \cdot [n(\Delta, D-1) - 1] + \Delta + 1$.

F51: [Fi93] Let G be d-regular.

(a) If $n > d^{D-1} + 2d - 2$, then $\lambda = d$.

(b) If $n > d^D + 1$, then $\kappa = d$.

F52: [So92,Fi94] Let G be a d-regular digraph, $d \geq 2$.

(a) If $n > \begin{cases} 3d, & D = 2, \\ 2d^{D-1} + d^{D-2} + \cdots + d^2 + 2d, & D \geq 3, \end{cases}$ then G is super-λ.

(b) If $n > \begin{cases} 3d^2 + 1, & D = 3, \\ 2d^{D-1} + d^{D-2} + \cdots + d^3 + 2d^2 + 1, & D \geq 4, \end{cases}$ then G is super-κ.

EXAMPLE

E5: Figure 4.7.5 shows a regular digraph for which $n = 6$, $\Delta = \delta = d = 2$, and $D = 2$. Since $n > d^{D-1} + 2d - 2$ and $n > d^D + 1$, Fact 51 guarantees that it is maximally connected ($\kappa = \lambda = d$).

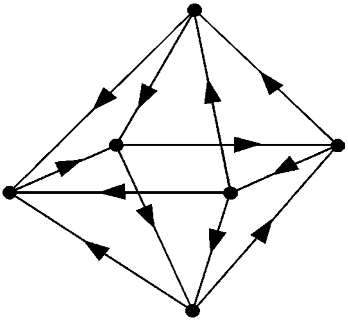


Figure 4.7.5 $\kappa = \lambda = d = 2$.

REMARKS

R25: To our knowledge, Fact 47, due to Watkins, was the first result in which the order n, the diameter D and the connectivity κ were related (in the undirected case). It follows easily from counting the minimum number of vertices involved in κ internally disjoint u-v paths between a pair of vertices u, v at distance D, as Menger's theorem guarantees.

R26: A similar reasoning gives a lower bound for the number of edges m of a (di)graph with edge-connectivity λ, namely, $m \geq \lambda D$. However, it is not difficult to realize that this is not a very strong result. (The situation seems to depend heavily on the values of λ and D: for $\lambda = 3$ there are constructions giving a lower bound of the order of $\frac{7}{2}D$, whereas for $\lambda = 4$ we have a bound which is "asymptotically optimal", that is, of the order of $4D$.)

R27: If we take into account the connectivity parameters κ or λ, the Moore bound can be refined. Intuitively, a disconnecting set with few vertices or edges is a kind of "bottleneck" that prevents the order from being large, as shown in Fact 48 (reformulated as Fact 49) and Fact 50.

R28: Fact 50 is an improvement of Fact 48. Notice that if we set $\kappa = \Delta$ in the upper bound on n of Fact 50(b), we obtain the Moore bound $n(\Delta, D)$.

Large Graphs

Similar results for graphs were derived independently by Esfahanian [Es85], Soneoka et al. [SoNaImPe87], [So92], and [Fi93], [Fi94].

DEFINITION

D19: The ***Moore bound*** for an undirected graph with maximum degree Δ and diameter D is given by $n(\Delta, D) = 1 + \Delta + \Delta(\Delta - 1) + \cdots + \Delta(\Delta - 1)^{D-1}$.

FACTS

F53: [SoNaImPe87]

(a) If $n > (\delta - 1)[n(\Delta - 1, D - 2) + 1] + \Delta - 1$, then $\lambda = \delta$.

(b) If $n > (\delta - 1)(\Delta - 1)^{D-1} + 2$, then $\kappa = \delta$.

F54: [So92,Fi94]

(a) Let $D \geq 2$ and $\delta \geq 2$. If $n > \delta \cdot [n(\Delta - 1, D - 2) + 1] + (\Delta - 1)^{D-1}$, then G is super-λ.

(b) Let $D \geq 3$, $\delta \geq 3$ and $g \geq 5$. If $n > (\delta - 1)[n(\Delta, D - 1) + \Delta]$, then $\kappa = \delta$.

4.7.2 Bounded Connectivity

The techniques used for proving the results of the preceding subsection can often be used to derive bounds on the connectivity or edge-connectivity of a (di)graph. In this subsection, we provide some examples.

π-Semigirth

The following definition generalizes semigirth (Definition 12).

DEFINITION

D20: [FaFi89] Let $G = (V, E)$ be a digraph with minimum degree δ and diameter D, and let π be an integer, $0 \leq \pi \leq \delta - 2$. The ***$\pi$-semigirth*** of G, denoted $\ell_\pi(G)$, is the greatest integer ℓ_π between 1 and D such that, for any $u, v \in V$,

(a) if $dist(u, v) < \ell_\pi$, the shortest u-v path is unique and there are at most π distinct u-v walks of length $dist(u, v) + 1$.

(b) if $dist(u, v) = \ell_\pi$, there is only one shortest u-v walk.

FACT

F55: [FaFi89] Let G be a connected digraph with minimum degree $\delta > 1$, diameter D, π-semigirth ℓ_π for $0 \leq \pi \leq \delta - 2$, and with k-iterated line digraph $L^k G$. Then,

(a) If $D \leq 2\ell_\pi$, then $\lambda \geq \delta - \pi$.

(b) If $D \leq 2\ell_\pi - 1$, then $\kappa \geq \delta - \pi$.

(c) If $k \geq D - 2\ell_\pi$, then $\lambda(L^k G) \geq \delta - \pi$.

(d) If $k \geq D - 2\ell_\pi + 1$, then $\kappa(L^k G) \geq \delta - \pi$.

REMARKS

R29: Note that ℓ_0 corresponds to the ordinary semigirth ℓ. Moreover, for $\pi \geq 1$, ℓ_π is well defined even for a digraph with self-loops.

R30: The definition of ℓ_π is restricted to $\pi \leq \delta - 2$ since, otherwise, the above results become irrelevant.

Imbeddings

Here we cite one of the earliest results relating the connectivity of a graph to a topological property of that graph. Other more recent results of this kind can be found in [PluZh98], [PluZh02].

DEFINITION

D21: A graph G is said to be ***imbeddable*** in a given surface S if G can be drawn on S without edge crossings.

FACT

F56: [Co73] Let G be any graph embeddable in a oriented surface of genus $g > 0$ (where the genus is informally, the number of *handles* on its surface [see Chapter 7]). Then, $\kappa \leq \lfloor (5 + \sqrt{1 + 48g})/2 \rfloor$.

Adjacency Spectrum

Given a (di)graph G with some associated matrix A, a natural problem is to study how much can be said about the structure of G from the spectrum of A. This is a major topic in algebraic graph theory, and has been the object of research (see §6.5 of the *Handbook* or the classic textbooks [Bi94], [CvDoSa95]).

DEFINITIONS

D22: Given a graph G on n vertices, its ***adjacency matrix*** $A = (a_{uv})$ is the $n \times n$ matrix indexed by the vertices of G with entries $a_{uv} = 1$ if u and v are adjacent and $a_{uv} = 0$ otherwise.

D23: The ***toughness*** t of a graph G is defined as $t = \min_S \{|S|/c(G - S)\}$, where S runs over all vertex-cuts of G and $c(G - S)$ denotes the number of components of $G - S$.

FACTS

F57: [Al95,Br95] Let G be a connected non-complete d-regular graph and let λ be the maximum of the absolute values of the eigenvalues of G distinct from d. Then, $t > d/\lambda - 2$.

NOTATION: Given a graph G, let D_2 denote the maximum distance between vertex subsets of G with two vertices. (This parameter is a particular case of the so-called *conditional diameter*, introduced in [BaCaFaFi96].)

F58: [FiGaYe97] Let G be a δ-regular graph with $D_2 > 1$ and distinct eigenvalues (of its adjacency matrix A) $\lambda_0 (= \delta) > \lambda_1 > \cdots > \lambda_r$. Let $P(x) := 2(x - \lambda_r)/(\lambda_1 - \lambda_r) - 1$. Then, $\kappa(G) \geq \min\{\delta, \lceil \frac{2(P^2(\delta)-1)(n-2)}{2(P^2(\delta)-1)+n} \rceil\}$.

REMARKS

R31: Besides Fact 57, Brouwer [Br96] gave some other interesting examples of results about the connectivity of a graph G in terms of its spectrum.

R32: For other results concerning the toughness of a graph, mainly used in the study of vulnerability of network topologies [BoHaKa81], see, for instance [ChLi02].

R33: Notice that, from Fact 15(b), if $D_2 = 1$ then G is maximally connected. Otherwise, Fact 58 applies.

Laplacian Spectrum

DEFINITION

D24: Given a graph G, its ***Laplacian matrix*** is defined as $L = D - A$, where D is the diagonal matrix of the vertex degrees and A is the adjacency matrix of G (see, for instance [Bi94]). The ***Laplacian eigenvalues*** of G are the eigenvalues of its Laplacian matrix.

TERMINOLOGY: The second smallest Laplacian eigenvalue, θ_1, usually denoted by $a = a(G)$, is called the ***algebraic connectivity*** of G because it has some properties which are similar to those satisfied by the connectivity κ.

FACTS

F59: Since L is positive semidefinite, its eigenvalues are all nonnegative, with the first one equal to zero. If G is δ-regular with (distinct) eigenvalues $\lambda_0(=\delta) > \lambda_1 > \cdots \geq \lambda_r$, then its Laplacian eigenvalues are $\theta_0, \theta_1, \ldots, \theta_r$, where $\theta_i = \delta - \lambda_i$, $i = 1, 2, \ldots, r$.

F60: [Fi73] Let G be a graph with second smallest Laplacian eigenvalue a.

(a) $\kappa \geq a \geq 0$, and $a = 0$ if and only if G is not connected.

(b) For any spanning subgraph H of G we have $a(H) \leq a(G)$.

(c) For any vertex subset U of G we have $a(G - U) \geq a(G) - |U|$.

F61: Let G be a δ-regular graph with n vertices, $D_2 > 1$, and Laplacian eigenvalues $\theta_0(=0) < \theta_1 < \theta_2 < \cdots < \theta_r$. If $\delta < \frac{n(\theta_r-\theta_1)^2+8\theta_1\theta_r(n-1)}{n(\theta_r-\theta_1)^2+8\theta_1\theta_r}$, then $\kappa = \delta$.

REMARK

R34: Fact 61 is just a consequence of Fact 58 in terms of the Laplacian eigenvalues.

4.7.3 Symmetry and Regularity

Boundaries, Fragments, and Atoms

The concepts of fragment and atom are very useful in the study of connectivity, both in the undirected and the directed case, and, in particular, for (di)graphs with strong symmetries. For graphs, the concept of atom was introduced independently by Watkins [Wa70] and Mader [Ma71]. The notion of atom for digraphs was introduced by Chaty [Ch76] and first used extensively by Hamidoune [Ha77], [Ha80], [Ha81].

Because of the close relationship between a graph G and its corresponding symmetric digraph $\overset{\leftrightarrow}{G}$, we only give the definitions for digraphs. (For undirected graphs, the corresponding definitions are unsigned.)

DEFINITIONS

D25: The ***positive boundary*** of a vertex subset F in a digraph G, denoted $\partial^+ F$, is the set of vertices that are adjacent *from* F, and the ***negative boundary***, $\partial^- F$, is the set of vertices adjacent *to* F.

D26: The ***positive edge-boundary*** and the ***negative edge-boundary***, denoted ω^+F and ω^-F, respectively, are given by

$$\omega^+F = \{(u,v) \in E : u \in F \text{ and } v \in V - F\}$$
$$\omega^-F = \{(u,v) \in E : u \in V - F \text{ and } v \in F\}$$

D27: Let G be a strongly connected digraph with connectivity κ. A vertex subset F is a ***positive fragment*** of G if $|\partial^+F| = \kappa$ and $V - (F \cup \partial^+F) \neq \emptyset$, and F is a ***negative fragment*** if $|\partial^-F| = \kappa$ and $V - (\partial^-F \cup F) \neq \emptyset$.

D28: Let G be a digraph with edge-connectivity λ. A vertex subset F is a ***positive α-fragment*** of G if $|\omega^+F| = \lambda$, and F is a ***negative α-fragment*** if $|\omega^-F| = \lambda$.

D29: A vertex u of a positive [negative] α-fragment F is called ***interior*** if none of the edges adjacent from [to] u belongs to ω^+F [ω^-F].

D30: An ***atom*** is a (positive or negative) fragment of minimum cardinality.

EXAMPLE

E6: For the digraph of Figure 4.7.6, $\kappa = 2$ and F is a positive (respectively, negative) fragment with positive (respectively, negative) boundary $\{u, v\}$ (respectively, $\{z, t\}$). Analogously, $\omega^+(F) = \{(x, v), (y, u)\}$ and $\omega^-(F) = \{(z, x), (t, y)\}$. In this digraph, each single vertex is an atom.

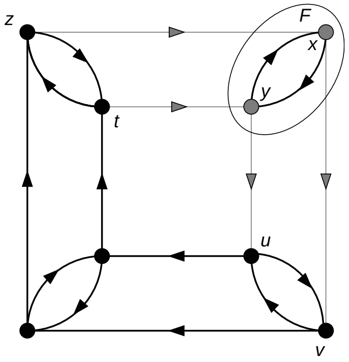


Figure 4.7.6 F **is a fragment.**

FACT

F62: If $F \cup \partial^+F \neq V$ [$F \cup \partial^-F \neq V$], then ∂^+F [∂^-F] is a vertex-cut of G. Similarly, if F is a proper (nonempty) subset of V, then ω^+F [ω^-F] is an edge-cut. Using these concepts, we have the following alternative definitions of the connectivity parameters:

$$\kappa = \min\{|\partial^+F| : \; F \subset V,\, F \cup \partial^+F \neq V \;\text{ or }\; |F| = 1\}$$
$$\lambda = \min\{|\omega^+F| : \; F \text{ is a nonempty, proper subset of } V\}$$

Fragments and Atoms in Undirected Graphs

FACTS

F63: [Wa70] In a connected graph, any two distinct atoms are disjoint.

F64: [Ma71] Let G be a graph with order n and connectivity κ. Let F_1 and F_2 be distinct fragments of G, with at most $n - 3\kappa/2$ vertices. Then $F_1 \cap F_2 = \emptyset$.

F65: Let G be a graph with connectivity κ. If T is a disconnecting set with κ vertices and A is an atom, then either $A \subset T$ or $A \cap T = \emptyset$. Moreover, $\kappa \geq 2|A|$.

REMARKS

R35: To quote a remark from Prof. Watkins: "It is an amazing coincidence that Prof. Mader and I not only conceived of the notion of "atom" independently and simultaneously, but we also accorded this notion almost identical names. In my paper [Wa70], I used the term "atomic part", while Prof. Mader [Ma71] used the term "atom". Since "atom" is shorter, I subsequently used that word, too."

R36: As already mentioned, the seminal papers on atoms are those of Watkins [Wa70] and Mader [Ma71]. Notice that Fact 64 is a generalization of Fact 63.

R37: Results on atoms and the connectivity of infinite graphs can be found in [JuWa77] and [Ha89].

Fragments and Atoms in Digraphs

The results above can be seen as consequences of the corresponding directed versions, which are due to Hamidoune.

FACTS

F66: [Ha77] Let G be a connected digraph with a positive (negative) atom A and a positive (negative) fragment F. Then, either $A \subset F$ or $A \cap F = \emptyset$. In particular, two distinct positive (negative) atoms are disjoint.

F67: [Ha80] If G is a connected digraph with $\lambda < \delta^+$ ($\lambda < \delta^-$), then every positive (negative) α-fragment contains an interior vertex.

REMARKS

R38: Contrary to the case of graphs, where the presence of an atom is always assured, a digraph does not necessarily have an atom with a prescribed sign.

R39: Fact 67 implies Jolivet's theorem (Fact 21).

Graphs with Symmetry

Graphs with high symmetry often have "good" properties, and their study has special relevance to other areas of mathematics. In particular, the results here show that, for connected graphs, high symmetry goes hand in hand with high connectivity. Graph automorphisms and symmetry are discussed in §6.1 and §6.2.

DEFINITION

D31: A (di)graph G is ***vertex-transitive*** (or ***vertex-symmetric***) if for any vertices u, v there is an automorphism of G which maps u into v. Similarly, G is called ***edge-transitive*** (or ***edge-symmetric***) if for any (possibly oriented) edges uv, wz there is an automorphism of G that maps uv into wz.

FACTS

F68: Edge-symmetry implies vertex-symmetry, but the converse is not true.

F69: [Ma71,Wa70] Let G be a vertex-symmetric connected graph with degree d. Then $\lambda = d$ and $\kappa \geq 2\lfloor d/3 \rfloor + 2$. Furthermore, if G does not contain K_4, then $\kappa = d$.

F70: [Ma70] Let G be an edge-symmetric connected graph with degree d. Then $\kappa = \lambda = d$.

F71: [Wa70] Let G be a vertex-transitive graph with an atom A. Then the subgraph $G[A]$ induced on A is also vertex-transitive. Moreover, the set of atoms of G constitute a partition of $V(G)$.

F72: [Ha77] Let G be a vertex-transitive digraph with a positive (negative) atom A. Then, the induced subdigraph $G[A]$ is also vertex-transitive. Furthermore, the set of positive (negative) atoms of G constitute a partition of $V(G)$.

F73: [Ha81] Let G be a vertex-symmetric strongly connected digraph with (constant) outdegree d^+. Then $\lambda = d^+$ and $\kappa \geq \frac{1}{2}d^+$. Moreover, if G is an oriented graph, then $\kappa \geq \frac{2}{3}d^+$.

REMARKS

R40: The first two statements in Fact 69 are consequences of Fact 63. Their improvements are due to Mader.

R41: From Facts 71 and 72, the order of a (positive or negative) atom of G divides the order of G. Consequently, every connected vertex-transitive (di)graph with a prime number of vertices is maximally connected ($\kappa = \lambda = \delta$). In fact, it is known that such (di)graphs must be Cayley (di)graphs of cyclic groups.

R42: By Fact 66, Hamidoune [Ha77] proved Kameda's result stating that every minimal k-connected digraph has one vertex of out-degree or in-degree k [Ka74], and Hamidoune also proved that every edge-transitive digraph is maximally connected.

Cayley Graphs

The *Cayley graphs* are among the most interesting vertex-symmetric (di)graphs, mainly because of their relationship with group theory (see §6.1 and §6.2). In particular, the study of the connectivity of Cayley graphs has striking connections with some key results in additive number theory, such as the well-known Cauchy-Davenport theorem: If p is a prime number and A, B are two nonempty subsets of the cyclic group Z_p, then either $A + B = Z_p$ or $|A + B| \geq |A| + |B| - 1$.

DEFINITIONS

D32: Let Γ be a finite group with identity element e and generating set $S \subset \Gamma - \{e\}$. The ***Cayley digraph*** $G = (\Gamma, S)$ has vertices labeled with the elements of Γ, and edges of the form (u, ug) where $g \in S$. In particular, when $S^{-1} = S$ (where $S^{-1} = \{x^{-1} : x \in S\}$) we obtain a symmetric Cayley digraph or, simply, a ***Cayley graph***.

D33: If Γ is a cyclic group, then the Cayley graph is called a ***circulant graph***.

D34: A generating set S of a group Γ is called ***minimal*** when any proper subset $S' \subset S$ does not generate Γ.

D35: The ***symmetric group*** on n elements, denoted Σ_n, is the group of all permutations of the set $\{1, 2, \ldots, n\}$.

D36: Let H be a subgroup of a group G, and let $x \in G$. Then the set $xH = \{xh|h \in H\}$ is a ***left coset*** of G with respect to H.

FACTS

F74: [Im79] Let S be a generating set of the symmetric group $\Gamma = \Sigma_n$ with $n \geq 5$, such that $xSx^{-1} = S$ for every $x \in \Gamma$. Then, the Cayley digraph (Γ, S) is maximally connected (that is, $\kappa = |S|$).

F75: [Ha84] Let Γ be a finite group with identity e and generating set S. Let A be a positive (respectively, negative) atom of (Γ, S) containing e. Then A is the subgroup of Γ generated by $S \cap A$, and the positive (respectively, negative) atoms of (Γ, S) are the left cosets of Γ with respect to A.

F76: [Ha84] Let Γ be a finite group with a minimal generating set S. Let $S' \subset S^{-1}$. Then, the Cayley digraph $(\Gamma, S \cup S')$ is maximally connected.

F77: [HaSe96] Let Γ be an Abelian group of order n and let S be a generating subset of Γ such that $|S \cup \{0\}| \leq n-1$. Let D be the diameter of $G = (\Gamma, S)$. Then there is a vertex-cut of size less than $(4n\ln(n/2))/D$ whose deletion separates G into a negative fragment B and a positive fragment $\overline{B}$ such that $|B| = |\overline{B}|$. Moreover, G can be separated into two equal parts of size $|B|$ by deleting less than $(8e/|S|)n^{1-1/|S|}\ln(n/2)$ vertices.

REMARKS

R43: Fact 75, due to Hamidoune, provides a very short proof of Fact 74.

R44: For the case of Cayley graphs, Fact 76 was previously proved by Godsil [Go81]. Subsequently, Akers and Krishnamurthy [AkKr87], Hamidoune, Lladó, and Serra [HaLlSe92], and Alspach [Als92] improved these results by considering Cayley (di)graphs with a ***hierarchical*** generating set (that is, when the group generated by the first k generators is a proper subgroup of the group generated by the first $k+1$ for each k).

Circulant Graphs

Because of their circular symmetry, the circulant graphs have been proposed as good models for local area network topologies, where they are called *loop networks*. In this context, other good topologies are provided by Cayley graphs of Abelian groups, also called loop networks (see [BeCoHs95],[BoTi84]).

FACTS

F78: [Ha84] Let Γ be the cyclic group Z_n. Let S be the strictly increasing sequence of s integers $(1=)b_1 < b_2 < \ldots < b_s (< n)$ such that $b_{i+1} - b_i \geq \min\{2, b_i - b_{i-1}\}$ for $i = 2, 3, \ldots, s-1$. Then the circulant digraph (Γ, S) is maximally connected $(\kappa = s)$.

F79: [Ha84] Let Γ be the cyclic group Z_n. Let S be the strictly increasing sequence of s integers $(1=)b_1 < b_2 < \ldots < b_s (< n/2)$ such that $b_{i+1} - b_i \geq \min\{2, b_i - b_{i-1}\}$ for $i = 2, 3, \ldots, s-1$, and let $S' \subset -S$, where $-S$ denotes the set of (additive) inverses of the elements in S. Then the circulant digraph $(\Gamma, S \cup S')$ is maximally connected $(\kappa = |S \cup S'|)$.

F80: The Cauchy-Davenport theorem is equivalent to stating that, for any generating set $S \subset Z_p$, p prime, the Cayley digraph (Z_p, S) is maximally connected (that is, $\kappa = |S|$).

REMARKS

R45: The case $S' = -S$ in Fact 79 (that is, for circulant graphs) was proved in [BoFe70] using the "convexity conditions" $b_{i+1} - b_i \geq b_i - b_{i-1}$ (see also [BoTi84]).

R46: Fact 80, noted by Hamidoune, is a bridge between additive number theory and graph theory. (For a comprehensive survey on the subject, we refer the reader to [Ha96].)

Distance-Regular Graphs

The concept of distance-regularity was introduced by Biggs in the early 1970s. Distance-regular graphs have important connections with other branches of mathematics, such as geometry, coding theory, and group theory, as well as with other areas of graph theory. In our context, their high regularity seems also to induce a high degree of connectedness.

DEFINITIONS

D37: Let G be a regular graph with diameter D and let k be an integer between 1 and D. Graph G is said to be ***distance-regular*** if, for any two vertices u and v with $dist(u, v) = k$, the numbers c_k, a_k, and b_k of vertices that are adjacent to v and whose distance from u are $k-1$, k, and $k+1$, respectively, depend only on k.

D38: An n-vertex k-regular graph G is called $(n, k; a, c)$***-strongly-regular*** if any two adjacent vertices have a common neighbours and any two non-adjacent vertices have c common neighbors.

FACTS

F81: Let G be a connected graph. Then G is strongly-regular if and only if G is distance-regular of diameter two.

F82: Every strongly regular graph is maximally edge-connected.

F83: [BrMe85] Every strongly regular graph is maximally connected and super-κ.

CONJECTURE

[Br96] Every distance-regular graph is maximally connected.

REMARKS

R47: Fact 82 is a consequence of Facts 3 and 81.

R48: The conjecture above has been proved for some families of distance-regular graphs, such as the so-called *odd graphs* O_k (having the k-subsets of a $(2k-1)$-set as its vertices and adjacencies defined by void intersection); see [Gh92]. If true, the conjecture would imply a previous conjecture of Godsil [Go81], stated in the context of association schemes, that every distance-regular graph is maximally edge-connected.

4.7.4 Generalizations of the Connectivity Parameters

The standard connectivity parameters have been generalized in different ways, giving rise to numerous articles, see, for instance, [BaBeLiPi87], [BeOePi02], [Ha83], [Wo73]. Here we will consider several examples, some of which have special relevance to the study of network vulnerability.

Conditional Connectivity

The next two definitions generalize the concept of superconnectivity.

DEFINITIONS

D39: Given a graph $G = (V, E)$ and a nonnegative integer s, a vertex subset $V' \subset V$ is said to be ***n*-*trivial*** if it contains the boundary $\partial(H)$ of some subgraph $H \subset G$ with s' vertices, $1 \leq s' \leq s$. Similarly, an edge subset $E' \subset E$ is said to be ***s*-*trivial*** if it contains the edge-boundary $\omega(H)$ of some subgraph $H \subset G$ with s' vertices, $1 \leq s' \leq s$.

D40: The ***conditional connectivity*** κ_s of a graph G is the minimum cardinality of a disconnecting set that is not s-trivial. The ***conditional edge-connectivity*** λ_s of G is the minimum cardinality of a disconnecting edge set that is not s-trivial.

FACT

F84: [FaFi89,FiFaEs90,FaFi94] Let G be a graph with minimum degree $\delta > 1$, diameter D and girth g. Let $\ell = \lfloor \frac{g-1}{2} \rfloor$. Then,

(a) If $D \leq 2\ell$, then $\lambda_0 = \delta$.

(b) If $D \leq 2\ell - 1$, then $\kappa_0 = \delta$ and $\lambda_1 \geq 2\delta - 2$.

(c) If $D \leq 2\ell - 2$, then $\kappa_1 \geq 2\delta - 2$ and $\lambda_2 \geq 3\delta - 4$.

(d) If $D \leq 2\ell - 3$, then $\kappa_2 \geq 3\delta - 4$.

CONJECTURE [FaFi94]

(a) If $D \leq 2\ell - s$, then $\lambda_s \geq (s+1)\delta - 2s$.

(b) If $D \leq 2\ell - s - 1$, then $\kappa_s \geq (s+1)\delta - 2s$.

REMARKS

R49: Harary [Ha83] introduced the general concept of conditional connectivity. In our context, the graphs are assumed to be those for which κ_s and λ_s are well-defined.

R50: Note that the conditional connectivities κ_0 and λ_0 correspond to the standard connectivities κ and λ (thus, Fact 84 generalizes Fact 36). If $\kappa_1 > \delta$, then G is super-κ, and if $\lambda_1 > \delta$, then G is super-λ.

R51: The conjecture above was proved to be true for even s (provided that $\delta > 2$ and $\ell > (s+1)/2$) [FaFi96b]. Moreover, if s is large enough in comparison with the minimum degree δ, further improvements of the sufficient conditions were given in [BaCaFaFi97b], [Ba99].

Distance Connectivity

Here we consider a generalization of the concepts of connectivity and edge-connectivity of a (di)graph, introduced in [FiFa94] and [BaCaFi96], which takes into account the distance between vertices.

DEFINITIONS

D41: Let $G = (V, E)$ be a digraph. Given $u, v \in V$ such that $(u, v) \notin E$, recall (from §4.1) that a set $S \subset V - \{u, v\}$ is called a $(u|v)$***-set*** if there is no u-v path in $G - S$, and $\kappa(u|v)$ is the minimum cardinality of a $(u|v)$-set. Similarly, a given edge set $T \subset E$ is called a $(u|v)$***-edge-set*** for some $u, v \in V$ if there is no u-v path in $G - T$, and $\lambda(u|v)$ is the minimum cardinality of a $(u|v)$-edge-set.

D42: Let $G = (V, E)$ be a digraph with diameter D. Given t, $1 \leq t \leq D$, the $t-$***distance connectivity*** of G, denoted by $\kappa(t; G) = \kappa(t)$, is defined as $\kappa(t) = \min\{\kappa(u|v) : u, v \in V,\ dist(u, v) \geq t\}$ if $t \geq 2$, and $\kappa(1) = \kappa$, where κ is the standard connectivity of G. Analogously, the t***-distance edge-connectivity*** is $\lambda(t; G) = \lambda(t) = \min\{\lambda(u|v) : u, v \in V,\ dist(u, v) \geq t\}$ for $t \geq 1$.

FACTS

F85:

(a) $\kappa = \kappa(1) = \kappa(2) \leq \kappa(3) \leq \cdots \leq \kappa(D)$.

(b) $\lambda = \lambda(1) \leq \lambda(2) \leq \cdots \leq \lambda(D)$.

F86: Let G be a digraph with minimum degree $\delta > 1$ and semigirth ℓ (see Definition 12). Then,

(a) If $\lambda < \delta$ then $D \geq 2\ell + 1$ and $\lambda = \lambda(2\ell + 1)$.

(b) If $\kappa < \delta$ then $D \geq 2\ell$ and $\kappa = \kappa(2\ell)$.

F87:

(a) $\lambda = \delta$ if and only if $D \leq 2\ell$ or $\lambda(2\ell + 1) \geq \delta$.

(b) $\kappa = \delta$ if and only if $D \leq 2\ell - 1$ or $\kappa(2\ell) \geq \delta$.

F88: Every digraph with distance connectivity $\lambda(3) \geq \delta$ has maximum edge-connectivity.

F89: Let G be an undirected graph with associated symmetric digraph $\overset{\leftrightarrow}{G}$. Since a minimum t-distance disconnecting set of $\overset{\leftrightarrow}{G}$ cannot contain digons, $\kappa(t; \overset{\leftrightarrow}{G}) = \kappa(t; G)$ and $\lambda(t; \overset{\leftrightarrow}{G}) = \lambda(t; G)$.

F90: Let G be an undirected graph with girth g and $\delta > 1$.

(a) If $\lambda < \delta$ then $\begin{cases} D \geq g \text{ and } \lambda = \lambda(g), & g \text{ odd}, \\ D \geq g - 1 \text{ and } \lambda = \lambda(g - 1), & g \text{ even}. \end{cases}$

(b) If $\kappa < \delta$ then $\begin{cases} D \geq g - 1 \text{ and } \kappa = \kappa(g - 1), & g \text{ odd}, \\ D \geq g - 2 \text{ and } \kappa = \kappa(g - 2), & g \text{ even}. \end{cases}$

F91:

(a) $\lambda = \delta$ if and only if $\begin{cases} D \geq g-1 \text{ or } \lambda(g) \geq \delta, & g \text{ odd}, \\ D \geq g-2 \text{ or } \lambda(g-1) \geq \delta, & g \text{ even}. \end{cases}$

(b) $\kappa = \delta$ if and only if $\begin{cases} D \geq g-2 \text{ or } \kappa(g-1) \geq \delta, & g \text{ odd}, \\ D \geq g-3 \text{ or } \kappa(g-2) \geq \delta, & g \text{ even}. \end{cases}$

F92: Any graph with distance connectivity $\lambda(3) \geq \delta$ has maximum edge-connectivity.

REMARKS

R52: In Fact 87, since $\kappa(t)$ and $\lambda(t)$ are defined only for $t \leq D$, the two sufficient conditions on the diameter and the distance connectivity are complementary to one another.

R53: Since the semigirth of any digraph is at least one, Fact 86(a) implies Fact 88, which complements Jolivet's result (Fact 21).

R54: Fact 90 follows from Fact 86 by considering Fact 89 and $\ell(\overset{\leftrightarrow}{G}) = \lfloor (g-1)/2 \rfloor$.

High Distance Connectivity

DEFINITIONS

D43: Given a vertex u of a digraph G, the ***out-*** and ***in-eccentricity*** of u, are $ecc^+(u) = \max_{v \in V}\{dist(u,v)\}$ and $ecc^-(u) = \max_{v \in V}\{dist(v,u)\}$, respectively.

D44: For any integer t, $1 \leq t \leq D$, the ***minimum t-degree*** of a digraph G is $\delta(t) = \min\{\delta^+(t), \delta^-(t)\}$, where $\delta^+(t) = \min_{u \in V}\{deg^+(u) : ecc^+(u) \geq t\}$ and $\delta^-(t) = \min_{u \in V}\{deg^-(u) : ecc^-(u) \geq t\}$.

D45: A connected digraph G with diameter D is said to be ***s-geodetic***, for some $1 \leq s \leq D$, if any two vertices of G are joined by at most one path of length less than or equal to s. If $s = D$, the digraph is called ***strongly geodetic*** (see [BoKoZn68],[PlZn74]).

FACTS

F93: $\delta = \delta(1) = \cdots = \delta(r) \leq \delta(r+1) \leq \cdots \leq \delta(D)$.

F94: For any t, $1 \leq t \leq D$, $\kappa(t) \leq \lambda(t) \leq \delta(t)$.

TERMINOLOGY: A digraph G is called ***maximally t-distance connected*** when $\kappa(t) = \lambda(t) = \delta(t)$, and ***maximally t-distance edge connected*** when $\lambda(t) = \delta(t)$.

F95: If a digraph G is maximally connected, then G is maximally t-distance connected for any $1 \leq t \leq r$.

F96: [BaCaFaFi97a] Let G be a s-geodetic digraph. Then,

(a) $\lambda(t) = \min\{\delta(t), \lambda(2\ell+1)\}$, for any $t \leq 2s+1$.

(b) $\kappa(t) = \min\{\delta(t), \kappa(2\ell)\}$, for any $t \leq 2s$.

F97: [BaCaFaFi97a] Let G be a s-geodetic digraph.

(a) G is maximally t-distance connected for any $t \leq 2s$ if $D \leq 2\ell - 1$.

(b) G is maximally t-distance edge connected for any $t \leq 2s+1$ if $D \leq 2\ell$.

F98: Let G be a graph with girth g and diameter D. Then, for any $1 \leq t \leq D$,

(a) G is maximally t-distance edge connected if $\begin{cases} D \leq g-1, & g \text{ odd} \\ D \leq g-2, & g \text{ even.} \end{cases}$

(b) G is maximally t-distance connected if $\begin{cases} D \leq g-2, & g \text{ odd} \\ D \leq g-3, & g \text{ even.} \end{cases}$

Maximal Connectivity

Instead of looking for minimum disconnecting sets, we can consider those (minimal) disconnecting sets with maximum cardinalities. This leads to considering the following connectivity parameters.

NOTATION: Denote by $\kappa_{\max}$ and $\lambda_{\max}$ the maximum cardinality of a minimal disconnecting (vertex-)set and a minimal disconnecting edge set, respectively.

FACTS

F99: $\kappa_{\max} \geq \kappa$ and $\lambda_{\max} \geq \lambda$.

F100: [PeLaHe86] For any non-trivial graph G with order n and maximum degree $\Delta \neq n-1$ we have $\kappa_{\max} \leq \lambda_{\max}$. Furthermore, if G is 2-connected, then $\lambda_{\max} \geq \Delta$.

F101: [PeLaHe86] Let G be an n-vertex graph with minimum degree δ.

(a) If $\delta \geq \lfloor n/2 \rfloor$, then $\lambda_{\max} \geq \delta$.

(b) If $\delta \geq \lfloor (n+i)/2 \rfloor$ for some i with $1 \leq i \leq n/2$, then $\lambda_{\max} \geq i\lfloor (n-i+2)/2 \rfloor$.

(c) If $\delta \geq \lfloor (n+i)/2 \rfloor$ for some i with $n/2 < i < n-2$, then $\lambda_{\max} \geq \lceil n/2 \rceil \cdot \lfloor (i+1)/2 \rfloor$.

Hamiltonian Connectivity

DEFINITIONS

D46: A graph G is ***hamiltonian connected*** if between any pair of vertices u, v there is a Hamilton u-v path in G.

D47: A graph G ***k-leaf-connected*** if $|V(G)| > k$ and for each subset S of $V(G)$ with $k = |S|$ there exists a spanning tree T with precisely S as set of endvertices (vertices of degree 1).

FACTS

F102: [GuWa86] Let u and v be non-adjacent vertices of G with $d(u)+d(v) \geq |V(G)|+k-1$. If $G+uv$ is k-leaf-connected, then G is k-leaf-connected.

F103: [GuWa86] For all natural numbers n, k, $2 \leq k < n-2$ there are k-leaf-connected graphs with $\lceil (k+1)n/2 \rceil$ edges (the minimum number of edges that a k-leaf-connected graph on n vertices can have).

REMARK

R55: The generalization of the concept of hamiltonian connectivity (Definition 47) is due to Murty. Notice that G is hamiltonian-connected if and only if G is 2-leaf-connected.

References

[Ai67] M. Aigner, On the linegraph of a directed graph, *Math. Z.* 102 (1967), 56–61.

[AkKr87] S.B. Akers and B. Krishnamurthy, On group graphs and their fault tolerance, *IEEE Trans. Comput.* 36 (1987), 885–888.

[Al95] N. Alon, Tough Ramsey graphs without short cycles, *J. Algebraic Combin.* 4 (1995), 189–195.

[Als92] B. Alspach, Cayley graphs with optimal fault tolerance, *IEEE Trans. Comput* 41 (1992), 1337–1339.

[AyFr70] J.N. Ayoub and I.T. Frisch, On the smallest-branch cuts in directed graphs, *IEEE Trans. Circuit Theory* CT-17 (1970), 249–250.

[Ba99] M.C. Balbuena, Extraconnectivity of s-geodetic digraphs and graphs, *Discrete Math.* 195 (1999), 39–52.

[BaBeLiPi87] K.S. Bagga, L.W. Beineke, M.J. Lipman, and R.E. Pippert, On the edge-integrity of graphs, *Congressus Numerantium* 60 (1987), 141–144.

[BaCaFaFi96] M.C. Balbuena, A. Carmona, J. Fàbrega, and M.A. Fiol, On the connectivity and the conditional diameter of graphs and digraphs, *Networks* 28 (1996), 97–105.

[BaCaFaFi97a] M.C. Balbuena, A. Carmona, J. Fàbrega, and M.A. Fiol, On the order and size of s-geodetic digraphs with given connectivity, *Discrete Math.* 174 (1997), 19–27.

[BaCaFaFi97b] M.C. Balbuena, A. Carmona, J. Fàbrega, and M.A. Fiol, Extraconnectivity of graphs with large minimum degree and girth, *Discrete Math.* 167/168 (1997), 85–100.

[BaCaFi96] M.C. Balbuena, A. Carmona, and M.A. Fiol, Distance connectivity in graphs and digraphs, *J. Graph Theory* 22 (1996), 281–292.

[BeCoHs95] J.-C. Bermond, F. Comellas, and D.F. Hsu, Distributed loop computer networks: a survey, *J. Parallel Distrib. Comput.* 24 (1995) 2–10.

[BeOePi02] L.W. Beineke, O. Oellermann, and R.E. Pippert, The average connectivity of a graph, *Discrete Math.* 252 (2002), 31–45.

[Bi94] N. Biggs, *Algebraic Graph Theory*, Second edition, Cambridge University Press, Cambridge, 1994.

[Bo79] B. Bollobás, On graphs with equal edge connectivity and minimum degree, *Discrete Math.* 28 (1979), 321–323.

[BoFe70] F. Boesch and A.P. Felger, A class of non vulnerable graphs. *Networks* 2 (1970), 261–283.

[BoHaKa81] F. Boesch, F. Harary, and J.A. Kabell, Graphs as models of communication network vulnerability: Connectivity and persistence. *Networks* 11 (1981), 57–63.

[BoKoZn68] J. Bosák, A. Kotzig, and Š. Znám, Strongly geodetic graphs, *J. Combin. Theory* 5 (1968), 170–176.

[BoTi84], F. Boesch and R. Tindell, Circulants and their connectivities, *J. Graph Theory* 8 (1984), 487–499.

[Br95] A.E. Brouwer, Toughness and spectrum of a graph, *Linear Algebra Appl.* 226-228 (1995), 267–271.

[Br96] A.E. Brouwer, Spectrum and connectivity of graphs, *CWI Q.* 9 (1996), 37–40.

[BrMe85] A.E. Brouwer and D.M. Mesner, The connectivity of strongly regular graphs, *European J. Combin.* 6 (1985), 215–216.

[CaFa99] M.C. Balbuena and J. Fàbrega, On the superconnectivity and the conditional diameter of graphs and digraphs, *Networks* 34 (1999), 197–205.

[Ch66] G. Chartrand, *SIAM J. Appl. Math.* 14 (1966), 778–781.

[Ch76] G. Chaty, On critically and minimally k vertex (arc) strongly connected digraphs, *Proc. Keszthely* (1976), 193–203.

[ChLi02] E. Cheng and M.J. Lipman, Vulnerability issues of star graphs, alternating group graphs and split-stars: Strength and toughness, *Discrete Appl. Math.* 118 (2002), 163–179.

[Co73] R.J. Cook, Heawood's theorem and connectivity, *Mathematica Lond.* 20 (1973), 201–207.

[CvDoSa95] D. Cvetković, M. Dragoš, and H. Sach *Spectra of Graphs, Theory and Applications.* Third edition, J.A. Barth Verlag, Leipzig, 1995.

[DaRo99] M. Daven and C.A. Rodger, (k, g)-cages are 3-connected, *Discrete Math.* 199 (1999), 207–215.

[DaVo95] P. Dankelmann and L. Volkmann, New sufficient conditions for equality of minimum degree and edge-connectivity, *Ars Comb.* 40 (1995), 270–278.

[DaVo97] P. Dankelmann and L. Volkmann, Degree sequence conditions for maximally edge-connected graphs and digraphs, *J. Graph Theory* 26 (1997), 27–34.

[DaVo00] P. Dankelmann and L. Volkmann, Degree sequence conditions for maximally edge-connected graphs depending on the clique number, *Discrete Math.* 211 (2000), 217–223.

[Es85] A.H. Esfahanian, Lower-bounds on the connectivities of a graph. *J. Graph Theory* 9 (1985), 503–511.

[FaFi89] J. Fàbrega and M.A. Fiol, Maximally connected digraphs, *J. Graph Theory* 13 (1989), 657–668.

[FaFi94] J. Fàbrega and M.A. Fiol, Extraconnectivity of graphs with large girth, *Discrete Math.* 127 (1994), 163–170.

[FaFi96a] J. Fàbrega and M.A. Fiol, Bipartite graphs and digraphs with maximum connectivity, *Discrete Appl. Math.* 69 (1996), 269–277.

[FaFi96b] J. Fàbrega and M.A. Fiol, On the extraconnectivity of graphs, *Discrete Math.* 155 (1996), 49–57.

[Fi73] M. Fiedler, Algebraic connectivity of graphs, *Czech. Math. J.* 23 (1973), 298–305.

[Fi92] M.A. Fiol, On super-edge-connected digraphs and bipartite digraphs, *J. Graph Theory* 16 (1992), 545–555.

[Fi93] M.A. Fiol, The connectivity of large digraphs and graphs, *J. Graph Theory* 17 (1993), 31–45.

[Fi94] M.A. Fiol, The superconnectivity of large digraphs and graphs, *Discrete Math.* 124 (1994), 67–78.

[FiFa94] M.A. Fiol and J. Fàbrega, On the distance connectivity of graphs and digraphs, *Discrete Math.* 125 (1994), 169–176.

[FiFaEs90] M.A. Fiol, J. Fàbrega, and M. Escudero, Short paths and connectivity in graphs and digraphs, *Ars Combin.* 29B (1990), 17–31.

[FiGaYe97] M.A. Fiol, E. Garriga, and J.L.A. Yebra, The alternating polynomials and their relation with the spectra and conditional diameters of graphs, *Discrete Appl. Math.* 167/168 (1997), 297–307.

[FiYeAl84] M.A. Fiol, J.L.A. Yebra, and I. Alegre, Line digraph iterations and the (d, k) digraph problem, *IEEE Trans. Comput.* C-33 (1984), 400–403.

[FrOeSw00] G. Fricke, O.R. Oellermann, and H.C. Swart, The edge-connectivity, average edge-connectivity and degree conditions, manuscript (2000).

[FuHuRo97] H.L. Fu, K.C. Huang, and C.A. Rodgers, Connectivity of cages, *J. Graph Theory* 24 (1997), 187–191.

[Gh92] A. Ghafoor, Connectivity, persistence and fault diagnosis of interconnection networks based on O_k and $2O_k$ graphs, *Discrete Appl. Math.* 37/38 (1992), 205–226.

[Go81] C. D. Godsil, Equiarboreal graphs, *Combinatorica* 1 (1981), 163–167.

[GoEn79] D.L. Goldsmith and R.C. Entringer, A sufficient condition for equality of edge-connectivity and minimum degree of a graph, *J. Graph Theory* 3 (1979), 251–255.

[GoWh78] D.L. Goldsmith and A.T. White, On graphs with equal edge-connectivity and minimum degree, *Discrete Math.* 23 (1978), 31–36.

[GuWa86] M.A. Gurgel and Y. Wakabayashi, On k-leaf-connected graphs, *J. Combin. Theory Ser. B* 41 (1986), 1–16.

[Ha77] Y.O. Hamidoune, Sur les atomes d'un graphe orienté, *C. R. Acad. Sci. Paris Sér. A* 284 (1977), 1253–1256.

[Ha80] Y.O. Hamidoune, A property of α-fragments of a digraph, *Discrete Math.* 31 (1980), 105–106.

[Ha81] Y.O. Hamidoune, Quelques problèmes de connexité dans les graphes orientés, *J. Combin. Theory Ser. B* 30 (1981), 1–10.

[Ha83] F. Harary, Conditional connectivity, *Networks* 13 (1983), 347–357.

[Ha84] Y.O. Hamidoune, On the connectivity of Cayley digraphs, *European J. Combin.* 5 (1984), 309–312.

[Ha89] Y.O. Hamidoune, Sur les atomes d'un graphe de Cayley infini, *Discrete Math.* 73 (1989), 297–300.

[Ha96] Y.O. Hamidoune, Additive group theory applied to network topology, pages 1–39 in D.-Z. Du et al. (Eds.), *Combinatorial network theory, Appl. Optim.* 1, Kluwer Academic Publishers, Dordrecht, 1996.

[HaLlSe92] Y.O. Hamidoune, A.S. Lladó, and O. Serra, The connectivity of hierarquical Cayley digraphs, *Discrete Appl. Math.* 37/38 (1992), 275–280.

[HaSe96] Y.O. Hamidoune and O. Serra, On small cuts separating Abelian Cayley graphs into two equal parts, *Math. Syst. Theory* 29 (1996), 407–409.

[HeVo03a] A. Hellwig and L. Volkmann, Maximally edge-connected digraphs, *Austral. J. Combin.* 27 (2003), 23–32.

[HeVo03b] A. Hellwig and L. Volkmann, Maximally local-edge-connected graphs and digraphs, *Ars Combin.*, to appear.

[Im79] W. Imrich, On the connectivity of Cayley graphs, *J. Combin. Theory Ser. B* 26 (1979), 323–326.

[ImSoOk85] M. Imase, T. Soneoka, and K. Okada, Connectivity of regular directed graphs with small diameter, *IEEE Trans. Comput.* C-34 (1985), 267–273.

[JiMu98] T. Jiang and D. Muyabi, Connectivity and separating sets of cages, *J. Graph Theory* 29 (1998), 35–44.

[Jo72] J.L. Jolivet, Sur la connexité des graphes orientés, *C.R. Acad. Sci. Paris* 274A (1972), 148–150.

[JuWa77] H.A. Jung and M.E. Watkins, On the connectivities of finite and infinite graphs, *Mh. Math.* 83 (1977), 121–131.

[Ka74] T. Kameda, Note on Halin's theorem on minimally connected graphs, *J. Combin. Theory Ser. B* 17 (1974), 1–4.

[Ke72] A.K. Kelmans, Asymptotic formulas for the probability of k-connectedness of random graphs, *Theory Probab. Appl.* 17 (1972), 243–254.

[Le74] L. Lesniak, Results on the edge-connectivity of graphs, *Discrete Math.* 8 (1974), 351–354.

[Ma70] W. Mader, Über den Zusammenhäng symmetrischer Graphen, *Arch. Math. (Basel)* 21 (1970), 331–336.

[Ma71] W. Mader, Eine Eigenschaft der Atome endlicher Graphen, *Arch. Math. (Basel)* 22 (1971), 333–336.

[MaBa03?] X. Marcote and C. Balbuena, Edge-connectivity and edge-superconnectivity of cages, submitted.

[MaBaPe03?] X. Marcote, C. Balbuena and I. Pelayo, On the connectivity of cages with girth five, six and eight, submitted.

[MaBaPeFa03?] X. Marcote, C. Balbuena, I. Pelayo, and J. Fàbrega, (δ, g)-cages with $g \geq 10$ are 4-connected, submitted.

[MaPeBa02] X. Marcote, I. Pelayo, and C. Balbuena, Every cubic cage is quasi 4-connected, *Discrete Math*, to appear.

[PeBaGo01] I. Pelayo, C. Balbuena, and J. Gómez, On the connectivity of generalized p-cycles, *Ars Combin.* 58 (2001), 215–231.

[PeLaHe86] K. Peters, R. Laskar, and S. Hedetniemi, Maximal/minimal connectivity in graphs, *Ars Comb.* 21 (1986), 59–70.

[Pl75] J. Plesník, Critical graphs of given diameter, *Acta Fac. Rerum Natur. Univ. Comenian. Math.* 30 (1975), 71–93.

[PlZn74] J. Plesník and Š. Znám, Strongly geodetic directed graphs, *Acta Fac. Rerum Natur. Univ. Comenian.*, Math. Publ. 29 (1974), 29–34.

[PlZn89] J. Plesník and Š. Znám, On equality of edge-connectivity and minimum degree of a graph, *Arch. Math. (Brno)* 25 (1989), 19–25.

[PluZh98] M. Plummer and X. Zha, On the connectivity of graphs embedded in surfaces, *J. Combin. Theory Ser. B* 72 (1998), 202–228.

[PluZh02] M. Plummer and X. Zha, On the connectivity of graphs embedded in surfaces II, *Electron. J. Comb.* 9, No.1, Research paper R38, 27p. (2002).

[ReKuHoLe82] S.M. Reddy, J.G. Kuhl, S.H. Hosseini, and H. Lee, On digraphs with minimum diameter and maximum connectivity, *Proc. 20th Annual Allerton Conference* (1982), 1018–1026.

[So92] T. Soneoka, Super edge-connectivity of dense digraphs and graphs, *Discrete Appl. Math.* 37/38 (1992), 511–523.

[SoNaIm85] T. Soneoka, H. Nakada, and M. Imase, Sufficient conditions for dense graphs to be maximally connected, *Proc. ISCAS85* (1985), 811–814.

[SoNaImPe87] T. Soneoka, H. Nakada, M. Imase, and C. Peyrat, Sufficient conditions for maximally connected dense graphs, *Discrete Math.* 63 (1987), 53–66.

[ToVo93] J. Topp and L. Volkmann, Sufficient conditions for equality of connectivity and minimum degree of a graph, *J. Graph Theory* 17 (1993), 695–700.

[Vo88] L. Volkmann, Bemerkungen zum p-fachen Kantenzusammenhang von Graphen, [Remarks on the p-fold edge connectivity of graphs.] *An. Univ. Bucure ti Mat.* 37 (1988), 75–79.

[Vo89] L. Volkmann, Edge-connectivity in p-partite graphs, *J. Graph Theory* 13 (1989), 1–6.

[Vo03] L. Volkmann, Degree sequence conditions for equal edge-connectivity and minimum degree, depending on the clique number, *J. Graph Theory* 42 (2003), 234–245.

[Wa67] M.E. Watkins, A lower bound for the number of vertices of a graph, *Amer. Math. Monthly* 74 (1967), 297.

[Wa70] M.E. Watkins, Connectivity of transitive graphs, *J. Combin. Theory* 8 (1970), 23–29.

[WaXuWa03] P. Wang, B. Xu, and J. Wang, A note on the edge-connectivity of cages, *Electron. J. Combin.* 10 (2003), # 2.

[Wo73] D.R. Woodall, The binding number of a graph and its Anderson number, *J. Combin. Theory Ser. B* 15 (1973) 225–255.

[Xu94] J.-M. Xu, A sufficient condition for equality of arc-connectivity and minimum degree of a digraph, *Discrete Math.* 133 (1994), 315–318.

[XuWaWa02] B. Xu, P. Wang, and J. Wang, On the connectivity of $(4, g)$-cages, *Ars Combin.* 64 (2002), 181–192.

GLOSSARY FOR CHAPTER 4

A-B **path**: see *path.*

$A|B$ **set**: see *separating set.*

adjacency matrix – of a graph (or digraph) G: the matrix $A = (a_{uv})$ indexed by the vertices of G with entries $a_{uv} = 1$ if uv is an edge (or arc) of G and $a_{uv} = 0$ otherwise.

algebraic connectivity – of a graph G: the smallest non-zero eigenvalue of the Laplacian matrix of G.

antisymmetric digraph (or oriented graph): a digraph such that between any pair of vertices there is at most one directed edge.

approximate (or approximation) algorithm: an algorithm that typically makes use of heuristics in reducing its computation but produces solutions that are not necessarily optimal.

arborescence: synonym for *rooted tree.*

associated symmetric digraph: see *digraph.*

asymmetric TSP (ATSP): see *TSP.*

atom – of a graph G: each minimum component obtained by removing a minimum disconnecting set from G.

balance condition – for a mixed graph G: for every $S \subseteq V(G)$, the difference between the number of arcs from S to $V(G)-S$ and the number of arcs from $V(G)-S$ to S is no greater than the number of undirected edges joining vertices in S and $V(G) - S$.

deBruijn graph of order k: a directed graph with 2^k vertices, each labeled with a unique k-bit string; vertex a is joined to vertex b by an arc if bitstring b is obtainable from bitstring a by either a cycle shift or a deBruijn shift; each arc is labeled by the first bit of the vertex at which it originates, followed by the label of the vertex at which it terminates.

balanced digraph: see *digraph.*

balanced orientation – of a graph (or mixed graph): an assignment of a direction to each edge of the graph (or each undirected edge of the mixed graph) so that the resulting digraph is balanced.

balanced vertex – in a digraph: a vertex whose indegree and outdegree are equal.

bipartite degree closure: bipartite graph of order $2n$ obtained by recursively joining pairs of non-adjacent vertices $x \in X$ and $y \in Y$ whose degree sum is at least $n + 1$, until no such pair remains.

bipartite graph: a graph G with two independent vertex subsets that partition $V(G)$.

bipartite index – of a graph: the smallest number of vertices whose removal leaves a bipartite graph.

block – in a connected graph: a maximal 2-connected subgraph.

boundary – of a vertex subset U: the set of vertices which are at distance one from U.

branching: synonym for *rooted tree.*

bridge – of a connected graph G: an edge whose deletion disconnects G; synonym for *cut-edge.*

(k,g)**-cage**: a regular graph of degree k and girth g with the minimum number of vertices.

Cayley (di)graph – of a group Γ with generating set S: a (di)graph whose vertices are identified with the elements of Γ, and there is an edge uv when $u^{-1}v \in \Gamma$.

Chinese Postman Problem: finding a postman tour of minimum length in a graph where all edges are undirected; see *postman*.

claw-free closure: the graph obtained by repeatedly applying the local completion until it is no longer possible to do.

clique number – of a graph G: the maximum number of vertices in a complete subgraph of G.

component$_1$ – of a graph: a maximal connected subgraph.

component$_2$ – of a digraph: a maximal strongly connected subdigraph.

connected graph: a graph in which there exists a walk between any pair of vertices.

___, **critically** k**-**: a graph G such that $\kappa(G) \geq k$ but, for each vertex $v \in V$, $\kappa(G-v) < k$.

___, **hamiltonian**: containing a spanning path between any two vertices.

___, k**-**: a graph with connectivity $\kappa \geq k \geq 1$.

___, k**-edge-**: a graph with edge-connectivity $\lambda \geq k \geq 1$.

contraction: an operation involving the identification (amalgamation) of vertices.

___, **edge**: given an edge uv, identification of its endpoints u and v (keeping the old adjacencies but removing the *self-loop* from $u = v$ to itself).

___, **subgraph**: identification of all the vertices of a given subgraph H by a succession of elementary contractions of the edges of H.

k**-contractible edge**: an edge of a k-connected graph whose contraction results in a k-connected graph.

k**-contractible subgraph**: a subgraph of a k-connected graph whose contraction results in a k-connected graph.

covering walk (or *postman tour*): – in an arbitrary graph G: a closed walk containing every edge of G.

critically k**-connected graph**: see *connected graph.*

cut-edge: synonym for *bridge.*

cut-vertex – of a connected graph: a vertex whose deletion disconnects it.

CVRP tour – in a weighted directed or undirected complete graph with vertex set $\{0, 1, \ldots, n\}$, with a demand $d_i \geq 0$ for $i = 1, 2, \ldots, n$, and two parameters Q and k: a collection of k cycles $C_1, C_2, \ldots, C_k$, which contain all the vertices, pairwise intersect only in vertex 0, and satisfy $\sum\limits_{i \in V(C_j)} d_i \leq Q$ for each $j = 1, 2, \ldots, k$.

cycle cover – of a graph G: a family S of cycles of G such that every edge of G belongs to at least one element of S.

cycle decomposition – of a graph (or digraph) G: a partition of the edge-set (or arc-set) of G such that each partition set forms a cycle (or directed cycle).

cycle double cover (CDC): a cycle cover S such that every edge of G belongs to exactly two elements of S.

cycle extendable: any cycle C of length $m < |V(G)|$ can be extended to a cycle of length $m + 1$ containing all of $V(C)$.

___, **fully**: a graph that is cycle extendable and having any vertex on a triangle.

cycle packing – in a graph G: a set of edge disjoint cycles in G.

cycle shift – $a_1a_2 \ldots a_k \longrightarrow b_1b_2 \ldots b_k$: a left shift such that $b_k = a_1$.

deBruijn shift – $a_1a_2 \ldots a_k \longrightarrow b_1b_2 \ldots b_k$: a left shift such that $b_k \neq a_1$.

bipancyclic: a bipartite graph of order $2n$ containing cycles of all even lengths from 4 to $2n$.

k-degree closure: graph obtained by recursively joining pairs of non-adjacent vertices whose degree sum is at least k, until no such pair remains.

degree sequence – of a graph G: the degrees of the vertices of G ordered in non-increasing (or non-decreasing) order.

detachment operation: see §4.2, Definition 12.

detachment – of a graph G: a graph that results from a sequence of detachment operations performed at each of the vertices of some vertex subset $W \subseteq V(G)$; used to transform or produce eulerian tours.

diameter – of a (di)graph G: the maximum distance between vertices of G.

digraph: a graph all of whose edges are directed; a *directed graph.*

___, **associated symmetric** – of a graph G: the digraph $\overset{\leftrightarrow}{G}$ obtained from G by replacing each edge uv by the two directed edges (u, v) and (v, u) forming a *digon.*

___, **balanced**: a digraph whose vertices are all balanced; in §4.3, this is called *symmetric digraph.*

___, **connected**: a digraph whose underlying graph is connected; also called *weakly connected.*

___, **critically k-connected**: a digraph G such that $\kappa(G) \geq k$ but, for each vertex $v \in V$, $\kappa(G - v) < k$.

___, **strongly connected**: a digraph with a u–v walk for any pair of vertices u, v; also called *strong digraph.*

___, **k-(strongly) connected**: a digraph with connectivity at least k; also called *k-strong* or *k-strongly connected.*

___, **symmetric$_1$**: a digraph such that between each pair of distinct vertices, either both (oppositely directed) arcs exist or neither does.

___, **symmetric$_2$**: (used in §4.3) a digraph such that indegree equals outdegree for each vertex.

disconnecting edge-set$_1$ – of a graph G: a subset of edges whose removal from G results into a non-connected graph.

disconnecting edge-set$_2$ – of a digraph G: a subset of arcs whose removal from G results into a non-strongly-connected digraph.

disconnecting (vertex-)set – of a graph G: a subset of vertices whose removal from G results into a non-connected graph.

___, **$(u|v)$-** – in a graph G: a disconnecting (vertex-)set whose removal from G leaves u and v in different components.

disconnecting (vertex-)set – of a digraph G: a subset of vertices whose removal from G results in a non-strongly-connected digraph.

***t*-distance connectivity** – of a graph (or digraph) G: the minimum cardinality of a $(u|v)$-set with $dist(u, v) \geq t$.

distance matrix of an instance of TSP: the matrix $D = [d_{ij}]$, where d_{ij} is the weight of the edge between vertices i and j; analogously defined for digraphs.

distance$_1$ – between vertices u, v in a graph (or digraph) G: the length of a shortest path joining u to v.

distance$_2$ – between vertex sets U, W in a graph G: the minimum of the distances between vertices of U and W.

distance-regular graph: a graph whose number of l-walks between any pair of vertices u, v only depend on $dist(u, v)$ and $l \geq 0$.

dominating circuit: A circuit C such that every edge of G is incident to a vertex of C.

double tracing: a closed walk that traverses every edge exactly twice.

___, **bidirectional**: a double tracing that uses every edge once in each of its two directions.

___, **strong**: a double-tracing that is both bidirectional and retract-free.

edge addition: given two non-adjacent vertices u, v of a graph $G = (V, E)$, $G + e = (V, E \cup \{uv\})$ is the graph obtained from G by addition of the edge $e = uv$.

edge contraction: see *contraction*.

edge-boundary – of a vertex subset U: the set of edges having exactly one endpoint in U.

edge-connectivity$_1$ – of a graph: the minimum number of edges whose removal leaves a connected graph; denoted λ (or κ_e).

edge-connectivity$_2$ – of a non-trivial digraph: the minimum number of arcs whose removal leaves a non-strongly connected graph; denoted λ or κ_e.

edge-cut: a *disconnecting edge-set*.

edge-disjoint paths: paths that have no edge in common.

edge-symmetric graph: a graph whose automorphism group acts transitively on its edge set.

Euclidean TSP: see *TSP*.

eulerian graph (or digraph or mixed graph): a graph that has an eulerian tour.

eulerian tour – in a graph (or digraph): is a closed walk that uses each edge (or arc) *exactly* once. An eulerian tour in a mixed graph is a closed walk that uses each edge and each arc exactly once.

even graph: an undirected graph whose vertices all have even degree.

exact algorithm: an algorithm that solves a certain optimization problem to optimality.

$\{F_1, F_2, \ldots, F_k\}$**-free**: containing no induced subgraph isomorphic to any F_i, $1 \leq i \leq k$.

1-factor – of graph G: a 1-regular subgraph H such that $V(H) = V(G)$ (i.e., spans G); thus, a 1-factor is a spanning *matching*.

2-factor of a graph G: the (vertex-disjoint) union of cycles of G that covers $V(G)$.

fragment – of a graph G: a component obtained from G by removing a minimum disconnecting set.

generalized p-cycle: a graph (or digraph) whose vertex set can be partitioned into p subsets, say $V_0, V_1, \ldots, V_{p-1}$ in such a way that every edge uv is of the form $u \in V_i$, $v \in V_{i+1}$ (arithmetic modulo p).

generalized TSP: see *TSP*.

s-geodetic digraph: see §4.7, Definition 45.

girth – of a graph G: the length of a shortest cycle.

half degree – of a vertex u in a digraph G: the indegree or outdegree of u in G.

hamiltonian connected: containing a spanning path between any two vertices.

hamiltonian cycle: a spanning cycle.

hamiltonian decomposition: a partitioning of the edge set of G into hamiltonian cycles if G is $2d$-regular or hamiltonian cycles and a perfect matching if G is $(2d+1)$-regular.

hamiltonian graph: a graph containing a hamiltonian cycle.

___, **k-ordered**: a hamiltonian graph such that for every ordered sequence of k vertices there is a hamiltonian cycle that encounters the vertices of the sequence in the given order.

___, **uniquely**: containing exactly one hamiltonian cycle.

imbedding – of a graph G on a surface S: a drawing of G on S with no crossing edges.

in-arcs – of a partition-cut $E(X, \overline{X})$: the subset of arcs whose head is in X; denoted $E^-(X, \overline{X})$.

incidence set – of a vertex v: the partition-cut $E(X, \overline{X})$, where $X = \{v\}$; denoted E_v (the out-arcs and in-arcs of E_v are denoted E_v^+ and E_v^-, respectively).

incidence-partition system: see §4.2, Definition 36.

indegree – of a vertex u in a digraph G: the number of vertices in G adjacent to u.

___, **t-**: the indegree of a vertex v with in-eccentricity at least t.

independence number: cardinality of a largest set of independent vertices.

independent set – of vertices: a subset of pairwise non-adjacent vertices.

independent vertices: vertices with no edges between them.

induced subgraph (or subdigraph) on a set $U \subset V$: the maximal subgraph (or subdigraph) of G with vertex set U; denoted $G(U)$ or $G[U]$.

in-eccentricity – of a vertex u in a digraph G: the maximum of the distances $dist(v, u)$ for all vertices v in G.

interior vertex – of a fragment F: a vertex with no adjacent vertices outside of F.

internally-disjoint u–v paths: u–v paths that have pairwise exactly vertices u and v in common.

in-tree: a rooted tree with all arcs reversed.

k-connected graph (or digraph): a graph (or digraph) with connectivity at least k.

k-edge-connected graph (or digraph): a graph (or digraph) with edge-connectivity at least k.

k-paths problem: given $2k$ distinct vertices $u_1, u_2, \cdots, u_k$ and $v_1, v_2, \cdots, v_k$ in a graph G, determining whether there exist k mutually edge-disjoint paths $P_1, P_2, \cdots, P_k$ in G such that P_i connects u_i and v_i for $i = 1, 2, \ldots, k$.

k-system that dominates: a collection of k edge-disjoint circuits and stars (with at least 3 endvertices), such that each edge of G is either contained in one of the circuits or stars, or is adjacent to one of the circuits.

kappa transformations: various combinations of splitting, splicing, and reversing closed trails; they form the basis for constructing eulerian tours and for transforming one tour into another; see §4.2.6.

Laplacian matrix – of a graph G: the matrix $L = (l_{uv})$ with entries $l_{uv} = deg(u) - 1$ if u, v are adjacent vertices of G, and $l_{uv} = 0$ otherwise.

line digraph – of a digraph G: is the digraph whose vertices are the directed edges of G and vertex (u, v) is adjacent to vertex (v, w).

___, **k-iterated** – of a digraph G: the digraph obtained by applying recursively k times the line digraph operation on G.

line graph – of a graph G: the graph whose vertices can be put into $1-1$ correspondence with the edges of G in such a way that two vertices of $L(G)$ are adjacent if and only if the corresponding edges of G are incident.

k-linked graph: a graph that has at least $2k$ vertices, and for every sequence of $2k$ different vertices, $u_1, u_2, \ldots, u_k, v_1, v_2, \ldots, v_k$, there exists a u_i–v_i path P_i, $i = 1, 2, \ldots, k$, such that the k paths are vertex-disjoint.

k-parity-linked graph: a graph in which one can find k disjoint paths with prescribed endvertices and prescribed parities of the lengths.

local completion – at a vertex x of a graph G such that the induced subgraph $G[N(x)]$ is connected: the graph obtained by replacing $G[N(x)]$ by a complete subgraph on $V(N(x))$.

matching: a subset of edges no two of which have a common vertex.

___, **perfect** – in a graph G: a matching such that every vertex in G is incident to some edge in the matching.

maximally connected graph (or digraph): a graph (or digraph) whose connectivity equals its minimum degree.

maximally edge-connected graph (or digraph): a graph (or digraph) whose edge-connectivity equals its minimum degree.

maximum degree – of a (di)graph G: the maximum of the (positive and negative) degrees of the vertices of G.

minimally k-connected graph (or digraph): a graph (or digraph) G such that $\kappa(G) \geq k$ but, for each edge $e \in E$, $\kappa(G - e) < k$.

minimally k-edge-connected graph (or digraph): a graph (or digraph) G such that $\lambda(G) \geq k$, but for each $e \in E$, $\lambda(G - e) < k$.

minimum (t-)degree – of a (di)graph G: the minimum among all t-(out and in)degrees of the vertices of G.

minimum degree – of a (di)graph G: the minimum of the (out and in) degrees of the vertices of G.

minimum degree – of a (di)graph G: the minimum of the (positive and negative) degrees of the vertices of G.

mixed graph: a graph that has both undirected and directed edges.

Moore bound – of a graph G: an upper bound for the maximum number of vertices, given its maximum degree and its diameter.

negative α-fragment – of a digraph G: the subset of edges whose negative edge-boundary is a minimum disconnecting edge set of G.

negative atom – of a digraph G: a negative fragment with minimum cardinality.

negative boundary – of a vertex subset F in a digraph G: the set of vertices in G which are at distance one to F.

negative edge-boundary – of a vertex subset F in a digraph G: the set of edges in G which have only its final vertex in F.

negative fragment – of a digraph G: the subset of vertices whose negative boundary is a minimum disconnecting set of G.

neighborhood$_1$ – of a vertex x: the set of all vertices adjacent to x.

neighborhood$_2$ – of the set S: vertices adjacent to some vertex in S.

out-arcs – of a partition-cut $E(X,\overline{X})$: the subset of arcs whose tail is in X; denoted $E^+(X,\overline{X})$.

outdegree – of a vertex u in a digraph: the number of vertices adjacent from u.

___, t-: the outdegree of a vertex v with out-eccentricity at least t.

out-eccentricity – of a vertex u in a digraph G: the maximum of the distances $dist(u,v)$ for all vertices v in G.

out-tree: synonym for *rooted tree.*

pancyclic: containing cycles of all lengths from 3 to $|V(G)|$.

p-partite graph: a graph whose vertex-set can be partitioned into p independent vertex subsets.

partition-cut – in a graph $G=(V,E)$ associated with $X \subset V(G)$: the set of edges in G with one endpoint in X and one endpoint in $\overline{X}=V(G)-X$; denoted $E(X,\overline{X})$.

path: a simple walk, that is, a walk in which all defining terms are distinct.

___, $A{-}B$: given $A,B \subset V$, an $u{-}v$ path P such that u is the only vertex of P belonging to A and v is the only vertex of P that belongs to B.

___, u-v – in a graph G: a path in G joining u to v; analogously defined for digraphs.

positive α-fragment – of a digraph G: the subset of edges whose positive edge-boundary is a minimum disconnecting edge set of G.

positive atom – of a digraph G: a positive fragment with minimum cardinality.

positive boundary – of a vertex subset F in a digraph G: the set of vertices in G that are at distance one from F.

positive edge-boundary – of a vertex subset F in a digraph G: the set of edges in G which have only its initial vertex in F.

positive fragment – of a digraph G: the set of vertices whose positive boundary is a minimum disconnecting set of G.

postman problem: the class of problems of finding a minimum-length (or minimum-weight) postman tour in a graph under various conditions.

___, **directed**: a postman tour of minimum length in a digraph.

___, **mixed**: a postman tour of minimum length in a mixed graph (with both directed and undirected edges).

___, **rural**: relaxation of the basic postman version (undirected, directed, mixed) where only a subset of edges have to be included at least once.

___, **stacker crane**: a rural postman version of mixed postman problem but where each directed edge is traversed at least once.

___, **undirected**: a postman tour of minimum length in a graph where all edges are undirected.

___, **windy**: an undirected postman problem where the cost of edge traversal depends on the direction the (undirected) edge is traversed.

postman tour (or *covering walk*): – in an arbitrary graph G: a closed walk containing every edge of G.

k-th power – of a connected graph G: is that graph with $V(G^k) = V(G)$ for which $uv \in E(G)$ if and only if $1 \leq d_G(u, v) \leq k$.

quasi 4-connected graph: a 3-connected graph G such that, for each vertex-cut $F \subset V$ with $|F| = 3$, F is the neighbourhood of a vertex of degree 3 and $G - F$ has exactly two components.

retract or *retracing* – in a walk W: a section of the walk of the form v_{i-1}, e_i, v_i, e_{i+1}, v_{i+1} such that $e_i = e_{i+1}$ (and $v_{i+1} = v_{i-1}$).

retract-free walk: a walk that has no retracts.

rooted tree: a directed tree having a distinguished vertex r, called the *root*, such that for every other vertex v, there is a directed r-v path. Occasionally encountered synonyms for rooted tree are *out-tree*, *branching*, and *arborescence.*

semigirth ℓ – of a digraph G: the parameter ℓ_π for $\pi = 0$.

semigirth, π- – of a digraph G: the greatest integer such that, for any pair of vertices u, v: (a) if $dist(u, v) < \ell$, the shortest u-v directed walk is unique and there are at most π u-v directed walks of length $dist(u, v) + 1$; (b) if $dist(u, v) = \ell$, there is only one shortest u-v directed walk; denoted ℓ_π.

separating set – of sets $A, B \subset V$: a set $X \subset V$ such that every A–B path in G contains a vertex of X.

(left) shift operation – converting a k-bit string $a = a_1 a_2 a_3 \cdots a_k$ to a k-bit string $b_1 b_2 \ldots b_k$: an operation such that $b_i = a_{i+1}$, for $i = 1, 2, \ldots, k - 1$.

spanning subgraph (or subdigraph) – of a graph G: a subgraph (or subdigraph) that contains all the vertices of G.

splitting operation: see §4.2, Definition 11.

stacker crane problem: a rural postman version of mixed postman problem but where each directed edge is traversed at least once.

strongly connected digraph: see *digraph.*

strongly regular graph: a connected distance-regular graph with diameter two.

subgraph contraction: see *contraction.*

super-κ graph: a maximally connected graph whose minimum disconnecting sets are the vertices adjacent some vertex of minimum degree; analogously defined for digraphs.

super-λ graph: a maximally edge-connected graph whose minimum disconnecting edge sets are the edges incident on some vertex of minimum degree; analogously defined for digraphs.

symmetric digraph$_1$: see *digraph.*

symmetric digraph$_2$: see *digraph.*

symmetric digraph$_3$: a digraph which is both vertex- and edge-transitive; used in §4.7.

symmetric graph: a graph which is both vertex- and edge-transitive.

symmetric TSP (STSP) – for a complete (undirected) weighted graph: finding a minimum-weight hamiltonian cycle in K_n.

total degree – of a vertex v in a mixed graph G: the total number of arcs and undirected edges incident on v.

totally separating set: Given $A \subset V$, a subset $X \subset V - A$ totally separates A if each component of $G - X$ has at most one vertex of A.

toughness – of a graph G: maximum t such that $t\ c(S) \leq |S|$, where $c(S)$ is the number of components of $G - S$.

traceable: containing a spanning path.

transition system: see §4.2, Definition 37.

traveling salesman problem: finding a minimum-weight hamiltonian cycle in a weighted complete graph. A non-complete graph is made complete by adding the missing edges and assigning to them a prohibitively large weight; also referred to as the *symmetric TSP.*

triangle inequality – on a weighted digraph (or graph): the condition $d_{ij} + d_{jk} \geq d_{ik}$ for all distinct vertices i, j, k, where d_{ij} is the weight of the arc from vertex i to vertex j (of the edge between i and j).

TSP: *traveling salesman problem.*

___, **asymmetric (ATSP)**: finding a minimum-weight hamiltonian cycle in a weighted *complete digraph* $\overset{\leftrightarrow}{K}_n$ (every pair of vertices has both oppositely directed arcs).

___, **Euclidean**: the special case of TSP in which the vertices are points in the Euclidean plane and the weight on each edge is the Euclidean distance between its endpoints.

___, **generalized** – for a weighted complete digraph $\overset{\leftrightarrow}{K}_n$ and a partition $V_1, \ldots, V_k$ of its vertices: finding a minimum-weight cycle containing exactly one (at least one) vertex from each set V_i, $i = 1, \ldots, k$.

$(u|v)$**-(disconnecting) set**: see *disconnecting set.*

u–v **path**: a path from vertex u to vertex v.

vehicle routing problem, capacitated (CVRP) – in a weighted directed or undirected complete graph on $n+1$ vertices with a demand $d_i \geq 0$ for $i = 1, 2, \ldots, n$ and with two parameters Q and k: finding a CVRP tour for which the total weight of the cycles is minimum.

(vertex-)connectivity$_1$ – of a graph: the minimum number of vertices whose removal leaves a non-connected or trivial graph; denoted κ or κ_v.

(vertex-)connectivity$_2$ – of a strongly connected digraph: the minimum number of vertices whose removal leaves a non-strongly connected or trivial digraph; denoted κ or κ_v.

vertex splitting: operation in which a vertex w is replaced by an edge uv in such a way that some of the vertices adjacent to w are now adjacent to u and the rest are adjacent to v.

vertex-symmetric graph: a graph whose automorphism group acts transitively on its vertex set; that is, for any pair of vertices u, v, there exists an automorphism of G mapping u to v.

walk: an alternating sequence of vertices and edges such that for each edge, one endpoint precedes and the other succeeds that edge in the sequence.

weakly k-linked graph: a graph that has at least $2k$ vertices, and for every k pairs of vertices (u_i, v_i), there exists a u_i–v_i path P_i, $1 \leq i \leq k$, such that the k paths are edge-disjoint.

weakly connected digraph: a digraph whose underlying graph is connected; also called *connected digraph.*

wheel: a graph consisting of a cycle C and an additional vertex that is adjacent to every vertex of C.

Chapter 5

COLORINGS and RELATED TOPICS

5.1 GRAPH COLORING

Zsolt Tuza, Hungarian Academy of Sciences
University of Veszprém, Hungary

5.2 FURTHER TOPICS in GRAPH COLORING

Zsolt Tuza, Hungarian Academy of Sciences
University of Veszprém, Hungary

5.3 INDEPENDENT SETS and CLIQUES

Gregory Gutin, Royal Holloway, University of London, UK

5.4 FACTORS and FACTORIZATION

Michael Plummer, Vanderbilt University

5.5 PERFECT GRAPHS

Alan Tucker, SUNY at Stony Brook

5.6 APPLICATIONS to TIMETABLING

Edmund Burke, University of Nottingham, UK
Dominique de Werra, École Polytechnique Fédérale
de Lausanne, Switzerland
Jeffrey Kingston, University of Sydney, Australia

GLOSSARY

5.1 GRAPH COLORING

Zsolt Tuza, Hungarian Academy of Sciences
University of Veszprém, Hungary

5.1.1 General Concepts
5.1.2 Vertex Degrees
5.1.3 Critical Graphs and Uniquely Colorable Graphs
5.1.4 Girth and Clique Number
5.1.5 Edge-Coloring and χ-Binding Functions
5.1.6 Coloring and Orientation
5.1.7 Colorings of Infinite Graphs
References

Introduction

§5.1 concentrates on the classical concept of *chromatic number* and on the more recent but closely related concept of *choice number*, mostly in connection with other important graph invariants. Further developments of graph colorings appear in §5.2.

Various problems, some of which are equivalent to colorings (e.g., 1-factorizations) are dealt with in the other sections of this chapter. The book [JeTo95] is a wealthy source of additional information, where results are organized around more than 200 open problems. On list coloring and related topics, a comprehensive survey can be found in [Tu97] and its update [KrTuVo99].

TERMINOLOGY NOTE: We consider a *graph* to be without self-loops to have at least one vertex, and, except within the last subsection, to be finite.

5.1.1 General Concepts

Graph coloring deals with the general and widely applicable concept of partitioning the underlying set of a structure into parts, each of which satisfies a given requirement (e.g., to be an independent set). One of the most famous problems in this area, and even within graph theory, what is now known as the Four Color Theorem (see §5.2.2), has been a driving force of research on graphs for nearly a century.

Proper Vertex-Coloring and Chromatic Number

DEFINITIONS

D1: A ***vertex-coloring*** of a graph $G = (V, E)$ is a function

$$\varphi : V \to \mathcal{C}$$

from the set of vertices to a set $\mathcal{C}$ of "colors".

D2: The coloring φ is ***proper*** if no two adjacent vertices are assigned the same color.

D3: A ***k-coloring*** is a vertex-coloring with at most k colors.

NOTATION: If k is a positive integer, we assume (unless specified otherwise explicitly) that $\mathcal{C} = \{1, 2, \ldots, k\}$.

D4: A proper k-coloring φ may also be viewed as a *vertex partition*

$$V_1 \cup \cdots \cup V_k = V$$

where the disjoint subsets $V_i = \varphi^{-1}(i)$ are called the ***color classes***. (Saying that a k-coloring φ is proper is equivalent to assuming that $\varphi^{-1}(i)$ is an *independent set* for each i.)

D5: A graph is ***k-colorable*** if it admits a proper vertex-coloring with at most k colors.

D6: A graph is ***k-chromatic*** if it is k-colorable but not $(k-1)$-colorable.

D7: The ***chromatic number*** of a graph G, denoted $\chi(G)$, is the smallest nonnegative integer k such that G is k-colorable.

EXAMPLES

E1: The complete graph K_n on n vertices has chromatic number n.

E2: A graph G (other than the null graph) has chromatic number

$$\begin{aligned} \chi(G) = 1 &\quad \text{if and only if } G \text{ is edgeless;} \\ \chi(G) \leq 2 &\quad \text{if and only if } G \text{ is bipartite.} \end{aligned}$$

In particular, cycles of even length are 2-chromatic, while cycles of odd length are 3-chromatic.

FACTS

F1: An immediate consequence of the definitions is that

$$\chi(G) \geq \frac{|V|}{\alpha(G)}$$

for every graph G, where $\alpha(G)$ denotes the *(vertex) independence number.*

F2: [Bo88] The lower bound $\chi(G) \geq \frac{|V|}{\alpha(G)}$ is almost tight for almost all graphs, since the *random graph* $\mathbf{G}_{n,p}$ satisfies

$$\chi(\mathbf{G}_{n,p}) = \left(\frac{1}{2} + o(1)\right)\left(\log\frac{1}{1-p}\right)\frac{n}{\log n}$$

for every fixed p $(0 < p < 1)$ with probability $1 - o(1)$ as $n \to \infty$.

F3: Let $f : \mathbb{R}^{\geq 0} \to \mathbb{R}^{+}$ be a nondecreasing, continuous function with $f(0) = 1$. If $G = (V, E)$ is a graph such that, for every $1 \leq i \leq |V|$, every subgraph of G on i vertices has independence number at least $f(i)$, then

$$\chi(G) \leq \int_0^{|V|} \frac{1}{f(x)} dx$$

List Coloring and Choice Number

Many results on the chromatic number can be discussed in the following more general setting.

DEFINITIONS

D8: A (vertex) ***list assignment*** L on a graph G associates a set L_v of colors with each vertex v of G. Each L_v is interpreted as the set of allowed colors for vertex v.

D9: The graph G is ***L-colorable*** (or ***list colorable***, when L is understood from context) if it admits a proper vertex-coloring φ such that $\varphi(v) \in L_v$ for all v.

TERMINOLOGY NOTE: The term "list" is used in the literature for historical reasons only. No particular ordering on the color set L_v is assumed here.

D10: If $|L_v| = k$ for all $v \in V$, then the list assignment L is called a ***k-assignment***.

D11: A graph G is ***k-choosable*** if it is L-colorable for every k-assignment L.

D12: The ***choice number*** of G, denoted $ch(G)$, is the smallest nonnegative integer k such that G is k-choosable. (In part of the literature, the choice number is called ***list chromatic number***, and also the notation $\chi_\ell(G)$ is commonly used for $ch(G)$.)

EXAMPLES

E3: The complete graph K_n on n vertices has choice number n.

E4: The choice number may be strictly larger than the chromatic number. In particular, the complete bipartite graphs $K_{3,3}$ and $K_{2,4}$ are 2-colorable but not 2-choosable (the former with the lists $(1,2),(1,3),(2,3)$ in both bipartition subsets, and the latter with lists $(1,2),(3,4)$ and $(1,3),(1,4),(2,3),(2,4)$ in its 2- and 4-element class, respectively). It is true, however, that cycles of even length are 2-choosable, while cycles of odd length have choice number 3 (although the former is not immediate to verify).

FACTS

F4: [ErRuTa79] The complete bipartite graph $K_{n,n}$ has choice number $(1+o(1))\log_2 n$ as $n \to \infty$.

F5: (J. Kahn in [Al93]; [TuVo94]) Asymptotically $ch(\mathbf{G}_{n,p}) = (1+o(1))\,\chi(\mathbf{G}_{n,p})$ for every constant p $(0 < p < 1)$ with probability $1 - o(1)$ as $n \to \infty$.

F6: [Kr00] If the edge probability $p = p(n)$ is such that $p(n) \to 0$ as $n \to \infty$ and $p(n) \geq n^{-1/4+\varepsilon}$ for some $\varepsilon > 0$, then $ch(\mathbf{G}_{n,p}) = (1+o(1))\,\chi(\mathbf{G}_{n,p})$ with probability $1 - o(1)$.

F7: [AlKrSu99, Vu99] The bound $ch(\mathbf{G}_{n,p}) = O(\chi(\mathbf{G}_{n,p}))$ holds also for $p(n) > 2/n$; and upper bounds of the form $O(n\,p(n)/\log(n\,p(n)))$ follow deterministically, too, from conditions (analogous to the expected values of parameters in $\mathbf{G}_{n,p}$) on degrees, pair-degrees, and complementary degrees.

The Hajós Construction

FACTS

F8: [Ha61] Every graph of chromatic number at least k can be constructed from the complete graph K_k by a sequence of operations of the following three types:

> (1) Insert new vertices and/or edges.
>
> (2) Having constructed vertex-disjoint graphs G_1 and G_2, select edges u_iv_i in G_i $(i = 1, 2)$, remove u_1v_1 and u_2v_2, identify u_1 with u_2, and insert the new edge v_1v_2.
>
> (3) Identify nonadjacent vertices.

F9: [Gr96] Every graph of choice number at least k can be constructed from any one complete bipartite graph of choice number k by a sequence of the operations (1) and (2) above, and the following third type:

> (3′) Having constructed a graph $G = (V, E)$ that has an uncolorable list assignment L where $|L_v| \geq k$ for all $v \in V$ and two nonadjacent vertices u, v have the same list in L, then identify u with v.

Lovász's Topological Lower Bound

DEFINITION

D13: The ***neighborhood complex*** of a graph $G = (V, E)$ is the simplicial complex $\mathcal{N}(G)$ whose vertices are the vertices of G, and such that the set $X \subseteq V$ is a simplex if all the $x \in X$ have a common neighbor in G.

FACT

F10: [Lo78] If the neighborhood complex $\mathcal{N}(G)$ of a graph G is a k-connected topological space, then $\chi(G) \geq k + 3$.

Alon and Tarsi's Graph Polynomial Characterization

DEFINITION

D14: The ***graph polynomial***, also called the ***edge difference polynomial***, of a graph $G = (V, E)$ with $E \neq \emptyset$ and vertex set $V = \{v_1, \ldots, v_n\}$ is

$$P_G = P_G(x_1, \ldots, x_n) := \prod_{\substack{i<j \\ v_iv_j \in E}} (x_i - x_j)$$

Given a list assignment L on G, we also define

$$Q_i = Q_i(x_i) := \prod_{q \in L_i} (x_i - q)$$

for $i = 1, \ldots, n$.

FACT

F11: [AlTa92] A graph G admits a list L-coloring if and only if its graph polynomial P_G does not belong to the ideal generated by the polynomials Q_i.

List Reduction

FACT

F12: [TuVo97] Let $G = (V, E)$ be a graph, let L be a k-assignment on G ($k \geq 2$), let $X \subset V$ be any subset, and let $F \subseteq E$ be the set of those edges which have at least one vertex in X. If (V, F) is a 2-choosable graph, then one can properly color the set X with colors $\varphi(x) \in L_x$, for all $x \in X$, in such a way that at most one color occurs in the neighborhood of each vertex $v \in V \setminus X$. In particular, $ch(G) \leq ch(G - X) + 1$.

5.1.2 Vertex Degrees

DEFINITIONS

D15: The (Erdős–Hajnal) ***coloring number*** of G, denoted $col(G)$ (and also called the ***Szekeres–Wilf number***) is the smallest positive integer k for which there exists an order $v_1 v_2 \cdots v_n$ of the vertices, such that every v_i has fewer than k neighbors v_j with $j > i$.

D16: A graph is ***d-degenerate*** if none of its subgraphs has minimum degree larger than d. ($col(G)$ is the smallest integer k such that G is $(k-1)$-degenerate.)

FACTS

F13: For every graph G,

$$\omega(G) \leq \chi(G) \leq ch(G) \leq col(G) \leq \Delta(G) + 1$$

where $\omega(G)$ denotes the *clique number* of G.

F14: (Nordhaus–Gaddum Theorem) [NoGa56] For any graph $G = (V, E)$ and its edge-complement $\overline{G}$,

$$\chi(G) + \chi(\overline{G}) \leq |V| + 1$$

F15: [ErRuTa79] The Nordhaus–Gaddum upper bound also holds for list colorings:

$$ch(G) + ch(\overline{G}) \leq |V| + 1 \quad \text{and} \quad col(G) + col(\overline{G}) \leq |V| + 1$$

F16: [HaSz70] Every graph $G = (V, E)$ with maximum degree Δ admits a proper $(\Delta + 1)$-coloring such that each color class has cardinality $\left\lfloor \frac{|V|}{\Delta+1} \right\rfloor$ or $\left\lceil \frac{|V|}{\Delta+1} \right\rceil$.

F17: (Brooks's Theorem) [Br41] If the graph G is connected, then

$$\chi(G) \leq \Delta(G)$$

unless G is complete or $\Delta(G) = 2$ and G is an odd cycle.

F18: [Vi76, ErRuTa79] (analogue to Brooks's theorem for list colorings) If the graph G is connected, then

$$ch(G) \leq \Delta(G)$$

unless G is a complete graph or $\Delta(G) = 2$ and G is an odd cycle. (See also §5.2.1 for generalizations.)

REMARK

R1: Concerning the sequence of inequalities $\omega \leq \chi \leq ch \leq col \leq \Delta + 1$, Brooks's theorem and its analogue characterize equality in $\chi = \Delta + 1$ and $ch = \Delta + 1$. The other end, $\omega = \chi$ is studied in the theory of perfect graphs (see §5.5). So far, the problems [Tu97a] of finding tight conditions for ensuring $\omega = ch$ or $\chi = ch$ are open, except for $ch = 2$.

FACTS

F19: [ErRuTa97] A connected graph is 2-choosable if and only if the sequential removal of degree-1 vertices yields the trivial graph K_1, or an even cycle, or an even cycle plus a degree-2 vertex whose two neighbors are at distance two along the cycle.

F20: [Al00] There exists a sequence of real numbers ϵ_d with $\epsilon_d \to 0$ as $d \to \infty$, such that the inequality $ch(G) \geq (\frac{1}{2} - \epsilon_d)\log_2 n$ holds whenever the graph G on n vertices contains a subgraph of minimum degree d. Equivalently, $ch(G) \geq (\frac{1}{2} - o(1))\log_2 col(G)$ as $col(G)$ gets large.

5.1.3 Critical Graphs and Uniquely Colorable Graphs

The two extremes are considered here: k-chromatic graphs that are almost $(k-1)$-colorable, and k-chromatic graphs with just one proper k-coloring.

DEFINITION

D17: For $k \geq 2$, a k-chromatic graph $G = (V, E)$ is ***k*-critical** if

$$\chi(G - e) = k - 1$$

for every edge $e \in E$; and G is ***k*-vertex-critical** if

$$\chi(G - v) = k - 1$$

for every vertex $v \in V$.

FACTS

F21: Every k-vertex-critical graph is connected, without any vertex cutsets inducing a complete subgraph, and contains a k-critical spanning subgraph.

F22: [To78] A graph G is vertex-critical if and only if the complement of each block in the complementary graph $\overline{G}$ is vertex-critical.

F23: For $k = 2$ and $k = 3$, a graph is "k-critical" if and only if it is "k-vertex-critical". The unique 2-critical graph is K_2, and the 3-critical graphs are the odd cycles.

F24: (same as Brooks's Theorem) Every k-critical graph G has minimum degree at least $k - 1$, and if G is $(k-1)$-regular, then either $G = K_k$ or $k = 3$ and G is an odd cycle.

F25: [Mi92] Every k-critical graph $G \neq K_k$ $(k \geq 4)$ contains all trees of k edges as subgraphs.

F26: [Ga63] If G is k-critical, then its vertices of degree $k-1$ induce a subgraph in which every block is a complete graph or an odd cycle. Conversely, if H is a K_k-free graph of maximum degree at most $k-1 \geq 3$ where each block is a complete graph or an odd cycle, then there exists a k-critical graph G in which the vertices of degree $k-1$ induce a subgraph isomorphic to H. (This graph H may also be the null graph.)

F27: [St85] For $k \geq 4$, let F be a connected graph of maximum degree at most $k-1$, such that each of its blocks is a complete graph or an odd cycle. There exist only finitely many k-critical graphs in which the vertices of degree $k-1$ induce F and in which the other vertices induce either K_{k-1} or a $(k-2)$-colorable graph.

F28: [GrLo74] A graph H is a proper subgraph of some k-critical graph if and only if H and each of its edge contractions H/e (for $e \in E(H)$) is $(k-1)$-colorable.

F29: [Si72, To72] There exists an infinite sequence of 4-critical graphs on n vertices with minimum degree at least $\frac{1}{2}n^{1/3}$. (It is not known whether the minimum degree can be as large as cn in 4-critical or 5-critical graphs.)

F30: [Di57] A k-critical graph ($k \geq 4$) on $n > k$ vertices has at least $\frac{k-1}{2}n + \frac{k-3}{2}$ edges.

F31: [Ga63] A k-critical graph ($k \geq 4$) on $n > k$ vertices has at least $\frac{k-1}{2}n + \frac{k-3}{2(k^2-3)}n$ edges.

F32: [Lo73] For the largest possible independence number $\alpha_{n,k}$ in k-critical graphs on n vertices, the difference $n - \alpha_{n,k}$ is at least $\frac{1}{6}kn^{1/(k-2)}$ and, for infinitely many values of n, at most $2kn^{1/(k-2)}$.

F33: [RöTu85] Let $t(k,\ell)$ denote the minimum number of edges in graphs with k vertices and independence number less than ℓ (i.e., the complement of the Turán number, cf. §8.1). If G is $(k+1)$-critical and has at least $2t(k,\ell)-1$ vertices, then in order to obtain an ℓ-colorable subgraph, one must delete at least $t(k,\ell)$ edges. This bound is tight. (For unrestricted $(k+1)$-chromatic graphs, the minimum is $t(k+1,\ell+1)$.)

F34: [RöTu85] Let $k \geq 2$, and G be a graph with n vertices and m edges. Suppose that the automorphism group of G acts transitively, either

(a) on the vertices and $\chi(G) \geq 2k+1$, or

(b) on the edges and $\chi(G) \geq k^2+1$.

Then, to obtain a k-colorable subgraph of G, one has to delete at least $\sqrt{n}$ vertices in Case (a) and $\sqrt{m}$ edges in Case (b). (It is not known whether vertex/edge transitive 4-chromatic graphs can always be made bipartite by removing a bounded number of vertices/edges.)

F35: (V. Rödl in [To85]) The number of nonisomorphic 4-critical graphs is at least c^{n^2}, for some constant $c > 1$.

OPEN PROBLEMS

P1: Determine tight asymptotic bounds on the minimum and maximum numbers of edges in k-critical graphs on n vertices ($k \geq 4$ fixed, $n \to \infty$).

P2: (J. Nešetřil and V. Rödl) For $k \geq 3$, does there exist a function $f_k(n)$ such that every $(k+1)$-critical graph with at least $f_k(n)$ vertices contains a k-critical subgraph with at least n vertices?

FACT

F36: [KeKe54] The function $f_3(n)$ exists, i.e., every 'large' 4-critical graph contains a 'long' odd cycle.

EXAMPLES

E5: The complete join of two disjoint cycles of length $2t+1$ is a 6-critical graph on $n = 4t+2$ vertices with $\frac{1}{4}n^2 + n$ edges ([Di52]). More generally, the number of edges in k-critical graphs on n vertices can be as large as $(\frac{1}{2} - c_k)n^2$, where $c_k \to 0$ as $k \to \infty$.

E6: On $n = 8t+4$ vertices, let V_i $(1 \leq i \leq 4)$ be disjoint vertex subsets of cardinality $2t+1$ each, V_1 and V_2 induce the cycle C_{2t+1}, V_3 and V_4 be independent and completely joined, and let the $V_1 - V_3$ edges and the $V_2 - V_4$ edges form a perfect matching. This graph is 4-critical, with more than $n^2/16$ edges ([To70]).

Uniquely Colorable Graphs

DEFINITION

D18: A graph $G = (V, E)$ is ***uniquely k-colorable*** if it admits just one proper k-coloring with $k = \chi(G)$ colors (apart from the renumbering of colors).

The uniquely 1-colorable graphs are the edgeless graphs, and therefore we assume below that $k \geq 2$.

EXAMPLES

E7: The uniquely 2-colorable graphs are the connected bipartite graphs.

E8: If $G - v$ is a uniquely k-colorable graph, and v a vertex of degree k with one neighbor in each color class of $G - v$, then also the graph G is uniquely k-colorable. In particular, uniquely 4-colorable planar graphs are obtained from a planar embedding of K_4 by sequentially inserting new degree-3 vertices inside triangular faces (where the neighbors of each new vertex are the vertices of the corresponding face).

FACTS

F37: [ChGe69] Every uniquely k-colorable graph is $(k-1)$-connected.

F38: [Bo78] If the graph G is k-colorable on n vertices, with minimum degree larger than $\frac{3k-5}{3k-2}\,n$, then G is uniquely k-colorable. Moreover, this lower bound is best possible for infinitely many n.

F39: [Bo78] Let the graph G be k-colorable on n vertices, with a k-coloring where the union of any two color classes is connected. If the minimum degree is larger than $\frac{k-2}{k-1}\,n$, then G is uniquely k-colorable. Moreover, this lower bound is best possible for infinitely many n.

5.1.4 Girth and Clique Number

The results below show that the exclusion of short cycles does not make the chromatic number bounded; but on graphs without large complete subgraphs, the general upper bounds in terms of vertex degrees can be improved.

FACT

F40: [Er59] For a suitably chosen constant c, and for every $k > 1$ and $g > 2$ there exists a non-k-colorable graph of girth g on at most $c\,k^{2g}$ vertices. Moreover, for $k \geq 4$, the maximum girth of k-chromatic graphs on n vertices grows with $\Theta(\frac{\log n}{\log k})$ as $n \to \infty$.

REMARK

R2: The proof in [Er59] is probabilistic. Constructions usually are of much larger size than guaranteed by Erdős's theorem. For g small, some examples are listed below; for unrestricted g, see, e.g., [Lo68, NeRö79]. A fairly small general construction, involving Ramanujan graphs on $O(k^{4g})$ vertices, can be found in [LuPhSa88].

EXAMPLES

E9: [Zy49] If G_k is a k-chromatic triangle-free graph, let G_{k+1} consist of k vertex-disjoint copies G_k^i of G_k $(i = 1, \ldots, k)$, together with a new independent set X of size $|V(G_k)|^k$. To each k-tuple $(v_1, \ldots, v_k) \in V(G_1) \times \cdots \times V(G_k)$, join a distinct common neighbor in X. This G_{k+1} is $(k+1)$-chromatic and triangle-free.

E10: [De48] If G_k is a k-chromatic graph of girth at least 6, take an independent set X of cardinality $s = k|V(G_k)| - k + 1$, together with $m = \binom{s}{|V(G_k)|}$ vertex-disjoint copies G_k^i of G_k $(i = 1, \ldots, m)$. For each of the m distinct $|V(G_k)|$-tuples $Y_i \subset X$, draw a perfect matching between Y_i and $V(G_k^i)$. The resulting graph G_{k+1} is a $(k+1)$-chromatic graph of girth 6.

E11: [My55] For each vertex v of a k-chromatic triangle-free graph G_k, take a distinct new vertex v' adjacent to all neighbors of v, and join a new vertex w to all of these v'. The graph obtained is $(k+1)$-chromatic and triangle-free.

E12: (Kneser graphs) [Lo78, Bá78] The vertices of the Kneser graph $K(n,k)$ are the k-element subsets of $\{1, \ldots, n\}$ $(n/2 > k > 1)$, and two vertices are adjacent if and only if the corresponding two k-sets are disjoint. Then $\chi(K(n,k)) = n - 2k + 2$, and $K(n,k)$ is triangle-free if $n < 3k$. Moreover, a vertex-critical subgraph of $K(n,k)$ is induced by the vertices corresponding to the k-tuples that have no pair of consecutive elements in the cyclic order $\{1, 2, \ldots, n\}$ ([Sc78]).

CONJECTURE

C1: [Re97] For every graph G,

$$\chi(G) \leq \frac{1}{2}(\Delta(G) + \omega(G)) + 1$$

FACTS

F41: [Re97] For every m there is a number Δ_m such that, if $\Delta(G) \geq \Delta_m$ and $\omega(G) \leq \Delta(G) + 1 - 2m$, then $\chi(G) \leq \Delta(G) + 1 - m$. Moreover, there is a constant $\epsilon > 0$ such that if $\Delta(G) \geq 3$, then $\chi(G) \leq (1 - \epsilon)(\Delta(G) + 1) + \epsilon\,\omega(G)$.

F42: Combining Fact 3 with the known estimates [AjKoSz80] on the Ramsey numbers $R(s,t)$, it follows that if $\omega(G) \leq t$, then for some constant $c = c(t)$, we have

$$\chi(G) \leq c \left(\frac{n}{\log n}\right)^{1-1/t}$$

F43: [BoKo77, Ca78, La78] Let $t \geq 3$. If $\omega(G) \leq t \leq \Delta(G)$, then

$$\chi(G) \leq \frac{t}{t+1}\,(\Delta(G)+2)$$

F44: [Er67] The maximum ratio in the set

$$\left\{ \frac{\chi(G)}{\omega(G)} \;\middle|\; |V_G| = n \right\}$$

grows with $\Theta(n/\log^2 n)$ as $n \to \infty$.

F45: [Jo96a] For every $r \in \mathbb{N}$ there exists a constant c_r such that if the graph G is K_r-free, then

$$ch(G) \leq \frac{c_r\,\Delta(G)\log\log\Delta(G)}{\log\Delta(G)}$$

F46: [Jo96] If the graph G is triangle-free, then $ch(G) \leq c\,\Delta(G)/\log\Delta(G)$, for some constant c independent of $\Delta(G)$.

F47: [Ki95] If the graph G has girth at least 5, then

$$ch(G) \leq \frac{(1+\epsilon_{\Delta(G)})\,\Delta(G)}{\log\Delta(G)}$$

where $\epsilon_\Delta \to 0$ as $\Delta \to \infty$.

F48: [KoMa77] For every integer $\Delta > 3$, there exists a triangle-free graph G with maximum degree Δ and $\chi(G) > \frac{\Delta}{2\log\Delta}$.

F49: [Br02] Let $G = (V, E)$ be a maximal triangle-free d-regular graph, with $d > |V|/3$. Then $\chi(G) \leq 4$; moreover, if G has a nontrivial automorphism, then $\chi(G) \leq 3$.

F50: [Ko78] For every Δ there is a $g = g(\Delta)$ such that $\chi(G) \leq \frac{1}{2}\Delta + 2$ whenever G has girth at least g and maximum degree at most Δ.

F51: [MiSc97] If the cycles in G have at most p distinct even lengths and at most q odd lengths, then $\chi(G) \leq \min\{2p+3, 2q+2\}$. (The bound is tight for all p and q.)

F52: [RaSc01] If every induced cycle of G has length 4 or 5, then $\chi(G) \leq 3$. (This generalizes the fact [Su81] that every triangle-free graph without induced P_6 and C_6 is 3-colorable.) Also, all pairs F_1, F_2 of graphs with the property that every graph without induced F_1 and F_2 is 3-colorable can be characterized ([Ra02]).

CONJECTURE

C2: (Erdős–Faber–Lovász Conjecture) The union of any n edge-disjoint copies of K_n has chromatic number n.

FACT

F53: [Ka92] If G is the edge-disjoint union of n complete graphs of n vertices, then $\chi(G) = n + o(n)$ and also $ch(G) = n + o(n)$ as $n \to \infty$.

The Conjectures of Hadwiger and Hajós

CONJECTURES

C3: (Hadwiger's Conjecture) [Ha43] From every k-chromatic graph, the complete graph K_k can be obtained by a sequence of edge contractions and vertex/edge deletions.

C4: (Hajós's Conjecture) For $k \leq 6$, every k-chromatic graph contains a subdivision of the complete graph K_k, i.e., k vertices mutually adjacent by internally disjoint paths.

FACTS

F54: [Di52] Both Hadwiger's and Hajós's conjectures are true for $k = 4$.

F55: [Ca79] If $\chi(G) = 4$, then G contains a subdivision of K_4 where each of the four cycles corresponding to the triangles of K_4 have odd length. (It is not known whether G also contains a subdivision of K_4 where each of the six paths obtained from the edges of K_4 have odd length.)

F56: [Za98] There is a polynomial-time algorithm that either properly colors an input graph G with 3 colors or outputs an odd-triangle subdivision of K_4 in G.

F57: [Ca79] For every $k \geq 7$ there exist graphs that are k-chromatic but do not contain any subdivision of K_k.

F58: [Wa37] The 4-colorability of all planar graphs implies the 4-colorability of all graphs not contractible to K_5. Hence, for $k = 5$, Hadwiger's conjecture (stated later than Wagner's theorem) is equivalent to the Four Color Theorem on planar graphs (cf. §5.2.2).

F59: [RoSeTh93] For $k = 6$, the validity of Hadwiger's conjecture can be deduced from the Four Color Theorem.

F60: [BoCaEr80, Ko82] For the random graph on n vertices, Hadwiger's conjecture is valid with probability $1 - o(1)$ as $n \to \infty$.

F61: [Ko84] Let $h(G)$ denote the largest number of vertices in a complete graph to which G can be contracted. Then the inequality $h(G) + h(\overline{G}) \leq \lfloor 6n/5 \rfloor$ holds for every graph G on $n \geq 5$ vertices, and the upper bound is tight.

See also §5.2.1 for a related result.

5.1.5 Edge-Coloring and χ-Binding Functions

TERMINOLOGY NOTE: In this subsection we shall explicitly use the term ***multigraph*** in those cases where multiple edges are allowed.

DEFINITIONS

D19: A ***proper edge-coloring*** of a graph or multigraph G is an assignment of colors to the edges of G, such that all edges incident with the same vertex get distinct colors.

D20: The ***chromatic index*** of a graph or multigraph G, denoted $\chi'(G)$, is the smallest number of colors in a proper edge-coloring of G.

D21: Definitions 19 and 20 for edge-colorings can be generalized to list edge-colorings in the natural way, as for vertex-colorings. The ***edge choice number*** (or ***list chromatic index*** or ***list edge chromatic number***) is the minimum list-size that guarantees a list edge-coloring of G; it is denoted by $ch'(G)$ (or by $\chi'_\ell(G)$).

D22: A ***proper total coloring*** of $G = (V, E)$ is an assignment φ of colors to the vertices and the edges of G, such that φ induces a proper coloring on both V and E, and such that $\varphi(e) \neq \varphi(v)$ whenever $e \in E$ is incident with $v \in V$.

D23: The smallest number of colors in a proper total coloring of G is denoted by $\chi''(G)$; and the analogous quantity for the smallest size of lists is denoted by $ch''(G)$ (or by $\chi''_\ell(G)$).

D24: Let $G = (V, E)$ be a graph or multigraph.

- The ***line graph*** $L(G)$ of G has as its vertices the edges of G, two of them being adjacent in $L(G)$ if they share a vertex in G.
- The ***total graph*** $T(G)$ has $V \cup E$ as its vertex set, and its subgraphs induced by V and E are isomorphic to G and $L(G)$, respectively; moreover, $v \in V$ is adjacent to $e \in E$ in $T(G)$ if v is an endpoint of e in G.
- The ***square*** G^2 has the same vertex set as G; two vertices are adjacent in G^2 if they are at distance 1 or 2 apart in G.

D25: A ***χ-binding function*** on a class $\mathcal{G}$ of graphs is a function $f : \mathbb{N} \to \mathbb{N}$ such that $\chi(G) \leq f(\omega(G))$ for all $G \in \mathcal{G}$.

REMARKS

R3: Clearly,

$$\chi'(G) = \chi(L(G)) \quad \text{and} \quad ch'(G) = ch(L(G))$$

and similarly,

$$\chi''(G) = \chi(T(G)) \quad \text{and} \quad ch''(G) = ch(T(G))$$

In general, however, much stronger results are valid for χ' and ch' than for χ and ch, due to the restricted structure of line graphs.

R4: The cliques in the line graph $L(G)$ (and in the total graph $T(G)$) correspond to the stars (with their centers) and to the — possibly multiple-edged — triangles in the (multi)graph G. Thus,

$$\Delta(G) \leq \omega(L(G)) \leq \frac{3}{2}\Delta(G)$$

Moreover, $ch'(G) \geq \chi'(G) \geq \Delta(G)$.

R5: The total graph $T(G)$ of $G = (V, E)$ is isomorphic to the square of the bipartite graph $B = B(G)$ whose bipartition classes are V and E, where $v \in V$ and $e \in E$ are adjacent in B if v is an endpoint of e in G.

FACTS

F62: (Vizing's Theorem) [Vi64] If G is a simple graph, then

$$\chi'(G) \leq \Delta(G) + 1$$

If a multigraph G has maximum edge multiplicity $\mu(G)$, then

$$\chi'(G) \leq \Delta(G) + \mu(G)$$

For unrestricted edge multiplicity the upper bound $\chi'(G) \leq \frac{3}{2}\Delta(G)$ follows (first proved in [Sh49]). Thus, $f(\omega) = \omega + 1$ is a χ-binding function on the class of line graphs of simple graphs, and $f(\omega) = 3\omega/2$ is one on the class of all line graphs.

F63: [KiSc83] If the graph G contains no induced $K_{1,3}$ (*claw*) and no induced $K_5 - e$, then $\chi(G) \leq \omega(G) + 1$.

F64: (König's Theorem) [Kö16] If G is a bipartite multigraph, then $\chi'(G) = \Delta(G)$.

F65: [MoRe98] There exists a constant C such that $\chi''(G) \leq \Delta(G) + C$ holds for every graph G.

F66: [Ga95] If G is a bipartite multigraph, then $ch'(G) = \Delta(G)$.

F67: [ElGo93] Suppose that G is a d-regular multigraph with $\chi'(G) = d$. If G has an odd number of proper edge d-colorings, or if G is planar, then $ch'(G) = d$.

F68: For every multigraph G, $ch''(G) \leq ch'(G) + 2$.

F69: [Ka96] For every graph G of maximum degree Δ, $ch'(G) \leq (1 + o(1))\Delta$, where $o(1) \to 0$ as $\Delta \to \infty$.

CONJECTURES

C5: **Total Coloring Conjecture** (Vizing [Vi64]; Behzad) For every multigraph G with maximum edge multiplicity $\mu(G)$,

$$\chi''(G) \leq \Delta(G) + \mu(G) + 1$$

In particular, if G is a simple graph, then $\chi''(G) \leq \Delta(G) + 2$.

C6: **Overfull Subgraph Conjecture** (A. J. W. Hilton) If $\chi'(G) = \Delta(G) + 1$ and $\Delta(G) > |V(G)|/3$, then there is an "overfull" subgraph in G, i.e., $H \subseteq G$ with

$$|E(H)| > \Delta(H) \left\lfloor \frac{1}{2}|V(H)| \right\rfloor$$

C7: **List Coloring Conjecture** (Vizing; Gupta; Albertson and Collins; Bollobás and Harris) For every multigraph G, we have $ch'(G) = \chi'(G)$.

C8: (Gravier and Maffray) More generally than Conjecture 7, for every claw-free graph G we have $ch(G) = \chi(G)$.

C9: (Borodin, Kostochka and Woodall; Juvan, Mohar and Škrekovski; Hilton and Johnson) For every graph G, $ch''(G) = \chi''(G)$.

C10: (Kostochka and Woodall) More generally than Conjecture 4, $ch(G^2) = \chi(G^2)$ for the square of every graph G.

Snarks

DEFINITIONS

D26: A graph is cyclically k-connected if at least k edges must be deleted in order to leave two components, each containing a cycle.

D27: A ***snark*** (often called a “nontrivial snark”) is a 3-regular, cyclically 4-edge-connected graph of girth at least 5, that is *not* edge-3-colorable.

REMARK

R6: According to the two possible cases provided by Vizing’s theorem, a commonly used terminology says that a graph G is of ***Class 1*** or ***Class 2*** if its chromatic index is equal to $\Delta(G)$ or $\Delta(G)+1$, respectively. Hence, snarks represent the smallest nontrivial subfamily of Class 2. There has been much effort to construct snarks with various specific properties. Several methods of construction with further references can be found, e.g., in [Ko02] and in the earlier paper [CaMeRuSp98].

FACTS

F70: [Go85] If the Cycle Double Cover Conjecture (see §7.6.1) is false, then a smallest counterexample is a snark of girth at least 8.

F71: [Ce84] If the 5-flow conjecture (see §5.2.2) is false, then a smallest counterexample is a cyclically 5-edge-connected snark of girth at least 7.

F72: [Ko96] For every $g \geq 5$ there exists a cyclically 5-edge-connected snark of girth at least g.

Uniquely Edge-Colorable Graphs

DEFINITION

D28: A graph $G = (V, E)$ is ***uniquely edge k-colorable*** if it admits just one proper edge-coloring with $k = \chi'(G)$ colors (apart from the renumbering of colors).

EXAMPLES

E13: The uniquely edge 1-colorable graphs are the matchings. The uniquely edge 2-colorable graphs are the paths and the even cycles. The star graph $K_{1,3}$ and the complete graph K_4 are uniquely 3-edge-colorable. (No complete characterization is available for $k = 3$.)

E14: [Tu76] The graph consisting of two 9-cycles $a_1a_2\ldots a_9$ and $b_1b_2\ldots b_9$ and the further edges a_ib_{2i} ($1 \leq i \leq 9$, subscript addition modulo 9) is uniquely 3-edge-colorable. This is the only known triangle-free non-planar example for $k = 3$.

FACTS

F73: [Th78] The star graph $K_{1,k}$ is the only uniquely edge k-colorable graph, for each $k \geq 4$.

F74: If G is 3-regular and uniquely 3-colorable, then the following operation called “Y–Δ replacement” yields again a uniquely 3-colorable 3-regular graph. Let u_1, u_2, u_3

be the neighbors of vertex v. Remove v and insert the new vertices v_1, v_2, v_3 and the new edges v_1v_2, v_1v_3, v_2v_3 and u_iv_i for $i = 1, 2, 3$.

Further χ-Bound Graph Classes

CONJECTURES

C11: (Gyárfás; Sumner) For every tree T, there exists a χ-binding function on the class of graphs not containing T as an induced subgraph.

C12: [Gy87] There exists a χ-binding function for the class of graphs in which every induced cycle has length 3 or 4.

FACTS

F75: [KiPe94] A χ-binding function exists on the class of graphs not containing T, if the forbidden induced tree T has radius 2.

F76: [Sc97] For every tree T, there exists a χ-binding function on the class of graphs not containing any subdivision of T as an induced subgraph.

5.1.6 Coloring and Orientation

Paths and Cycles

FACTS

F77: [Ga68, Ro67] A graph has $\chi(G) \leq k$ if and only if G admits an orientation without directed paths on more than k vertices.

F78: [Mi62] A graph G has chromatic number $\chi(G) \leq k$ if and only if G admits an orientation such that every cycle $C \subseteq G$ has at least $|C|/k$ arcs oriented in each of the two directions around C.

F79: [Tu92] It suffices to assume Minty's condition above just for the cycles of length $|C| \equiv 1 \pmod{k}$. (This implies the Gallai–Roy Theorem, too.) Moveover, if such an orientation is given, then a proper k-coloring of G can be found in polynomial time. In particular, if the undirected graph G contains no cycles of length 1 modulo k, then G can be properly k-colored in linear time.

F80: [Bo76] In every strongly connected orientation of a k-chromatic graph ($k \geq 2$), there is a directed cycle of length at least k.

Eulerian Subgraphs

DEFINITION

D29: An ***Eulerian spanning subgraph*** of an oriented graph $\vec{G} = (V, \vec{E})$ is a subgraph $\vec{H} = (V, \vec{F})$ of $\vec{G}$ with the same vertex set V, and with $d^+_H(v) = d^-_H(v)$ for all $v \in V$. (Some or all vertices of $\vec{H}$ may be isolated; hence, $\vec{F} = \emptyset$ shows that every $\vec{G}$ has at least one such subgraph.)

FACT

F81: [AlTa92] If the number of Eulerian spanning subgraphs of an oriented graph $\vec{G}$ with an even number of edges differs from the number of those with an odd number of edges, then $\vec{G}$ is L-colorable whenever the list assignment L satisfies $|L_v| > d^+(v)$ for all $v \in V$. (The spanning subgraph condition holds in every bipartite graph.)

Choosability and Orientations with Kernels

FACTS

F82: Suppose that in the oriented graph $\vec{G} = (V, \vec{E})$ every induced subgraph $\vec{H}$ contains an *independent* set $Y \subseteq V$ such that from each vertex $v \notin Y$ of $\vec{H}$ there is at least one arc to Y. If L is a list assignment with $|L_v| > d^+(v)$ for all $v \in V$, then G admits a list coloring.

F83: The condition above holds in every bipartite directed graph. A more general class where the required subsets Y (termed *kernel*s) exist is the class of so-called *kernel-perfect graphs* (see §5.5). In those graphs, it suffices to consider orientations without directed 3-cycles.

F84: If every induced subgraph of a graph G has average degree at most $2k$, then G has an orientation with maximum out-degree at most k, and such an orientation can also be found in polynomial time, via the König–Hall theorem. In particular, if G is bipartite and satisfies the average-degree condition, then $ch(G) \leq k + 1$.

More results can be derived by applying this machinery on edge-colorings and planar graphs (see §5.1.5 and §5.2.2).

Acyclic Orientations

DEFINITION

D30: (Cf. [MaTo84] and [Ai88, p. 323]) The ***acyclic orientation game*** starts with an undirected graph. In each round, Player A ('Algy') selects a non-oriented edge e of G, and Player S ('Strategist') orients that edge in one direction, under the condition that no directed cycles may occur. The game is over when the graph G admits just one acyclic orientation that extends the partial orientation obtained so far. The goal of A is to finish the game in as few rounds as possible, while S aims at making the game long.

NOTATION: We denote by $c(G)$ the number of rounds when both A and S play optimally on graph G.

FACTS

F85: [AiTrTu95] If $G = (V, E)$ is a graph on $n \geq 6$ vertices with $c(G) = |E|$, then $|E| \leq \frac{1}{4}n^2$, and for $n \geq 7$ equality holds if and only if G is the complete bipartite graph $K_{\lfloor n/2 \rfloor, \lceil n/2 \rceil}$. (It is not known, however, whether $c(G) \leq \frac{1}{4}n^2 + o(n^2)$ for all graphs on n vertices as $n \to \infty$.)

F86: [AiTrTu95] For every g, there exists a graph $G = (V, E)$ with girth at least g and $c(G) < |E|$.

F87: [AlTu95] For the random graph $\mathbf{G}_{n,p}$ with n vertices and edge probability p, $c(\mathbf{G}_{n,p}) = \Theta(n \log n)$ with probability $1 - o(1)$ as $n \to \infty$ whenever $p > 0$ is fixed. For unrestricted p, a general upper bound is $c(\mathbf{G}_{n,p}) = O(n \log^3 n)$.

5.1.7 Colorings of Infinite Graphs

FACTS

F88: [BrEr51] For any $k \in \mathbb{N}$, an infinite graph G has $\chi(G) \leq k$ if and only if every finite subgraph of G is k-colorable.

F89: [Jo94] For any $k \in \mathbb{N}$, an infinite graph G has $ch(G) \leq k$ if and only if every finite subgraph of G is k-choosable.

F90: [Bo77] If $\chi(G) = \infty$, then for every infinite arithmetic progression $A \subseteq \mathbb{N}$, G contains a cycle whose length belongs to A.

F91: [GaKo91] That $\chi(G)$ is well-defined for every graph G, is equivalent to the set-theoretic Well-Ordering Theorem; and assuming that every set has a cardinality, it is equivalent to the Axiom of Choice.

F92: [Ko88] The following assertion is consistent: There exists a graph $G = (V, E)$ such that $|V| = \aleph_{\omega+1}$, $\chi(G) = \aleph_1$, and $\chi(H) \leq \aleph_0$ whenever $|V(H)| \leq \aleph_\omega$.

Coloring Euclidean Spaces

DEFINITIONS

D31: The ***unit distance graph*** $\mathcal{U}^n$ has the points of $\mathbb{R}^n$ as its vertices; the edges are the pairs of points whose Euclidean distance is 1.

D32: Given a (finite or infinite) "distance set" $D = \{d_1, d_2, \ldots\} \subset \mathbb{N}$, the ***distance graph*** $G(D)$ has vertex set $\mathbb{Z}$; two vertices $i, j \in \mathbb{Z}$ are adjacent if and only if $|i - j| \in D$.

FACTS

F93: [Ha45] In the plane, $\chi(\mathcal{U}^2) \leq 7$.

F94: [MoMo61] By a 4-chromatic subgraph on seven vertices, $\chi(\mathcal{U}^2) \geq 4$.

F95: [Co02] In 3-dimensional space, $\chi(\mathcal{U}^3) \leq 15$.

F96: A 5-chromatic subgraph on 9 vertices yields $\chi(\mathcal{U}^3) \geq 5$.

F97: [LaRo72] As $n \to \infty$, $\chi(\mathcal{U}^n)$ is at most $(3 + o(1))^n$.

F98: [FrWi81] As $n \to \infty$, $\chi(\mathcal{U}^n)$ is at least $(1 + o(1))\left(\frac{6}{5}\right)^n$.

F99: [Sc95] The choice number $ch(\mathcal{U}^n)$ is countably infinite if and only if $n = 2$ or $n = 3$. (Infinity was first observed in [JeTo95].)

F100: For every finite distance set D, $\chi(G(D)) \leq |D| + 1$.

F101: For every D with $|D| = 3$, $\chi(G(D))$ has been determined ([Zh02]). For $k > 3$, however, the chromatic number for k-element distance sets has not been characterized so far.

F102: [EgErSk85] If D is the set of all prime numbers, then $\chi(G(D)) = 4$. Moreover, $\chi(G(D \setminus \{3\})) = 3$ and $\chi(G(D \setminus \{2\})) = 2$.

F103: (Y. Katznelson; [RuTuVo02]; Y. Peres and W. Schlag) Let $\varepsilon_1 \geq \varepsilon_2 \geq \cdots > 0$ be a sequence of positive reals, and let $\mathcal{D}$ be the family of distance sets $D = \{d_1, d_2, \ldots\}$ such that $d_{i+1}/d_i \geq \varepsilon_i$ for all $i \geq 1$. If $\lim_{i\to\infty} \varepsilon_i > 0$, then $\chi(G(D))$ is finite for all those D; and if $\lim_{i\to\infty} \varepsilon_i = 0$, then there exists a $D \in \mathcal{D}$ such that $\chi(G(D)) = \infty$.

References

[Ai88] M. Aigner, *Combinatorial Search*, Wiley–Teubner, 1988.

[AiTrTu95] M. Aigner, E. Triesch, and Zs. Tuza, Searching for acyclic orientations of graphs, *Discrete Math.* 144 (1995), 3–10.

[AjKoSz80] M. Ajtai, J. Komlós, and E. Szemerédi, A note on Ramsey numbers, *J. Combin. Theory, Ser. A* 29 (1980), 354–360.

[Al93] N. Alon, Restricted colorings of graphs, pp. 1–33 in K. Walker (Ed.), *Surveys in Combinatorics*, Proc. 14th British Combinatorial Conference, London Math. Soc. Lecture Notes Series 187, Cambridge University Press, 1993.

[Al00] N. Alon, Degrees and choice numbers, *Random Struct. Alg.* 16 (2000), 364–368.

[AlKrSu99] N. Alon, M. Krivelevich, and B. Sudakov, List coloring of random and pseudo-random graphs, *Combinatorica* 19 (1999), 453–472.

[AlTa92] N. Alon and M. Tarsi, Colorings and orientations of graphs, *Combinatorica* 12 (1992), 125–134.

[AlTu95] N. Alon and Zs. Tuza, The acyclic orientation game on random graphs, *Random Struct. Alg.* 6 (1995), 261–268.

[Bá78] I. Bárány, A short proof of Kneser's conjecture, *J. Combin. Theory, Ser. A* 25 (1978), 325–326.

[Bo77] B. Bollobás, Cycles modulo k, *Bull. London Math. Soc.* 9 (1977), 97–98.

[Bo78] B. Bollobás, Uniquely colorable graphs, *J. Combin. Theory, Ser. B* 25 (1978), 54–61.

[Bo88] B. Bollobás, The chromatic number of random graphs, *Combinatorica* 8 1988, 49–55.

[BoCaEr80] B. Bollobás, P. A. Catlin, and P. Erdős, Hadwiger's conjecture is true for almost every graph, *Europ. J. Combin.* 1 (1980), 195–199.

[Bo76] J. A. Bondy, Diconnected orientations and a conjecture of Las Vergnas, *J. London Math. Soc.* (2) 14 (1976), 277–282.

[BoKo77] O. V. Borodin and A. V. Kostochka, On an upper bound of a graph's chromatic number, depending on the graph's degree and density, *J. Combin. Theory, Ser. B* 23 (1977), 247–250.

[Br02] S. Brandt, A 4-colour problem for dense triangle-free graphs, *Discrete Math.* 251 (2002), 33–46.

[Br41] R. L. Brooks, On colouring the nodes of a network, *Proc. Cambridge Philos. Soc.* 37 1941, 194–197.

[BrEr51] N. G. de Bruijn and P. Erdős, A colour problem for infinite graphs and a problem in the theory of relations, *Nederl. Akad. Wetensch. Proc. Ser. A* 54 (1951), 371–373.

[Ca78] P. A. Catlin, A bound on the chromatic number of a graph, *Discrete Math.* 22 (1978), 81–83.

[Ca79] P. A. Catlin, Hajós' graph-coloring conjecture: variations and counterexamples, *J. Combin. Theory, Ser. B* 26 (1979), 268–274.

[CaMeRuSp98] A. Cavicchioli, M. Meschiari, B. Ruini, and F. Spaggiari, A survey on snarks and new results: products, reducibility and a computer search, *J. Graph Theory* 28 (1998), 57–86.

[Ce84] U. A. Celmins, On cubic graphs that do not have an edge-3-colouring, PhD Thesis, University of Waterloo, Canada, 1984.

[ChGe69] G. Chartrand and D. P. Geller, On uniquely colorable planar graphs, *J. Combin. Theory* 6 (1969), 271–278.

[Co02] D. Coulson, A 15-colouring of 3-space omitting distance one, *Discrete Math.* 256 (2002), 83–90.

[De48] B. Descartes, Solutions to problems in Eureka No. 9, *Eureka* 10 (1948).

[Di52] G. A. Dirac, A property of 4-chromatic graphs and some remarks on critical graphs, *J. London Math. Soc.* 27 (1952), 85–92.

[Di57] G. A. Dirac, A theorem of R. L. Brooks and a conjecture of H. Hadwiger, *Proc. London Math. Soc. (3)* 7 (1957), 161–195.

[EgErSk85] R. B. Eggleton, P. Erdős, and D. K. Skilton, Coloring the real line, *J. Combin. Theory, Ser. B* 39 (1985), 86–100.

[ElGo96] M. N. Ellingham and L. Goddyn, List edge colourings of some 1-factorizable multigraphs, *Combinatorica* 16 (1996), 343–352.

[Er59] P. Erdős, Graph theory and probability, *Canad. J. Math.* 11 (1959), 34–38.

[Er67] P. Erdős, Some remarks on chromatic graphs, *Colloq. Math.* 16 (1967), 253–256.

[ErHa66] P. Erdős and A. Hajnal, On chromatic number of graphs and set-systems, *Acta Math. Acad. Sci. Hungar.* 17 (1966), 61–99.

[ErRuTa79] P. Erdős, A. L. Rubin, and H. Taylor, Choosability in graphs, Proc. West-Coast Conference on Combinatorics, Graph Theory and Computing, Arcata, California, *Congr. Numer.* XXVI (1979), 125–157.

[FrWi81] P. Frankl and R. M. Wilson, Intersection theorems with geometric consequences, *Combinatorica* 1 (1981), 357–368.

[Ga63] T. Gallai, Kritische Graphen I., *Publ. Math. Inst. Hungar. Acad. Sci.* 8 (1963), 165–192.

[Ga68] T. Gallai, On directed paths and circuits, pages 115–118 in P. Erdős and G. O. H. Katona (Eds.), *Theory of Graphs*, Colloq. Math. Soc. J. Bolyai, Tihany (Hungary), 1966, Academic Press, San Diego, 1968.

[Ga95] F. Galvin, The list chromatic index of a bipartite multigraph, *J. Combin. Theory, Ser. B* 63 (1995), 153–158.

[GaKo91] F. Galvin and P. Komjáth, Graph colorings and the axiom of choice, *Periodica Math. Hungar.* 22 (1991), 71–75.

[Go85] L. Goddyn, A girth requirement for the double cycle cover conjecture, pp. 13–26 in *Cycles in Graphs*, North-Holland Math. Stud. 115, 1985.

[Gr96] S. Gravier, A Hajós-like theorem for list coloring, *Discrete Math.* 152 (1996), 299–302.

[GrLo74] D. Greenwell and L. Lovász, Applications of product colouring, *Acta Math. Acad. Sci. Hungar.* 25 (1974), 335–340.

[Gy87] A. Gyárfás, Problems from the world surrounding perfect graphs, *Zastos. Mat.* XIX (1987), 413–441.

[Ha43] H. Hadwiger, Über eine Klassifikation der Streckenkomplexe, *Vierteljahrsch. Naturforsch. Ges. Zürich* 88 (1943), 133–142.

[Ha45] H. Hadwiger, Überdeckung des Euklidischen Raumes durch kongruente Mengen, *Portug. Math.* 4 (1945), 238–242.

[HaSz70] A. Hajnal and E. Szemerédi, Proof of a conjecture of Erdős, pp. 601–623 in P. Erdős, A. Rényi, and V. T. Sós (Eds.), *Combinatorial Theory and its Applications, Vol. II*, Colloq. Math. Soc. J. Bolyai 4, North-Holland, 1970.

[Ha61] Gy. Hajós, Über eine Konstruktion nicht n-färbbarer Graphen, *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 10 (1961), 116–117.

[JeTo95] T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley-Interscience, 1995.

[Jo96] A. Johansson, An improved upper bound on the choice number for triangle free graphs, Manuscript, January 1996.

[Jo96a] A. Johansson, The choice number of sparse graphs, Preliminary version, April 1996.

[Jo94] P. D. Johnson, The choice number of the plane, *Geombinatorics* 3 (1994), 122–128.

[Ka92] J. Kahn, Coloring nearly-disjoint hypergraphs with $n + o(n)$ colors, *J. Combin. Theory, Ser. A* 59 (1992), 31–39.

[Ka96] J. Kahn, Asymptotically good list-colorings, *J. Combin. Theory, Ser. A* 73 (1996), 1–59.

[KeKe54] J. B. Kelly and L. M. Kelly, Paths and circuits in critical graphs, *Amer. J. Math.* 76 (1954), 786–792.

[KiPe94] H. A. Kierstead and S. G. Penrice, Radius two trees specify χ-bounded classes, *J. Graph Theory* 18 (1994), 119–129.

[KiSc83] H. A. Kierstead and J. H. Schmerl, Some applications of Vizing's theorem to vertex colorings of graphs, *Discrete Math.* 45 (1983), 277–285.

[Ki95] J. H. Kim, On Brooks' theorem for sparse graphs, *Combin. Probab. Comput.* 4 (1995), 97–132.

[Ko96] M. Kochol, Snarks without small cycles, *J. Combin. Theory, Ser. B* 67 (1996), 34–47.

[Ko02] M. Kochol, Superposition and constructions of graphs without nowhere-zero k-flows, *Europ. J. Combin.* 23 (2002), 281–306.

[Ko88] P. Komjáth, Consistency results on infinite graphs, *Israel J. Math.* 61 (1988), 285–294.

[Kö16] D. König, Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre, *Math. Ann.* 77 (1916), 453–465.

[Ko78] A. V. Kostochka, Degree, girth and chromatic number, pages 679–696 in A. Hajnal and V. T. Sós (Eds.), *Combinatorics*, Colloq. Math. Soc. J. Bolyai 18, Keszthely (Hungary), 1976, North-Holland, 1978.

[Ko82] A. V. Kostochka, The minimum Hadwiger number for graphs with a given mean degree of vertices, *Metody Diskret. Analiz.* 38 (1982), 37–58. (in Russian)

[Ko84] A. V. Kostochka, On Hadwiger numbers of a graph and its complement, pages 537–545 in A. Hajnal, L. Lovász, and V. T. Sós (Eds.), *Finite and Infinite Sets*, Colloq. Math. Soc. J. Bolyai 37, Eger (Hungary) 1981, North-Holland, 1984.

[KoMa77] A. V. Kostochka and N. P. Mazurova, An inequality in the theory of graph coloring, *Metody Diskret. Analiz.* 30 (1977), 23–29. (in Russian)

[KrTuVo99] J. Kratochvíl, Zs. Tuza, and M. Voigt, New trends in the theory of graph colorings: Choosability and list coloring, pages 183–197 in R. L. Graham et al. (Eds.), *Contemporary Trends in Discrete Mathematics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 49, Amer. Math. Soc., 1999.

[Kr00] M. Krivelevich, The choice number of dense random graphs, *Combin. Probab. Comput.* 9 (2000), 19–26.

[LaRo72] D. G. Larman and C. A. Rogers, The realization of distances within sets in Euclidean space, *Mathematika* 19 (1972), 1–24.

[La78] J. Lawrence, Covering the vertex set of a graph with subgraphs of smaller degree, *Discrete Math.* 21 (1978), 61–68.

[Lo68] L. Lovász, On chromatic number of finite set-systems, *Acta Math. Acad. Sci. Hungar.* 19 (1968), 59–67.

[Lo73] L. Lovász, Independent sets in critical chromatic graphs, *Studia Sci. Math. Hungar.* 8 (1973), 165–168.

[Lo78] L. Lovász, Kneser's conjecture, chromatic number, and homotopy. *J. Combin. Theory, Ser. A* 25 (1978), 319–324.

[LuPhSa88] A. Lubotzky, R. Phillips, and P. Sarnak, Ramanujan graphs, *Combinatorica* 8 (1988), 261–277.

[MaTo84] U. Manber and M. Tompa, The effect of the number of Hamiltonian paths on the complexity of a vertex coloring problem, *Proc. 24th FOCS* (1984), 220–227.

[Mi92] P. Mihók, An extension of Brooks' theorem, *Annals of Discrete Math.* 51 (1992), 235–236.

[MiSc97] P. Mihók and I. Schiermeyer, Chromatic number of classes of graphs with prescribed cycle lengths, manuscript, 1997.

[Mi62] G. J. Minty, A theorem on n-colouring the points of a linear graph, *Amer. Math. Monthly* 67 (1962), 623–624.

[MoRe98] M. Molloy and B. Reed, A bound on the total chromatic number, *Combinatorica* 18 (1998), 241–280.

[MoMo61] L. Moser and W. Moser, Problem and solution P10, *Canad. J. Math.* 4 (1961), 187–189.

[My55] J. Mycielski, Sur le coloriage des graphes, *Colloq. Math.* 3 (1955), 161–162.

[NeRö79] J. Nešetřil and V. Rödl, A short proof of the existence of highly chromatic hypergraphs without short cycles, *J. Combin. Theory, Ser. B* 27 (1979), 225–227.

[NoGa56] E. A. Nordhaus and J. W. Gaddum, On complementary graphs, *Amer. Math. Monthly* 63 (1956), 175–177.

[Ra02] B. Randerath, 3-colorability and forbidden subgraphs. I: characterizing pairs, *Discrete Math.*, to appear.

[RaSc01] B. Randerath and I. Schiermeyer, Colouring graphs with prescribed induced cycle lengths, *Discuss. Math. Graph Theory* 21 (2001), 267–282.

[Re97] B. Reed, ω, Δ, and χ, *J. Graph Theory* 27 (1997), 177–212.

[RoSeTh93] N. Robertson, P. D. Seymour, and R. Thomas, Hadwiger's conjecture for K_6-free graphs, *Combinatorica* 13 (1993), 279–361.

[RöTu85] V. Rödl and Zs. Tuza, On color critical graphs, *J. Combin. Theory, Ser. B* 38 (1985), 204–213.

[Ro67] R. Roy, Nombre chromatique et plus longs chemins d'un graphe, *Revue AFIRO* 1 (1967), 127–132.

[RuTuVo02] I. Z. Ruzsa, Zs. Tuza, and M. Voigt, Distance graphs with finite chromatic number, *J. Combin. Theory, Ser. B* 85 (2002), 181–187.

[Sc95] J. H. Schmerl, The list-chromatic number of Euclidean space, *Geombinatorics* 5 (1995), 65–68.

[Sc78] A. Schrijver, Vertex-critical subgraphs of Kneser graphs, Nieuw Arch. Wisk. 26 (1978), 454–461.

[Sc97] A. D. Scott, Induced trees in graphs of large chromatic number, *J. Graph Theory* 24 (1997), 297–311.

[Se81] P. D. Seymour, Nowhere-zero 6-flows, *J. Combin. Theory, Ser. B* 30 (1981), 130–135.

[Sh49] C. E. Shannon, A theorem on coloring the lines of a network, *J. Math. Phys.* 28 (1949), 148–151.

[Si72] M. Simonovits, On colour-critical graphs, *Studia Sci. Math. Hungar.* 7 (1972), 67–81.

[St85] M. Stiebitz, Colour-critical graphs with complete major-vertex subgraph, pp. 169–181 in H. Sachs (Ed.), *Graphs, Hypergraphs and Applications* (Proc. Conf. Graph Theory, Eyba, GDR, 1984), Teubner-Texte zur Mathematik 73, Teubner, 1985.

[Su81] D. P. Sumner, Subtrees of a graph and the chromatic number, pages 557–576 in *The Theory and Applications of Graphs*, Proc. Conf. Kalamazoo (Michigan), Wiley, 1981.

[Th78] A. G. Thomason, Hamiltonian cycles and uniquely edge colourable graphs, *Annals of Discrete Math.* 3 (1978), 259–268.

[To70] B. Toft, On the maximal number of edges of critical k-chromatic graphs, *Studia Sci. Math. Hungar.* 5 (1970), 461–470.

[To72] B. Toft, Two theorems on critical 4-chromatic graphs, *Studia Sci. Math. Hungar.* 7 (1972), 83–89.

[To78] B. Toft, An investigation of colour-critical graphs with complements of low connectivity, *Annals of Discrete Math.* 3 (1978), 279–287.

[To85] B. Toft, Some problems and results related to subgraphs of colour critical graphs, pp. 178–186 in R. Bodendiek, H. Schumacher, and G. Walther (Eds.), *Graphen in Forschung und Unterricht: Festschrift K. Wagner*, Barbara Franzbecker Verlag, 1985.

[Tu54] W. T. Tutte, A contribution to the theory of chromatic polynomials, *Canad. J. Math.* 6 (1954), 80–91.

[Tu70] W. T. Tutte, More about chromatic polynomials and the golden ratio, pages 439–453 in R. K. Guy et al. (Eds.), *Combinatorial Structures and their Applications*, Gordon and Breach, 1970.

[Tu76] W. T. Tutte, Hamiltonian circuits, pp. 193–199 in *Colloquio Internazionale sulle Teorie Combinatorie* (Roma, 1973), Tomo I, Atti dei Convegni Lincei, No. 17, Accad. Naz. Lincei, Rome, 1976.

[Tu92] Zs. Tuza, Graph coloring in linear time, *J. Combin. Theory, Ser. B* 55 (1992), 236–243.

[Tu97] Zs. Tuza, Graph colorings with local constraints — A survey, *Discuss. Math. Graph Theory* 17 (1997), 161–228.

[Tu97a] Zs. Tuza, Choice-perfect graphs and Hall numbers, manuscript, 1997.

[TuVo94] Zs. Tuza and M. Voigt, in *Kolloquium Kombinatorik* (German Combinatorics Conference), Hamburg, 1994.

[TuVo97] Zs. Tuza and M. Voigt, List colorings and reducibility, *Discrete Applied Math.* 79 (1997), 247–256.

[Vi64] V. G. Vizing, On an estimate of the chromatic class of a p-graph, *Metody Diskret. Analiz.* 3 (1964), 9–17. (in Russian)

[Vi76] V. G. Vizing, Coloring the vertices of a graph in prescribed colors, *Metody Diskret. Anal. v Teorii Kodov i Schem*, 29 (1976), 3–10. (in Russian)

[Vu99] V. H. Vu, On some simple degree conditions that guarantee the upper bound on the chromatic (choice) number of random graphs. *J. Graph Theory* 31 (1999), 201–226.

[Wa37] K. Wagner, Über eine Eigenschaft der ebenen Komplexe, *Math. Ann.* 114 (1937), 570–590.

[Za98] W. Zang, Coloring graphs with no odd-K_4, *Discrete Math.* 184 (1998), 205–212.

[Zh02] X. Zhu, Circular chromatic number of distance graphs with distance sets of cardinality 3, *J. Graph Theory* 41 (2002), 195–207.

[Zy49] A. A. Zykov, On some problems of linear complexes, *Mat. Sbornik* 24 (1940), 163–188. (in Russian)

5.2 FURTHER TOPICS IN GRAPH COLORING

Zsolt Tuza, Hungarian Academy of Sciences
University of Veszprém, Hungary

5.2.1 Multicoloring and Fractional Coloring
5.2.2 Graphs on Surfaces
5.2.3 Some Further Types of Coloring Problems
5.2.4 Colorings of Hypergraphs
5.2.5 Algorithmic Complexity
References

Introduction

In this section we consider variants of graph coloring, and also the algorithmic complexity of the problems. Some of the concepts here (e.g., face coloring of planar graphs) may be viewed as equivalents of proper vertex-coloring on restricted classes of graphs, a perspective for which there is an extensive literature. Some other topics here are generalizations in various directions.

The current section applies several concepts introduced in §5.1. Some familiarity with §5.1 is assumed. (Please see the chapter glossary as needed.)

5.2.1 Multicoloring and Fractional Coloring

DEFINITIONS

D1: A ***fractional vertex-coloring*** of G is a real function $\varphi^* : \mathcal{S} \to \mathbb{R}^{\geq 0}$ on the collection $\mathcal{S}$ of all independent vertex sets in $G = (V, E)$ such that for all $v \in V$

$$\sum_{\{S \in \mathcal{S} | v \in S\}} \varphi^*(S) \geq 1$$

This definition extends naturally to other types of coloring (e.g., fractional edge-coloring), leading in this way to the fractional versions of further graph invariants.

D2: The ***fractional chromatic number*** of G is

$$\chi^*(G) := \min_{\varphi^*} \sum_{S \in \mathcal{S}} \varphi^*(S)$$

where the minimum is taken over all fractional vertex-colorings φ^* of G. (The *fractional chromatic index* $\chi'^*(G)$ is defined analogously.) Equivalently, the fractional chromatic number of a graph $G = (V, E)$ is definable as the minimum ratio p/q such that there exists a cover of V by p independent sets $S_1, S_2, \ldots, S_p$ (not necessarily distinct), with each $v \in V$ contained in precisely q of them. It is denoted by $\chi^*(G)$.

D3: For two functions f and g from V to $\mathbb{N}$, with $g(v) \leq f(v)$ for all $v \in V$, the graph $G = (V, E)$ is (f, g)***-choosable*** if for every list assignment L with $|L_v| = f(v)$ there can be chosen subsets $C_v \subseteq L_v$ such that $|C_v| = g(v)$ for all $v \in V$ and $C_u \cap C_v = \emptyset$ for all $uv \in E$. The functions f and/or g may be constant, e.g., the terminology "(a,b)-choosable" means that $f(v) = a$ and $g(v) = b$ for all $v \in V$. If $g(v) = 1$ for every vertex v, we simply say that G is ***f*-choosable**.

EXAMPLE

E1: For every $t \in \mathbb{N}$, the odd cycle C_{2t+1} is $(2t+1, t)$-choosable ([AlTuVo97]) and its fractional chromatic number is $2 + \frac{1}{t}$.

FACTS

F1: For every graph G, we have

$$\omega(G) \leq \chi^*(G) \leq \chi(G)$$

In particular, if G is a perfect graph, then $\chi^*(G) = \chi(G)$.

F2: For every graph $G = (V, E)$, we have $\chi^*(G) \geq |V|/\alpha(G)$.

F3: For every graph G on n vertices, the value $\chi^*(G) = p/q$ is attained for some $q \leq n^{n/2}$ ([ChGaJo78]), but there exists a constant C ($C > 1.34619$) and infinitely many graphs G for which $q > C^n$ is necessary ([Fi95]).

F4: [ReSe98] For every natural number t, if $\chi^*(G) > 2t$, then some subgraph of G is contractible to K_{t+1}. (Cf. Hadwiger's conjecture in §5.1.4.)

F5: [Lo75] For every graph G,

$$\chi(G) \leq (1 + \log \alpha(G)) \max_{H \subseteq G} \frac{|V(H)|}{\alpha(H)}$$

As a consequence, χ/χ^* is bounded above by $1 + \log n$ (where n is the number of vertices).

F6: [Ry90] For fractional colorings of simple graphs, the Total Coloring Conjecture (cf. §5.1.5) holds true, i.e., $\chi''^*(G) \leq \Delta(G) + 2$ for every graph G.

F7: [Ka00] The edge choice number asymptotically equals the fractional chromatic index, i.e., for every $\varepsilon > 0$ there exists a $k = k(\varepsilon)$ such that $\chi'^*(G) \geq k$ implies $ch'(G) \leq (1+\varepsilon)\chi'^*(G)$, for every multigraph G.

F8: [AlTuVo97] For every graph G, the minimum value of a/b such that G is (a,b)-choosable is equal to $\chi^*(G)$.

F9: (Brooks's Theorem for unequal lists) [ErRuTa79] Suppose that G is connected, and that at least one of its blocks is neither a complete graph nor an odd cycle. If $f(v)$ is the degree of v for every vertex v, then G is f-choosable.

F10: *For multicoloring:* [TuVo96] Under the same conditions as Fact 9, G is also (mf, m)-choosable for all $m \in \mathbb{N}$.

F11: The List Reduction method (see [TuVo97] in §5.1.1) can be applied for (km, m)-choosability, too.

F12: [TuVo96a] Every 2-choosable graph is $(2m, m)$-choosable, for every $m \in \mathbb{N}$.

OPEN PROBLEMS

P1: [ErRuTa79] Is every (a,b)-choosable graph (am,bm)-choosable for all $m \in \mathbb{N}$?

P2: [ErRuTa79] Given any pair of graphs G and H on the same set of vertices, is $ch(G \cup H) \leq ch(G)ch(H)$?

REMARK

R1: An affirmative answer to Problem 1 would imply an affirmative answer on Problem 2 also. Various particular cases of the former are proved in [TuVo96].

5.2.2 Graphs on Surfaces

Historically, for more than a half century, the theory of graph coloring dealt with face colorings of maps, which can equivalently be interpreted as vertex-colorings of their dual graphs. Via duality, there is a natural correspondence between total colorings of graphs and simultaneously coloring the edges and faces of maps on surfaces. Adjacency of faces of a map means sharing an edge on their boundary; and incidence of a vertex or edge with a face means belonging to its boundary walk.

DEFINITIONS

D4: A ***plane graph*** is a planar graph together with a given imbedding in the plane.

D5: A ***triangulation*** (in the plane or on a higher surface) is a graph imbedding in a surface such that all the face boundaries are cycles of length 3.

D6: A graph is ***outerplanar*** if it has an imbedding in the plane such that all vertices lie on the boundary walk of the exterior face.

FACTS

F13: (Five Color Theorem) [He:1890] Every planar graph is 5-colorable.

F14: (Four Color Theorem) [ApHa77, ApHaKo77] Every planar graph is 4-colorable.

F15: [Ta:1880] A plane graph G is 4-colorable if and only if its dual G^* is 3-edge-colorable.

F16: [Bo79] Every planar graph has a proper 5-coloring such that the union of any two color classes induces a forest.

F17: (Grötzsch's Theorem) [Gr59] Every K_3-free planar graph is 3-colorable.

F18: [He:1898] A planar triangulation is 3-colorable if and only if all of its vertices have even degrees.

F19: Every planar graph is 5-choosable ([Th94]), but there exist non-4-choosable planar graphs ([Vo93]).

F20: [KrTu94] Every K_3-free planar graph is 4-choosable.

F21: [Th95] All planar graphs of girth at least five are 3-choosable.

F22: [AlTa92] All bipartite planar graphs are 3-choosable.

F23: [Vo95] There exist non-3-choosable K_3-free planar graphs.

F24: [ChGeHe71] Every planar graph has a vertex partition into two classes where each class induces an outerplanar graph.

F25: [He93] The edge set of any planar graph can be partitioned into two outerplanar graphs.

REMARK

R2: A simpler proof to the Four Color Theorem is given in [RoSaSeTh97]; but so far no proof without the extensive use of a computer is known.

Heawood Number and the Empire Problem

DEFINITION

D7: The ***Heawood number*** of a closed surface S of Euler characteristic ϵ is

$$H(\epsilon) = \left\lfloor \frac{7 + \sqrt{49 - 24\epsilon}}{2} \right\rfloor$$

More generally, for every natural number m we write

$$H(\epsilon, m) = \left\lfloor \frac{6m + 1 + \sqrt{(6m+1)^2 - 24\epsilon}}{2} \right\rfloor$$

FACTS

F26: [He:1890] If a surface S has Euler characteristic $\epsilon < 2$, then the connected regions of any map drawn on S — or equivalently, any graph imbedded in S — can be properly colored with at most $H(\epsilon)$ colors.

F27: [Fr34] Every graph drawn on the Klein bottle is 6-colorable.

F28: [RiYo68] On any other surface S except the Klein bottle, the maximum chromatic number of graphs imbeddable in S is equal to $H(\epsilon)$.

F29: [Di52] $(-1 \neq \epsilon \neq 1)$, [AlHu79] $(\epsilon = -1$ or $1)$ If S is a surface of Euler characteristic $\epsilon < 2$, other than the Klein bottle, then every $H(\epsilon)$-chromatic graph imbedded in S contains $K_{H(\epsilon)}$ as a subgraph.

F30: [BöMoSt99] With the possible exception of $\epsilon = -1$, every graph G imbedded in a surface S of Euler characteristic ϵ with $ch(G) = H(\epsilon)$ contains the complete graph $K_{H(\epsilon)}$.

F31: [He:1890] If each country on a surface of Euler characteristic ϵ consists of at most m connected regions $(m \geq 2)$, then the countries can be colored with at most $H(\epsilon, m)$ colors so that any two neighbor countries are colored differently. (For planar maps this means $6m$ colors.)

F32: [JaRi84] For every $m \geq 2$, there exist planar maps with countries of m regions each, where $6m$ colors are necessary.

F33: $H(\epsilon, m)$ colors are necessary for every $m \geq 2$ on the torus (H. Taylor in [Ga80]), on the projective plane ([JaRi83]) and on the Klein bottle ([JaRi85] for $m \geq 3$ and [Bo89] for $m = 2$).

OPEN PROBLEM

P3: (Empire Problem) For which surfaces S and for which values of m do there exist maps on S with at most m connected regions in each country (here called an *empire* or an *m-pire*), such that $H(\epsilon, m)$ colors are necessary for a proper coloring of all countries?

Nowhere-Zero Flows

DEFINITION

D8: Let $\vec{G} = (V, \vec{E})$ be an oriented multigraph and $k \geq 2$ an integer. A ***nowhere-zero k-flow*** is a function $\phi : E \to \{1, 2, \ldots, k-1\}$ such that

$$\sum_{uv \in \vec{E}} \phi(uv) = \sum_{vw \in \vec{E}} \phi(vw)$$

holds for every vertex $v \in V$.

FACTS

F34: [Tu54] A plane graph G is k-colorable if and only if its planar dual G^* admits an orientation with a nowhere-zero k-flow. (The analogous property holds for a graph imbedded in any orientable surface.)

In particular, the Four Color Theorem is equivalent to the assertion that every planar graph without cut-edges has a nowhere-zero 4-flow; and Grötzsch's theorem asserts (in dual form) that every 4-edge-connected planar graph has a nowhere-zero 3-flow.

F35: [Tu50, Mi67] A 3-regular multigraph is bipartite if and only if it has a nowhere-zero 3-flow; and it is 3-edge-colorable if and only if it has a nowhere-zero 4-flow. In particular, the former assertion generalizes the fact that the *skeleton* (i.e., the graph) of every Eulerian planar triangulation is 3-colorable.

F36: [Se81] Every graph without cut-edges has a nowhere-zero 6-flow.

CONJECTURES

C1: **5-flow conjecture** [Tu54] Every graph without cut-edges has a nowhere-zero 5-flow.

C2: [Tu54] Every 4-edge-connected graph has a nowhere-zero 3-flow.

Chromatic Polynomials

DEFINITION

D9: The ***chromatic polynomial*** $P(G, \lambda)$, $\lambda \in \mathbb{N}$, of graph $G = (V, E)$ is the function whose value at λ ($\lambda = 1, 2, 3, \ldots$) is the number of proper colorings $\varphi : V \to \{1, \ldots, \lambda\}$ of G with at most λ colors. Here, two colorings are counted as different even if they yield the same color classes by renumbering the colors.

EXAMPLE

E2: The chromatic polynomials of the edgeless graph and the complete graph on n vertices are, respectively,

$$P(\overline{K}_n, \lambda) = \lambda^n \quad \text{and} \quad P(K_n, \lambda) = \binom{\lambda}{n} n! = \lambda(\lambda - 1) \cdots (\lambda - n + 1)$$

FACTS

F37: (Deletion-Contraction Formula) For every graph $G = (V, E)$ and every edge $e \in E$, we have $P(G, \lambda) = P(G - e, \lambda) - P(G/e, \lambda)$, where '$-e$' and '$/e$' mean the deletion and contraction of edge e, respectively.

F38: [Bi12] If the graph G has n vertices, then $P(G, \lambda)$ is a polynomial of degree n in λ, with integer coefficients, and $\chi(G)$ is the smallest natural number λ such that $P(G, \lambda) \neq 0$.

F39: (Golden Identity) [Tu70] If G is a planar triangulation on n vertices, then

$$P(G, \tau + 2) = \tau^{3n-10} P^2(G, \tau + 1)$$

where $\tau = \dfrac{1 + \sqrt{5}}{2}$ denotes the golden ratio.

REMARK

R3: Although one can deduce from the Golden Identity that $P(G, \tau + 2) > 0$ holds for every planar triangulation ($\tau + 2 = 3.618...$), this does not seem to lead closer to a computer-free proof of the Four Color Theorem. (As a matter of fact, $P(G, \tau + 1)$ is nonzero for every connected graph G.)

5.2.3 Some Further Types of Coloring Problems

Variants of Proper Coloring

We briefly mention some further coloring concepts, with only a few references.

DEFINITIONS

D10: A ***Grundy coloring*** of a graph is a proper vertex-coloring $\varphi : V \to \mathbb{N}$ such that every vertex v has a neighbor of color i for all $1 \leq i < \varphi(v)$.

D11: The ***Grundy number*** of a graph is the largest number of colors in a Grundy coloring.

D12: An ***achromatic coloring*** of a graph is a proper vertex-coloring such that each pair of color classes is adjacent by at least one edge.

D13: The largest possible number of colors in an achromatic coloring is called the ***achromatic number***.

D14: A **b-*coloring*** of a graph is a proper vertex-coloring such that each color class contains a vertex adjacent to some vertices in every other color class.

D15: The **b-*chromatic number*** of a graph is the largest number of colors in a b-coloring.

D16: A ***λ-coloring*** (also called a ***radio coloring*** or an ***$L(2,1)$-labeling***) is a vertex-coloring $\varphi : V \to \{0, 1, \ldots, k\}$ (i.e., $k+1$ colors may be used) such that if $uv \in E$ then $|\varphi(u) - \varphi(v)| \geq 2$, and if vertices u and v have a common neighbor then $\varphi(u) \neq \varphi(v)$.

D17: Given a set $T \subseteq \{0, 1, 2, \ldots\}$, a ***T-coloring*** is a mapping $\varphi : V \to \mathbb{N}$ such that $|\varphi(u) - \varphi(v)| \notin T$ for all edges $uv \in E$.

D18: A ***circular coloring*** is an assignment of a real number $\varphi(v)$ (with $0 \leq \varphi(v) < C$) to each vertex v, in such a way that if uv is an edge then $(\varphi(v) - \varphi(u)$ modulo $C) \geq 1$.

D19: An ***harmonious coloring*** of a graph is a partition of the vertex set into independent sets such that the union of any two induces at most one edge.

FACTS

F40: The smallest possible number of colors in a Grundy coloring of a graph G is just $\chi(G)$. On the other hand, the algorithmic complexity of finding the Grundy number – the largest number of colors – has not been determined so far. See, e.g., [ChSe79], [HeHeBe82]; and see also [Si83] with a different terminology.

F41: It is NP-hard to determine the achromatic number, even for trees ([CaEd97]), but polynomial-time solvable for trees of bounded degree ([CaEd98]). It is also hard to *approximate* the achromatic number on a general input graph within a factor $2 - \epsilon$, for any $\epsilon > 0$ ([KoKr01]).

REMARKS

R4: We refer to [IrMa99] and [KrTuVo02] for various estimates and complexity results regarding b-colorings (e.g., hardness on connected bipartite graphs).

R5: One challenging open problem on λ-colorings asks whether $k = \Delta^2$ is sufficient (i.e., whether $\Delta^2 + 1$ colors are enough) for every graph with maximum degree Δ.

R6: For results and further references on λ-colorings, see [GrYe92], [ChKu96], and [BoKlTaLe00].

R7: In the list coloring version of T-colorings, also the case $0 \notin T$ leads to interesting questions. See, e.g., [Te93], [Wa96], and [ChLiZh99].

R8: The goal of studying circular colorings is to minimize the value of the modulus C under which a circular coloring exists. We refer to the survey [Zh01] on undirected graphs, and to the recent paper [BoFiJuKaMo02] on the digraph version of the concept.

R9: The goal of harmonious coloring investigations is to minimize the number of vertex classes. See [Ed97] for a survey.

Graph Homomorphisms

DEFINITION

D20: A ***homomorphism*** from a graph $G = (V, E)$ to a graph $H = (X, F)$ is a vertex mapping $\varphi : V \to X$ such that

$$\varphi(u)\varphi(v) \in F \quad \text{for all } uv \in E$$

A homomorphism into the complete graph K_k can be viewed as a proper k-coloring and vice versa. Moreover, a homomorphism into H is also called an ***H-coloring of** G*.

NOTATION: The notation $G \to H$ means that G has at least one H-coloring, and $G \not\to H$ denotes that G is not H-colorable. The concept is extended to digraphs in a natural way (i.e., where uv and $\varphi(u)\varphi(v)$ are *ordered* pairs).

FACT

F42: [NeTa99] Let G_1 and G_2 be graphs such that $\chi(G_2) \geq 3$ and $G_1 \to G_2$ but $G_2 \not\to G_1$. Then there exists a graph G with $G_1 \to G \to G_2$ and $G_2 \not\to G \not\to G_1$.

For further results on H-coloring, see §5.2.5.

Coloring with Costs

DEFINITIONS

D21: A ***cost set*** $C = \{c_1, c_2, \ldots\}$ associates a cost $c_i > 0$ with each color i. It is assumed without loss of generality that $0 < c_1 < c_2 < \cdots$, and also that $|C| \geq \chi(G)$.

D22: Given a graph $G = (V, E)$ and a cost set C, the ***cost of a coloring*** $\varphi : V \to \mathbb{N}$ is the sum $\sum_{v \in V} c_{\varphi(v)}$. We denote by $\Sigma_C(G)$ the smallest possible cost of a proper vertex-coloring φ of G. If $C = \mathbb{N}$, this notion simplifies to $\Sigma(G) := \min_{\varphi} \sum_{v \in V} \varphi(v)$ and usually is called the ***chromatic sum*** or ***color cost*** of G.

D23: The ***cost chromatic number*** of G with respect to a cost set C is the smallest possible number of colors in a minimum-cost coloring. If $C = \mathbb{N}$, this parameter is often called the ***strength*** of G.

FACTS

F43: [ThErAlMaSc89] If G is connected and has m edges, then

$$\left\lceil \sqrt{8m} \right\rceil \;\leq\; \Sigma(G) \;\leq\; \left\lfloor \frac{3}{2}(m+1) \right\rfloor$$

F44: [MiMo97] For every finite cost set C there exists a tree T whose cost chromatic number is equal to $|C|$.

F45: [Tu90] For every $s \geq 2$, the minimum number of vertices in a tree of strength s equals $((2+\sqrt{2})^{s-1} - (2-\sqrt{2})^{s-1})/\sqrt{2}$. Moreover, for every $s \geq 3$, there exist precisely two trees of strength s, which are minimal in the sense that every tree of strength at least s is contractible to at least one of them.

F46: [MiMo97] Every tree of maximum degree Δ has cost chromatic number at most $\lceil \Delta/2 \rceil + 1$.

F47: The bound in Fact 46 is tight for every Δ with the cost set $C = \{1, 1.1, 1.11, \ldots\}$ ([MiMoSc97]) and also with $C = \mathbb{N}$ ([JiWe99]).

REMARK

R10: Beside the chromatic sum, various notions concerning coloring with costs have been motivated by scheduling problems. In some of them, given numbers of colors have to be assigned to the vertices. For results of this type, see, e.g., [Ma02].

Vertex Ranking

DEFINITION

D24: A ***vertex ranking*** of graph $G = (V, E)$ is a (necessarily proper) coloring $\varphi : V \to \mathbb{N}$ with the property that for any two vertices u, v of the same color, every u–v path contains some vertex z with $\varphi(z) > \varphi(u)$. The smallest possible number of colors, called ***ranking number***, will be denoted by $\chi_r(G)$. In the directed analogue for digraphs, the requirement is put on directed paths only.

EXAMPLE

E3: [BoDeJaKlKrMüTu98] The line graph of K_n $(n \geq 2)$ has ranking number $\frac{1}{3}(n^2 + g(n))$, where the function $g(n)$ is defined recursively with $g(1) = -1$, $g(2k) = g(k)$, and $g(2k+1) = g(k+1) + k$ for every natural number k. (No closed formula is available.)

FACTS

F48: For every graph G, $col(G) \leq \chi_r(G)$; and if the ranking number is at most k, then the graph is (km, m)-choosable for every $m \in \mathbb{N}$ ([TuVo96]).

F49: [BoDeJaKlKrMüTu98] If $\chi_r(G) = \chi(G)$, then also $\omega(G) = \chi(G)$. Moreover, $\chi_r(H) = \omega(H)$ holds for every induced subgraph H of G if and only if G contains no P_4 and C_4 as an induced subgraph.

Partial Colorings and Extensions

Also here, we mention some concepts with only a few references.

DEFINITIONS

D25: A ***partial coloring*** of graph $G = (V, E)$ on a vertex subset $W \subseteq V$ is a coloring $\varphi_W : W \to \mathbb{N}$.

D26: In the ***Precoloring Extension*** problem, abbreviated ***PrExt***, we are given a graph G, a ***color bound*** k, and a proper partial coloring φ_W. The question is whether φ_W can be extended to a proper k-coloring of the entire G.

D27: For a nonnegative integer t, the problem ***t-PrExt*** is the restricted version of Precoloring Extension where the given partial coloring uses each color at most t times. (Hence, 0-PrExt is an equivalent formulation of asking whether the graph in question is k-colorable.)

D28: In an ***on-line coloring*** the vertices $v_1, v_2, \ldots, v_n$ of graph G are received one by one in some unknown order. When v_i appears, we also get the information which its neighbors in $\{v_1, \ldots, v_{i-1}\}$ are. A color has to be assigned to v_i without any information on its adjacencies to the v_j, $i < j \leq n$.

FACTS

F50: If $G = (V, E)$ is k-colorable, and $t \leq k$ is a positive integer, then G has a proper partial coloring on at least $\frac{t}{k}\,|V|$ of its vertices.

F51: [AlGrHa00] If G is a k-colorable graph with n vertices, then for every list t-assignment it has a partial proper list coloring on at least $\left(1 - \left(\frac{k-1}{k}\right)^t\right) n$ vertices.

F52: If G is a k-choosable graph with n vertices, then it has a partial list coloring on more than $\frac{6}{7}\frac{t}{k}n$ vertices, for every list t-assignment $(1 \le t < k)$ [Ch99]. Moreover, if G has maximum degree k or the union of lists contains at most k colors, then at least tn/k vertices can be colored from their lists [Ja01]. (It is conjectured in [AlGrHa00] that this lower bound is valid for all G and all t-assignments.)

F53: (Cf. §5.1.5) If T is a tree of radius 2, then there exists an on-line χ-binding function on the class of graphs not containing T as an induced subgraph ([KiPeTr94]); but the class of induced-P_6-free graphs is not on-line χ-bound ([GyLe91]).

REMARKS

R11: In several graph classes, efficiently testable necessary and sufficient conditions can be given for the extendability of partial colorings. Details can be found in [HuTu96].

R12: Results on on-line coloring are surveyed in [Ki98a].

Partitions with Weaker Requirements

REMARKS

R13: There are many papers dealing with vertex- or edge-partitions into parts that are not necessarily independent, but satisfy some weaker properties. Usually it is assumed that the property to be satisfied in each part is *hereditary* or *induced-hereditary*, i.e., if it holds for a graph H then it also holds for all (induced) subgraphs of H. A detailed discussion on the general theory can be found in the survey [BoBrFrMiSe97].

R14: Sometimes conditions are imposed on the vertex degrees in each partition class. For results and references on this kind of problem, see, e.g., [Wo01].

5.2.4 Colorings of Hypergraphs

Beside some results on the coloring of finite set systems (hypergraphs), here we also mention the basic definitions and a few facts from the recently fast-developing theory of "mixed hypergraph coloring". For a detailed account, see the informative monograph [Vo02].

DEFINITIONS

D29: A ***hypergraph*** $\mathcal{H} = (X, \mathcal{F})$ has vertex set X; its edge set $\mathcal{F}$ consists of subsets of X. We assume that $\mathcal{F} \neq \emptyset$ and that $|F| \ge 2$, for all $F \in \mathcal{F}$.

D30: A ***proper vertex k-coloring*** of a hypergraph $\mathcal{H}$ is a mapping $\varphi : X \to \{1, 2, \ldots, k\}$ such that no edge of $\mathcal{H}$ is monochromatic. Equivalently, it is a vertex partition into k classes such that no color class contains any edge.

D31: A ***proper edge-coloring*** of a hypergraph $\mathcal{H}$ is an edge partition such that the edges in the same class are mutually vertex-disjoint.

D32: The ***chromatic number*** $\chi(\mathcal{H})$ and ***chromatic index*** $\chi'(\mathcal{H})$ are the smallest numbers of colors in a proper vertex and proper edge-coloring, respectively. ***List coloring***, ***choice number***, and ***choice index*** can be defined for hypergraphs analogously.

D33: A hypergraph is ***r*-uniform** if every edge has precisely r vertices.

D34: The **complete *r*-uniform hypergraph** of order n $(n \geq r)$, denoted $\mathcal{K}_n^r$, has $|X| = n$, and its edge set consists of all the r-element subsets of X.

D35: A ***mixed hypergraph*** $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ has vertex set X, and two types of edges: the ***C-edges*** in $\mathcal{C}$ and the ***D-edges*** in $\mathcal{D}$, respectively. It is called a ***bi-hypergraph*** if $\mathcal{C} = \mathcal{D}$, a ***C-hypergraph*** if $\mathcal{D} = \emptyset$, and a ***D-hypergraph*** if $\mathcal{C} = \emptyset$. We shall assume that at least one of $\mathcal{C}$ and $\mathcal{D}$ is nonempty, and also that every (C- and D-) edge has at least two vertices.

D36: A ***strict k-coloring*** of a mixed hypergraph is a vertex-coloring with exactly k colors, such that every C-edge has two vertices with a common color and every D-edge has two vertices with different colors. (In this way, the D-hypergraphs are just the hypergraphs in the usual sense.)

D37: A mixed hypergraph is said to be ***colorable*** if it admits at least one strict coloring, and ***uncolorable*** if it doesn't.

D38: If a mixed hypergraph $\mathcal{H}$ is colorable, then the smallest and largest number of colors in a strict coloring is called the ***lower*** and ***upper chromatic number***, denoted $\chi(\mathcal{H})$ and $\overline{\chi}(\mathcal{H})$, respectively.

D39: A mixed hypergraph $\mathcal{H}$ is said to be ***uniquely colorable*** if $\chi(\mathcal{H}) = \overline{\chi}(\mathcal{H})$ and $\mathcal{H}$ has only one strict coloring (apart from renaming the colors).

EXAMPLES

E4: If $\mathcal{H}$ is a C-hypergraph, then $\chi(\mathcal{H}) = 1$, since the entire vertex set may be colored with the same color; and if $\mathcal{H}$ is a D-hypergraph, then $\overline{\chi}(\mathcal{H}) = |X|$, because its vertices may get mutually distinct colors.

E5: The complete hypergraph (viewed as a D-hypergraph) has $\chi(\mathcal{K}_n^r) = \lceil \frac{n}{r-1} \rceil$; and when viewed as a C-hypergraph, it has $\overline{\chi}(\mathcal{K}_n^r) = r - 1$. If $\mathcal{C} = \mathcal{K}_n^p$, $\mathcal{D} = \mathcal{K}_n^q$, and $n > (p-1)(q-1)$, then $\mathcal{H}$ is uncolorable.

E6: [ErLo75] Let $X = X_1 \cup \cdots \cup X_r$, with $|X_i| = i$ for all $1 \leq i \leq r$, and let an r-element set be an edge if and only if for some i it contains X_i and has precisely one vertex in every X_j with $i < j \leq r$. This r-uniform hypergraph is not 2-colorable, for all $r \geq 2$.

E7: [TuVo00] The following mixed hypergraphs $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ are uncolorable: starting from a k-chromatic graph $G = (V, E)$, set $X = V$, $\mathcal{D} = E$, and let a k-subset $Y \subset V$ be a C-edge if the subgraph of G induced by Y has a hamiltonian path.

FACTS

F54: [Er64] For every $r \geq 2$,there exists a non-2-colorable r-uniform hypergraph with fewer than $r^2\, 2^{r+1}$ edges.

F55: [RaSr00] For sufficiently large values of r, every r-uniform hypergraph with at most $0.7\sqrt{n/\ln n}\, 2^n$ edges is 2-colorable, and efficient algorithms can also be designed to find a proper 2-coloring. (The previous bound $n^{\frac{1}{3}-\varepsilon}\, 2^r$ is given in [Be78].)

F56: (See [Lo68, NeRö79] in §5.1.4.) For every triple of integers $r, k, g \geq 3$, there exists an r-uniform hypergraph with chromatic number at least k and girth at least g.

F57: [KrVu01] For every positive integer r and for every real-valued function $p = p(n)$ with

$$n^{-(r-1)^2/(2r)+\epsilon} \leq p \leq 0.9 \quad (\epsilon > 0 \text{ fixed})$$

the random r-uniform hypergraph with n vertices and edge probability $p = p(n)$ has

$$ch(\mathbf{H}_{n,p}) = (1 + o(1))\chi(\mathbf{H}_{n,p}) \quad \text{with probability } 1 - o(1)$$

Also, if $p \geq Cn^{1-r}$ for a sufficiently large constant C, then

$$ch(\mathbf{H}_{n,p})/\chi(\mathbf{H}_{n,p}) = (1 + \delta_p)\, r^{1/(r-1)} \quad \text{almost surely}$$

where $\delta_p \to 0$ as $n^{r-1}p \to \infty$.

F58: [KoSt00] If a hypergraph on n vertices is $(k + 1)$-chromatic, critical, and does not contain any 2-element edges, then the number of its edges is at least $(k - 3/\sqrt[3]{k})\, n$.

F59: [TuVo00] The largest possible value of $\overline{\chi}(\mathcal{C}) - \chi(\mathcal{D})$ in *uncolorable* mixed hyperaphs $\mathcal{H} = (X, \mathcal{C}, \mathcal{D})$ with $|X| = n$ vertices is equal to $n - 4$.

F60: [TuVo00] The List Coloring problem can be reduced to the colorability problem of mixed hypergraphs, in the following sense. For every graph $G = (V, E)$ and for every list assignment L on G, there can be constructed (in linear time) a mixed hypergraph $\mathcal{H}$ on $|V| + \left|\bigcup_{v \in V} L_v\right|$ vertices such that $\mathcal{H}$ is colorable if and only if G is L-colorable.

F61: [TuVoZh02] Every colorable mixed hypergraph is the induced subhypergraph of a uniquely colorable mixed hypergraph. More generally, given a colorable mixed hypergraph $\mathcal{H}$ and two of its coloring partitions, say P with t classes and P' with t' classes such that P' is a refinement of P (i.e., every class in P' is a subset of some class in P), there exists a mixed hypergraph $\mathcal{H}'$ with $\chi(\mathcal{H}') = t$ and $\overline{\chi}(\mathcal{H}') = t'$ that contains $\mathcal{H}$ as an induced subhypergraph, moreover $\mathcal{H}$ has just one coloring with t and also with t' colors, and those two colorings induce P and P' on $\mathcal{H}$.

F62: [JiMuTuVoWe02] For every finite set $S \subset \mathbb{N}\backslash\{1\}$ there exists a mixed hypergraph $\mathcal{H}$ such that $\mathcal{H}$ admits a strict k-coloring if and only if $k \in S$.

F63: ([KrKrPrVo00]; A. Niculitsa) If each edge of a mixed hypergraph $\mathcal{H}$ induces a subtree in some fixed tree, then the set of values k for which $\mathcal{H}$ has a strict k-coloring consists of *consecutive* integers.

REMARK

R15: Starting with [MiTu97], several papers study the strict colorings of Steiner systems (block designs). For a servey, please see [MiTuVo03].

Clique Hypergraphs

DEFINITION

D40: The ***clique hypergraph*** of graph $G = (V, E)$ has the same vertex set V; a subset $H \subseteq V$ of cardinality at least two is considered to be a hyperedge if it induces a maximal complete subgraph in G. Its chromatic number will be denoted by $\chi_C(G)$.

OPEN PROBLEM

P4: [DuSaSaWo91] Does there exist a constant k such that $\chi_C(G) \leq k$ for every *perfect* graph G?

FACTS

F64: [BaGrGyPrSe] The following upper bounds are valid on $\chi_C(G)$ for every graph G: the domination number plus one, the independence number unless G is a complete graph or $G = C_5$, and $2\sqrt{n}$ (on n vertices).

F65: $\chi_C(G) \leq 2$ holds for comparability graphs ([DuSaSaWo91]), claw-free graphs without induced odd cycles longer than 3 and also the complements of such graphs ([BaGrGyPrSe]).

F66: [DuKiTr91] For the complements of comparability graphs, $\chi_C(G) \leq 3$.

F67: [MoSk99] If the graph G is planar, then $\chi_C(G) \leq 3$, and the clique hypergraph is 4-choosable also if G is imbeddable in the projective plane.

5.2.5 Algorithmic Complexity

FACTS

F68: Bipartite (i.e., 2-colorable) graphs can be recognized and properly 2-colored in linear time.

F69: The 2-choosable graphs can be recognized in linear time (by the structural characterization theorem in [ErRuTa79]).

F70: [Ma68, FiSa69] The coloring number can be determined in polynomial time.

F71: [Ka72] For every $k \geq 3$, it is NP-complete to decide whether $\chi(G) \leq k$. Also, it is NP-complete to decide whether a planar graph of maximum degree 4 is 3-colorable ([GaJoSt76]).

F72: [Ho81] ($k = 3$), [LeGa83] ($k \geq 4$) It is NP-complete to decide whether a k-regular graph is edge k-colorable.

F73: [BrLo98] For every fixed k, it can be decided in polynomial time whether G admits a vertex partition into k-element sets, each of which is independent in G or in its complement $\overline{G}$.

F74: [GrLoSc84] The chromatic number of perfect graphs can be determined in polynomial time.

F75: The 3-colorability of graphs without any induced P_6 subgraph can be decided, and a 3-coloring can be found if it exists, in polynomial time ([RaSc01]). On the other hand, deciding the 4-colorability of P_{12}-free graphs and the 5-colorability of P_8-free graphs are NP-complete ([SgWo01]).

F76: [KrTu02] For every $k \geq 3$, it is NP-complete to decide whether the clique hypergraph of a *perfect* graph G with $\omega(G) = k$ is 2-colorable.

F77: [KrTu02] If G is planar, then the decision of 2-colorability (and the determination of chromatic number) of its clique hypergraph is polynomial-time solvable.

F78: Planar graphs can be properly 5-colored, and also properly 4-colored, in polynomial time. On the other hand ([VeWe92]), it is #P-complete to determine the number of proper 4-colorings of a planar graph.

F79: [Th94] Given a planar graph with a 5-assignment L on its vertices, a proper L-coloring can be found in linear time.

REMARK

R16: The 5-coloring and 5-list-coloring algorithms are efficient. Though the proofs of the Four Color Theorem also yield polynomial algorithms that 4-color a planar graph, so far no 'practical' 4-coloring algorithm is known.

FACTS

F80: [KrKrTuWo01] For a fixed graph H, let $Forb(H)$ be the class of graphs not containing any induced subgraphs isomorphic to H. Determining $\chi(G)$ for the graphs $G \in Forb(H)$ is polynomial if H is an induced subgraph of P_4 or of $P_3 \cup K_1$ (the path of length two plus an isolated vertex), and NP-complete for any other H.

F81: [HeNe90] For every non-bipartite graph H it is NP-complete to decide whether an input graph G is H-colorable.

F82: [DyGr00] The number of homomorphisms $G \to H$ (where H is fixed and G is the input graph) can be determined in polynomial time if each component of H is a complete graph with loops at all of its vertices, or a complete bipartite graph without loops, or an isolated vertex; and it is #P-complete otherwise.

F83: [HeNeZh96] Suppose that the following property holds for the digraph H: A digraph G is *not* H-colorable if and only if there exists an oriented tree T such that $T \to G$ and $T \not\to H$. Then H-colorability is decidable in polynomial time.

F84: [GuWeWo92] There exist oriented trees T such that it is NP-complete to decide whether $G \to T$.

F85: For every $k \geq 3$, it is Π_2^p-complete to decide whether $ch(G) \leq k$ ([GuTa97]). It remains Π_2^p-complete for $k = 4$ on planar graphs and for $k = 3$ on triangle-free planar graphs ([Gu96]).

F86: [JaSc97] It is NP-complete to decide whether a complete bipartite graph with given lists on its vertices is colorable.

F87: [KrTu94] The List Coloring problem remains NP-complete on the instances satisfying all the following three conditions (also if restricted to planar graphs): every list has at most 3 colors, every color occurs in at most 3 lists, and every vertex has degree at most 3. On the other hand, both the decision and search versions of the problem can be solved in linear time if every list has at most 2 colors, or every color occurs in at most 2 lists, or every vertex has degree at most 2.

F88: [Lo73] For every $k \geq 2$, it is NP-complete to decide whether a hypergraph is k-colorable. It remains NP-complete for $k = 2$ on 3-uniform hypergraphs.

F89: [TuVoZh02] It is NP-complete to decide whether a mixed hypergraph is colorable; and given a mixed hypergraph $\mathcal{H}$ with a strict coloring, it is co-NP-complete to decide whether $\mathcal{H}$ is uniquely colorable.

F90: [Ma02b] For every $k \geq 2$, on the clique hypergraphs of graphs it is Σ_2^p-complete to test k-colorability, and Π_3^p-complete to decide whether a given list assignment on the vertices admits a proper list clique coloring. (To decide whether the clique hypergraph of every induced subgraph is k-colorable, is also Π_3^p-complete.)

F91: On an unrestricted input graph it is NP-complete to determine the chromatic sum, but it is polynomial on trees ([KuSc89]) and also on the line graphs of trees ([GiKu00]).

F92: [Ja97] If the cost set contains at least four colors, then on bipartite graphs it is NP-hard to determine the minimum cost of a proper coloring.

F93: [HuTu93] Precoloring extension on the complements of bipartite graphs, and also on split graphs, is solvable with exactly the same efficiency (in polynomial time) as the Bipartite Matching problem; but it is NP-complete on bipartite graphs. It remains NP-complete on bipartite graphs even if just 3 colors may be used ([Kr93]).

F94: [BiHuTu92] On interval graphs, 1-PrExt is solvable in polynomial time, but 2-PrExt is NP-complete. (On unit interval graphs, the unrestricted PrExt problem is known to be NP-complete [Ma02a].)

F95: 1-PrExt is polynomial-time solvable on chordal graphs ([Ma03]) but NP-complete on permutation graphs ([Ja97]).

F96: [BoDeJaKlKrMüTu98] It is NP-hard to determine the ranking number of bipartite graphs, and also of complements of bipartite graphs.

F97: [LaYu98] The ranking number of line graphs is NP-hard to compute.

F98: The ranking number can be determined in polynomial time for line graphs of trees ([dToGrSc95]), for graphs contained in chordal graphs of bounded clique size ([BoDeJaKlKrMüTu98]), interval graphs ([AsHe94]), and graphs in which there is only a polynomially bounded number of minimal separators ([BrKlKrMü02]).

F99: [KrTu99] It is NP-complete to decide whether an acyclic, planar directed graph has ranking number at most 3 (while for any k, the *undirected* connected graphs of ranking number at most k can be recognized in constant time [BoDeJaKlKrMüTu98]).

Approximation

DEFINITIONS

D41: Let $r(n) : \mathbb{N} \to \mathbb{R}^+$ be a function. An algorithm is an ***r(n)-approximation*** for chromatic number if, for every n and every input graph G with n vertices it outputs an integer k such that $\chi(G) \leq k \leq r(n)\chi(G)$. Analogous terminology applies to any minimization problem, e.g., to determine $ch(G)$.

D42: A ***doubly-periodic graph*** is an infinite graph whose vertices are labelled $v_{ij\ell}$ ($i, j \in \mathbb{Z}$, $\ell \in \{1, \ldots, n\}$), the subgraphs induced by $\{v_{ij1}, v_{ij2}, \ldots, v_{ijn}\}$ — called cells — are isomorphic for all pairs i, j, any other edge joins neighboring cells (i.e., cells (i, j) and (i', j') where $|i-i'| \leq 1$ and $|j-j'| \leq 1$), and both mappings $i \mapsto i+1$ and $j \mapsto j+1$ are automorphisms of G.

FACTS

F100: [BeGoSu98] Unless P = NP, no polynomial-time $\mathcal{O}(n^{1/7-\varepsilon})$-approximation exists for $\chi(G)$, with any $\varepsilon > 0$.

F101: [FeKi98] Unless ZPP = NP, no polynomial-time $\mathcal{O}(n^{1-\varepsilon})$-approximation exists for $\chi(G)$, with any $\varepsilon > 0$. The analogous result is valid for uniform hypergraphs too ([KrSu98]).

F102: [KhLiSa00] Unless $\mathsf{P} = \mathsf{NP}$, no polynomial-time algorithm can possibly color the k-colorable graphs with $k + 2\lfloor k/3 \rfloor - 1$ colors.

F103: The chromatic number can be approximated in polynomial time within the ratio $\mathcal{O}(n(\log\log n)^2/(\log n)^3)$ ([Ha93]), and also within

$$\max\left\{\mathcal{O}(n/\log^{m-1} n), \mathcal{O}(\Delta \log\log n/\log n)\right\}$$

for any fixed m ([Pa01]).

F104: The 3-chromatic graphs can be properly $\mathcal{O}(n^{3/14}\log^{\mathcal{O}(1)} n)$-colored in polynomial time ([BlKa97]). Moreover, there is a randomized polynomial-time algorithm that colors any graph with at most

$$\min\left\{\mathcal{O}(\Delta^{1-2/k}\log^{1/2}\Delta \log n), \mathcal{O}(n^{1-3/(k+1)}\log^{1/2} n)\right\}$$

colors, where $k = \chi(G) \geq 3$ and $\Delta = \Delta(G)$ ([KaMoSu98]).

F105: Due to the inequalities $c \log col(G) \leq ch(G) \leq col(G)$, the choice number is constant-approximable on classes of graphs with bounded choice number, which is closely related to assuming bounded average degree.

F106: [DuFü97] The difference $n - \chi(G)$ is approximable within 360/289.

F107: [BoGiHaKl95] The ranking number can be approximated within $\mathcal{O}(\log^2 n)$.

F108: [Ki98] For every k there is an on-line algorithm that properly colors every k-colorable graph with at most $\mathcal{O}(n^{1-1/k!})$ colors. For $k = 3$ and $k = 4$ the bound can be improved to $\mathcal{O}(n^{2/3}\log^{1/3} n)$ and $\mathcal{O}(n^{5/6}\log^{1/6} n)$, respectively.

F109: [Bu84] For every integer $k \geq 3$, there exists a doubly-periodic planar graph G of maximum degree 4 and a properly colored finite subgraph $F \subset G$ such that it is undecidable whether the coloring of F can be extended to a proper k-coloring of G. An analogous result holds for the undecidability of whether a partial homomorphism $F \rightarrow H$ can be extended to a homomorphism $G \rightarrow H$, whenever H is a finite non-bipartite graph ([DuEmGi98]). For bipartite H, the necessary and sufficient conditions of (un)decidability are not known.

References

[AlGrHa00] M. O. Albertson, S. Grossman, and R. Haas, Partial list colorings, *Discrete Math.*, 214 (2000), 235–240.

[AlHu79] M. O. Albertson and J. P. Hutchinson, The three excluded cases of Dirac's map-color theorem, *Ann. New York Acad. Sci.*, 318 (1979), 7–17.

[AlTa92] N. Alon and M. Tarsi, Colorings and orientations of graphs, *Combinatorica*, 12 (1992), 125–134.

[AlTuVo97] N. Alon, Zs. Tuza, and M. Voigt, Choosability and fractional chromatic numbers, *Discrete Math.*, 165/166 (1997), 31–38.

[ApHa77] K. Appel and W. Haken, Every planar map is four colorable. Part I: Discharging, *Illinois J. Math.*, 21 (1977), 429–490.

[ApHaKo77] K. Appel, W. Haken, and J. Koch, Every planar map is four colorable. Part II: Reducibility, *Illinois J. Math.*, 21 (1977), 491–567.

[AsHe94] B. Asprall and P. Heggernes, Finding minimum height elimination trees for interval graphs in polynomial time, *BIT* 34 (1994), 484–509.

[BaGrGyPrSe] G. Bacsó, S. Gravier, A. Gyárfás, M. Preissmann, and A. Sebő, Coloring the maximal cliques of graphs, *SIAM J. Discr. Math.*, to appear.

[Be78] J. Beck, On 3-chromatic hypergraphs, *Discrete Math.*, 24 (1978), 127–137.

[BeGoSu98] M. Bellare, O. Goldreich, and M. Sudan, Free bits, PCPs and non-approximability - towards tight results, *SIAM J. Comp.* 27 (1998), 804–915.

[Bi12] G. D. Birkhoff, A determinantal formula for the number of ways of coloring a map, *Annals of Math.*, 14 (1912), 42–46.

[BiHuTu92] M. Biró, M. Hujter, and Zs. Tuza, Precoloring extension. I. Interval graphs, *Discrete Math.*, 100 (1992), 267–279.

[BlKa97] A. Blum and D. Karger, An $\widetilde{\mathcal{O}}(n^{3/14})$-coloring algorithm for 3-colorable graphs, *Inf. Proc. Letters*, 61 (1997), 49–53.

[BoDeJaKlKrMüTu98] H. L. Bodlaender, J. S. Deogun, K. Jansen, T. Kloks, D. Kratsch, H. Müller, and Zs. Tuza, Rankings of graphs, *SIAM J. Discr. Math.*, 11 (1998), 168–181.

[BoGiHaKl95] H. L. Bodlaender, J. R. Gilbert, H. Hafsteinsson, and T. Kloks, Approximating treewidth, pathwidth and minimum elimination tree height, *J. Algorithms*, 18 (1995), 238–255.

[BoKlTaLe00] H. L. Bodlaender, T. Kloks, R. B. Tan, and J. van Leeuwen, Approximations for λ-coloring of graphs, Proc. STACS 2000, Lecture Notes in Computer Sci. Vol. 1770, 2000, 395–406. (extended version to appear)

[BöMoSt99] T. Böhme, B. Mohar, and M. Stiebitz, Dirac's map-color theorem for choosability, *J. Graph Theory*, 32 (1999), 327–339.

[BoFiJuKaMo02] D. Bokal, G. Fijavž, M. Juvan, P. M. Kayll, and B. Mohar, The circular chromatic number of a digraph, manuscript, 2002 (to appear).

[Bo79] O. V. Borodin, On acyclic colorings of planar graphs, *Discrete Math.*, 25 (1979), 211–236.

[Bo89] O. V. Borodin, Representing K_{13} as a 2-pire map on the Klein bottle, *J. Reine Angew. Math.*, 393 (1989), 132–133.

[BoBrFrMiSe97] M. Borowiecki, I. Broere, M. Frick, P. Mihók, and G. Semanišin, Survey of hereditary properties of graphs, *Discuss. Math. Graph Theory*, 17 1997, 5–50.

[BrKlKrMü02] H. J. Broersma, T. Kloks, D. Kratsch, and H. Müller, A generalisation of AT-free graphs and a generic algorithm for solving triangulation problems, *Algorithmica* 32 (2002), 594–610.

[BrLo98] K. Bryś and Z. Lonc, Clique and anticlique partitions of graphs, *Discrete Math.*, 185 (1998), 41–49.

[Bu84] S. A. Burr, Some undecidable problems involving the edge-coloring and vertex-coloring of graphs, *Discrete Math.*, 50 (1984), 171–177.

[CaEd97] N. Cairnie and K. Edwards, Some results on the achromatic number, *J. Graph Theory*, 26 (1997), 129–136.

[CaEd98] N. Cairnie and K. Edwards, The achromatic number of bounded degree trees, *Discrete Math.*, 188 (1998), 87–97.

[ChKu96] G. J. Chang and D. Kuo, The $L(2,1)$-labeling problem on graphs, *SIAM J. Discr. Math.*, 9 (1996), 309–316.

[ChLiZh99] G. J. Chang, D. D.-F. Liu, and X. Zhu, Distance graphs and T-coloring, *J. Combin. Theory B*, 75 (1999), 259–269.

[Ch99] G. G. Chappell, A lower bound for partial list colorings, *J. Graph Theory*, 32 (1999), 390–393.

[ChGeHe71] G. Chartrand, D. P. Geller, and S. T. Hedetniemi, Graphs with forbidden subgraphs, *J. Combin. Theory*, 10 (1971), 12–41.

[ChSe79] C. A. Christen and S. M. Selkow, Some perfect colouring properties of graphs, *J. Combin. Theory B*, 27 (1979),49–59.

[ChGaJo78] V. Chvátal, M. R. Garey, and D. S. Johnson, Two results concerning multicoloring, *Annals of Discrete Math.*, 2 (1978), 151–154.

[dToGrSc95] P. de la Torre, R. Greenlaw, and A. A. Schäffer, Optimal edge ranking of trees in polynomial time, *Algorithmica* 13 (1995), 592–618.

[Di52] G. A. Dirac, Map colour theorem, *Canad. J. Math.*, 4 (1952), 480–490.

[DuKiTr91] D. Duffus, H. A. Kierstead, and W. T. Trotter, Fibres and ordered set coloring, *J. Combin. Theory A*, 58 (1991), 158–164.

[DuSaSaWo91] D. Duffus, B. Sands, M. Sauer, and R. E. Woodrow, Two-colouring all two-element maximal antichains, *J. Combin. Theory A*, 57 (1991), 109–116.

[DuFü97] R. Duh and M. Fürer, Approximation of k-set cover by semi-local optimization, *29th STOC*, Proc. Ann. ACM Symp. on Theory of Computing, ACM, 1997, 256–265.

[DuEmGi98] P. Dukes, H. Emerson, and G. MacGillivray, Undecidable generalized colouring problems, *J. Combin. Math. Combin. Comput.*, 26 (1998), 97–112.

[DyGr00] M. Dyer and C. Greenhill, The complexity of counting graph homomorphisms, *Random Struct. Alg.*, 17 (2000), 260–289.

[Ed97] K. Edwards, The harmonious chromatic number and the achromatic number, pages 13–47 in *Surveys in Combinatorics*, London Math. Soc. Lecture Note Ser. Vol, 241, London Math. Soc., 1997.

[Er64] P. Erdős, On a combinatorial problem II., *Acta Math. Acad. Sci. Hungar.*, 15 (1964), 445–447.

[ErLo75] P. Erdős and L. Lovász, Problems and results on 3-chromatic hypergraphs and some related questions, pp. 609–627 in A. Hajnal, R. Rado and V. T. Sós (Eds.), *Infinite and Finite Sets*, Colloq. Math. Soc. J. Bolyai 10, North-Holland, 1975.

[ErRuTa79] P. Erdős, A. L. Rubin, and H. Taylor, Choosability in graphs, Proc. West-Coast Conference on Combinatorics, Graph Theory and Computing, Arcata, California, *Congr. Numer.*, XXVI (1979), 125–157.

[FeKi98] U. Feige and J. Kilian, Zero knowledge and the chromatic number, *J. Comput. System Sci.*, 57 (1998), 187–199.

[FiSa69] H.-J. Finck and H. Sachs, Über eine von H. S. Wilf angegebene Schranke für die chromatische Zahl endlicher Graphen, *Math. Nachr.*, 39 (1969), 373–386.

[Fi95] D. Fisher, Fractional colorings with large denominators, *J. Graph Theory*, 20 (1995), 403–409.

[Fr34] P. Franklin, A six-color problem, *J. Math. Phys.*, 13 (1934), 363–369.

[Ga80] M. Gardner, Mathematical Games, *Sci. Amer.*, 242 (2) (1980), 14–21.

[GaJoSt76] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer, Some simplified NP-complete graph problems, *Theor. Computer Sci.*, 1 (1976), 237–267.

[GiKu00] K. Giaro and M. Kubale, Edge-chromatic sum of trees and bounded cyclicity graphs, *Inf. Proc. Letters*, 75 (2000), 65–69.

[GrYe92] J. R. Griggs and R. K. Yeh, Labeling graphs with a condition at distance 2, *SIAM J. Discr. Math.*, 5 (1992), 586–595.

[GrLoSc84] M. Grötschel, L. Lovász, and A. Schrijver, Polynomial algorithms for perfect graphs, pages 325–356 in *Topics on Perfect Graphs*, North-Holland Math. Stud., 88, North-Holland, Amsterdam, 1984.

[Gr59] H. Grötzsch, Ein Dreifarbensatz für dreikreisfreie Netze auf der Kugel, *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe.* 8 (1959), 109–120.

[GuWeWo92] W. Gutjahr, E. Welzl, and G. Woeginger, Polynomial graph colourings, *Discrete Applied Math.*, 35 (1992), 29–46.

[Gu96] S. Gutner, The complexity of planar graph choosability, *Discrete Math.*, 159 (1996), 119–130.

[GuTa97] S. Gutner and M. Tarsi, Some remarks on $(a : b)$-choosability, manuscript, 1997.

[GyLe91] A. Gyárfás and J. Lehel, Effective on-line coloring of P_5-free graphs, *Combinatorica*, 11 (1991), 181–184.

[Ha93] M. M. Halldórsson, A still better performance guarantee for approximate graph coloring, *Inf. Proc. Letters*, 45 (1993), 19–23.

[He93] L. S. Heath, Edge coloring planar graphs with two outerplanar graphs, 2nd ACM–SIAM Symp. on Discrete Algorithms (San Francisco, 1991), ACM, 1991, 195–202.

[He:1890] P. J. Heawood, Map colour theorem, *Quart. J. Pure Appl. Math.*, 24 (1890), 332–338.

[He:1898] P. J. Heawood, On the four-colour map theorem, *Quart. J. Pure Appl. Math.*, 29 (1898), 270–265.

[HeHeBe82] S. M. Hedetniemi, S. T. Hedetniemi, and T. Beyer, A linear algorithm for the Grundy (coloring) number of a tree, Proc. 13th S-E Conf. on Combinatorics, Graph Theory and Computing, *Congr. Numer.* 36 (1982), 351–363.

[HeNe90] P. Hell and J. Nešetřil, On the complexity of H-coloring, *J. Combin. Theory B*, 48 (1990), 92–110.

[HeNeZh96] P. Hell, J. Nešetřil, and X. Zhu, Duality and polynomial testing of tree homomorphisms, *Trans. Amer. Math. Soc.*, 348 (1996), 1281–1297.

[Ho81] I. Holyer, The NP-completeness of edge-coloring, *SIAM J. Computing*, 10 (1981), 718–720.

[HuTu93] M. Hujter and Zs. Tuza, Precoloring extension. II. Graph classes related to bipartite graphs, *Acta Math. Univ. Comen.* 62 (1993), 1–11.

[HuTu96] M. Hujter and Zs. Tuza, Precoloring extension. III. Classes of perfect graphs, *Combin. Probab. Comput.*, 5 (1996), 35–56.

[IrMa99] R. W. Irving and D. F. Manlove, The b-chromatic number of a graph, *Discrete Applied Math.*, 91 (1999), 127–141.

[JaRi83] B. Jackson and G. Ringel, Maps of m-pires on the projective plane, *Discrete Math.*, 46 (1983), 15–20.

[JaRi84] B. Jackson and G, Ringel, Solution of Heawood's empire problem in the plane, *J. Reine Angew. Math.*, 347 (1984), 146–153.

[JaRi85] B. Jackson and G. Ringel, Heawood's empire problem, *J. Combin. Theory B*, 38 (1985), 168–178.

[Ja97] K. Jansen. Optimum cost chromatic partition problem, *Algorithms and Complexity*, Lecture Notes in Computer Sci. Vol. 1203, 25–36.

[JaSc97] K. Jansen and P. Scheffler, Generalized colorings for tree-like graphs, *Discrete Applied Math.*, 75 (1997), 135–155.

[Ja01] J. C. M. Janssen, A partial solution of a partial list colouring problem, Proc. 32nd S-E International Conf. on Combinatorics, Graph Theory and Computing, *Congr. Numer.* 152 (2001), 75–79.

[JiMuTuVoWe02] T. Jiang, D. Mubayi, Zs. Tuza, V. Voloshin, and D. B. West, The chromatic spectrum of mixed hypergraphs, *Graphs Combin.*, 18 (2002), 309–318.

[JiWe99] T. Jiang and D. B. West, Coloring of trees with minimum sum of colors, *J. Graph Theory*, 32 (1999), 354–358.

[Ka00] J. Kahn, Asymptotics of the list-chromatic index for multigraphs, *Random Struct. Alg.*, 17 (2000), 117–156.

[KaMoSu98] D. Karger, R. Motwani, and M. Sudan, Approximate graph coloring by semidefinite programming, *J. ACM*, 45 (1998), 246–265.

[Ka72] R. Karp, Reducibility among combinatorial problems, pages 85–104 in R. E. Miller and J. W. Thatcher (Eds.), *Complexity of Computer Computations*, Plenum Press, 1972.

[KhLiSa00] S. Khanna, N. Linial, and S. Safra, On the hardness of approximating the chromatic number, *Combinatorica*, 20 (2000), 393–415.

[Ki98] H. A. Kierstead, On-line coloring k-colorable graphs, *Israel J. Math.*, 105 (1998), 93–104.

[Ki98a] H. A. Kierstead, Recursive and on-line graph coloring, *Handbook of Recursive Mathematics*, Vol. 2, North-Holland, 1998, 1233–1269.

[KiPeTr94] H. A. Kierstead, S. G. Penrice, and W. T. Trotter, On-line coloring and recursive graph theory, *SIAM J. Discr. Math.*, 7 (1994), 72–89.

[KoKr01] G. Kortsarz and R. Krauthgamer, On approximating the achromatic number, *SIAM J. Discr. Math.*, 14 (2001), 408–422.

[KoSt00] A. V. Kostochka and M. Stiebitz, On the number of edges in colour-critical graphs and hypergraphs, *Combinatorica*, 20 (2000), 521–530.

[KrKrPrVo00] D. Král', J. Kratochvíl, A. Proskurowski, and H.-J. Voss, Coloring mixed hypertrees, pages 279–289 in *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Sci. Vol. 1928, Springer, 2000.

[KrKrTuWo01] D. Král', J. Kratochvíl, Zs. Tuza, and G. J. Woeginger, Complexity of coloring graphs without forbidden induced subgraphs, pages 254–262 in A. Brandstädt and V. B. Le, *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Sci. 2204, Springer, 2001.

[Kr93] J. Kratochvíl, Precoloring extension with fixed color bound, *Acta Math. Univ. Comen.* 62 (1993), 139–153.

[KrTu94] J. Kratochvíl and Zs. Tuza, Algorithmic complexity of list colorings, *Discrete Applied Math.*, 50 (1994), 297–302.

[KrTu99] J. Kratochvíl and Zs. Tuza, Rankings of directed graphs, *SIAM J. Discr. Math.*, 12 (1999), 374–384.

[KrTu02] J. Kratochvíl and Zs. Tuza, On the complexity of bicoloring clique hypergraphs of graphs, *J. Algorithms*, 45 (2002), 40–54.

[KrTuVo02] J. Kratochvíl, Zs. Tuza, and M. Voigt, On the b-chromatic number of graphs, pages 310–320 in L. Kučera, Ed., *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Sci. Vol. 2573, 2002.

[KrSu98] M. Krivelevich and B. Sudakov, Approximate coloring of uniform hypergraphs, *Proc. 6th Ann. European Symp. on Algorithms*, ESA'98, 1998, 477–489.

[KrVu01] M. Krivelevich and V. H. Vu, Choosability in random hypergraphs, *J. Combin. Theory B*, 83 (2001), 241–257.

[KuSc89] E. Kubicka and A. J. Schwenk, An introduction to chromatic sum, pages 39–45 in *Proc. 17th ACM Computer Science Conf.*, 1989.

[LaYu98] T. W. Lam and F. L. Yue, Edge ranking of graphs is hard, *Discrete Applied Math.*, 85 (1998), 71–86.

[LeGa83] D. Leven and Z. Galil, NP-completeness of finding the chromatic index of regular graphs, *J. Algorithms*, 4 (1983), 35–44.

[Lo73] L. Lovász, Coverings and colourings of hypergraphs, Proc. 4th S–E Conf. on Combinatorics, Graph Theory and Computing, *Congr. Numer.*, 8 (1973), 3–12.

[Lo75] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Math.*, 13 (1975), 383–390.

[Ma02] D. Marx, The complexity of tree multicoloring, *Proc. MFCS 2002*, Lecture Notes in Computer Sci. Vol. 2420, 532–542.

[Ma02a] D. Marx, The disjoint paths problem, manuscript, 2002.

[Ma02b] D. Marx, The complexity of clique colorings, manuscript, 2002.

[Ma03] D. Marx, Precoloring extension on chordal graphs, manuscript, 2003.

[Ma68] D. W. Matula, A min-max theorem for graphs with application to graph coloring, *SIAM Review*, 10 (1968), 481–482.

[MiTu97] L. Milazzo and Zs. Tuza, Upper chromatic number of Steiner triple and quadruple systems, *Discrete Math.*, 174 (1997), 247–259.

[MiTuVo03] L. Milazzo, Zs. Tuza, and V. Voloshin, Strict colorings of Steiner triple and quadruple systems: a survey, *Discrete Math.*, 261 (2003), 399–411.

[Mi67] G. J. Minty, A theorem on three-coloring the edges of a trivalent graph, *J. Combin. Theory*, 2 (1967), 164–167.

[MiMo97] J. Mitchem and P. Morriss, On the cost-chromatic number of graphs, *Discrete Math.*, 171 (1997), 201–211.

[MiMoSc97] J. Mitchem, P. Morriss, and E. Schmeichel, On the cost chromatic number of outerplanar, planar, and line graphs, *Discuss. Math. Graph Theory*, 17 (1997), 229–241.

[MoSk99] B. Mohar and R. Škrekovski, The Grötzsch theorem for the hypergraph of maximal cliques, Electron. J. Combin. 6 (1999), R26, 13 pp.

[NeTa99] J. Nešetřil and C. Tardif, Density, pages 229–235 in R. L. Graham et al. (Eds.), *Contemporary Trends in Discrete Mathematics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 49, Amer. Math. Soc., 1999.

[Pa01] V. T. Paschos, A note on the approximation ratio of graph-coloring, *Found. Comput. Decision Sci.*, 26 (2001), 267–271.

[RaSr00] J. Radhakrishnan and A. Srinivasan, Improved bounds and algorithms for hypergraph 2-coloring, *Random Struct. Alg.*, 16 (2000), 4–32.

[RaSc01] B. Randerath and I. Schiermeyer, 3-colorability $\in \mathcal{P}$ for P_6-free graphs, *Discrete Applied Math.*, to appear.

[ReSe98] B. Reed and P. Seymour, Fractional colouring and Hadwiger's conjecture, *J. Combin. Theory B*, 74 (1998), 147–152.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Natl. Acad. Sci. USA*, 60 (1968), 438–445.

[RoSaSeTh97] N. Robertson, D. P. Sanders, P. D. Seymour, and R. Thomas, The four-colour theorem, *J. Combin. Theory B*, 70 (1997), 2–44.

[Ry90] J. Ryan, Fractional total colouring, *Discrete Applied Math.*, 27 (1990), 287–292.

[Se81] P. D. Seymour, Nowhere-zero 6-flows, *J. Combin. Theory B*, 30 (1981), 130–135.

[SgWo01] J. Sgall and G. J. Woeginger, The complexity of coloring graphs without long induced paths, *Acta Cybern.* 15 (2001), 107–111.

[Si83] G. J. Simmons, On the ochromatic number of a graph, Proc. 14th S-E Conf. on Combinatorics, Graph Theory and Computing, *Congr. Numer.* 40 (1983), 339–366.

[Ta:1880] P. G. Tait, On the colouring of maps, *Proc. Royal Soc. Edinburgh Sect. A*, 10 (1878–1880), 501–503.

[Te93] B. A. Tesman, List T-colorings of graphs, *Discrete Applied Math.*, 45 (1993), 277–289.

[Th94] C. Thomassen, Every planar graph is 5-choosable, *J. Combin. Theory B*, 62 (1994), 180–181.

[Th95] C. Thomassen, 3-list-coloring planar graphs of girth 5, *J. Combin. Theory B*, 64 (1995), 101–107.

[ThErAlMaSc89] C. Thomassen, P. Erdős, Y. Alavi, P. J. Malde, and A. J. Schwenk, Tight bounds on the chromatic sum of a connected graph, *J. Graph Theory*, 13 (1989), 353–357.

[Tu50] W. T. Tutte, On the imbedding of linear graphs in surfaces, *Proc. London Math. Soc. (2)*, 51 (1950), 474–483.

[Tu54] W. T. Tutte, A contribution to the theory of chromatic polynomials, *Canad. J. Math.*, 6 (1954), 80–91.

[Tu70] W. T. Tutte, More about chromatic polynomials and the golden ratio, pages 439–453 in R. K. Guy et al. (Eds.), *Combinatorial Structures and their Applications*, Gordon and Breach, 1970.

[Tu90] Zs. Tuza, Contractions and minimal k-colorability, *Graphs Combin.*, 6 (1990), 51–59.

[TuVo96] Zs. Tuza and M. Voigt, On a conjecture of Erdős, Rubin and Taylor, *Tatra Mountain Math. Publ.*, 9 (1996), 69–82.

[TuVo96a] Zs. Tuza and M. Voigt, Every 2-choosable graph is $(2m,m)$-choosable, *J. Graph Theory*, 22 (1996), 245–252.

[TuVo00] Zs. Tuza and V. Voloshin, Uncolorable mixed hypergraphs, *Discrete Applied Math.*, 99 (2000), 209–227.

[TuVoZh02] Zs. Tuza, V. Voloshin, and H. Zhou, Uniquely colorable mixed hypergraphs, *Discrete Math.*, 248 (2002), 221–236.

[VeWe92] D. L. Vertigan and D. J. A. Welsh, The computational complexity of the Tutte plane: the bipartite case, *Combin. Probab. Comput.*, 1 (1992), 181–187.

[Vo93] M. Voigt, List colourings of planar graphs, *Discrete Math.*, 120 (1993), 215–219.

[Vo95] M. Voigt, A not 3-choosable planar graph without 3-cycles, *Discrete Math.*, 146 (1995), 325–328.

[Vo02] V. I. Voloshin, *Coloring Mixed Hypergraphs: Theory, Algorithms and Applications*, Amer. Math. Soc., Providence, 2002.

[Wa96] A. O. Waller, An upper bound for list T-coloring, *Bull. London Math. Soc.*, 28 (1996), 337–342.

[Wo01] D. R. Woodall, List colourings of graphs, *Proc. British Combinatorial Conference*, London Math. Soc. Lecture Note Series Vol. 288, Cambridge University Press, 2001.

[Zh01] X. Zhu, Circular chromatic number: a survey, *Discrete Math.*, 229 (2001), 371–410.

5.3 INDEPENDENT SETS AND CLIQUES

Gregory Gutin, Royal Holloway, University of London, UK

5.3.1 Basic Definitions and Applications
5.3.2 Integer Programming Formulations
5.3.3 Complexity and Approximation
5.3.4 Bounds on Independence and Clique Numbers
5.3.5 Exact Algorithms
5.3.6 Heuristics
References

Introduction

Finding maximum cliques and maximum independent sets are among the most applicable problems in graph theory. We give an overview of both algorithmic and theoretical results on these problems.

5.3.1 Basic Definitions and Applications

In this section, all graphs are *simple*, i.e., they do not have self-loops or multi-edges.

Some Combinatorial Optimization Problems

DEFINITIONS

D1: For a graph G, a set S of vertices is an ***independent set*** if no two vertices in S are adjacent.

D2: The number of vertices in a maximum-size independent set of G is called the ***independence number*** of G and is denoted $ind(G)$.

D3: A ***clique*** in a graph G is a maximal set of mutually adjacent vertices of G. The ***clique number***, denoted $\omega(G)$, is the number of a vertices in a largest clique of G.

D4: A ***vertex cover*** in a graph G is a set S of vertices such that at least one endpoint of every edge of G is in S.

D5: A ***matching*** in a graph G is a set of mutually non-adjacent edges of G.

REMARKS

R1: The ***maximum-clique problem*** (determining $\omega(G)$ for a given graph G) and the ***maximum-independent-set problem*** (determining $ind(G)$) are the main problems considered in this section. Observe that $\omega(G) = ind(\overline{G})$ for any graph G, where $\overline{G}$ denotes the edge-complement graph.

R2: Also considered in this section is the ***minimum-vertex-cover problem*** of finding a vertex cover of minimum cardinality.

R3: Sometimes we will consider graphs with non-negative weights on their vertices.

EXAMPLE

E1: It is easy to verify for the Petersen graph G shown in Figure 5.3.1 that $\omega(G) = 2$, $ind(G) = 4$, and a minimum vertex cover has size 5.

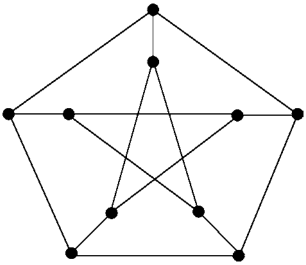


Figure 5.3.1 **The Petersen graph.**

Vertex-Weighted Graphs

DEFINITIONS

D6: A graph G is ***vertex-weighted*** if every vertex x is assigned a non-negative weight $w(x)$; the graph is denoted (G, w).

D7: The ***weight of a vertex set*** S in a vertex-weighted graph is the sum of the weights of the vertices in S.

NOTATION: The weight of a maximum-weight independent set in G is denoted $ind(G, w)$. The weight of a maximum-weight clique in G is denoted $\omega(G, w)$.

REMARKS

R4: The ***maximum-weight-independent-set problem*** (determining $ind(G, w)$) generalizes the unweighted one: assign weight 1 to every vertex of a graph. A similar remark holds for the ***maximum-weight-clique problem***.

R5: The first three of the following facts are immediate consequences of the definitions.

FACTS

F1: For every vertex-weighted graph (G, w), $ind(G, w) = \omega(\overline{G}, w)$. Thus, an obvious duality between the two problems.

F2: A set S of vertices in a graph $G = (V, E)$ is a vertex cover if and only if $V - S$ is an independent set.

F3: The size of a maximum matching in a graph G is less or equal to the size of a minimum cover of G.

F4: [Ko31,Eg31] If G is a bipartite graph, then the maximum size of a matching in G equals the minimum cardinality of a vertex cover of G.

Applications Involving Hamming Distance

There are numerous and varied applications of the combinatorial optimization problems introduced above. Perhaps among the most studied are those from coding theory.

DEFINITIONS

D8: The ***Hamming distance*** between a pair $u = (u_1, \ldots, u_n)$, $v = (v_1, \ldots, v_n)$ of binary vectors is the number of indices i for which $u_i \neq v_i$.

D9: The vertex set of the ***Hamming graph*** $H(n, d)$ consists of all binary vectors with n coordinates. A pair u, v of vertices in $H(n, d)$ are adjacent if the Hamming distance between them is at least d.

REMARKS

R6: The Hamming graph is of interest for error-correcting codes: A binary code consisting of a set of binary vectors, any pair of which have Hamming distance at least d can correct $\lfloor (d-1)/2 \rfloor$ errors. [MaSl79]

R7: A natural question arises: How many vectors with n coordinates can be in a code in which any two vectors are at least Hamming distance d apart? It is obvious from the definitions that this number equals the order of a maximum clique in $H(n, d)$.

R8: For further discussions and results of this application, see, for example, [BoBuPaPe99] and [Os01]. Other applications include fault diagnosis [BePe90,HaPaVa93], machine learning [HoSk89,HaJa90], and detecting embedded network structures in linear programs [GuGuMiMa00,GuGuMiZv].

5.3.2 Integer Programming Formulations

The simplest formulation of the maximum-weight-clique problem is based on the edges of the input graph.

Edge-based formulation: Let $G = (V, E)$ be a vertex-weighted graph with $V = \{v_1 \ldots, v_n\}$ and weights $w_i = w(v_i)$. The maximum weight of a clique can be found by solving the following integer program:

$$\begin{aligned} \max w = &\sum_{i=1}^{n} w_i x_i \\ \text{subject to } \; & x_i + x_j \leq 1 \quad \forall \{v_i, v_j\} \notin E \\ & x_i = 0 \text{ or } 1, \quad i = 1, \ldots, n \end{aligned}$$

FACTS

F5: In the edge-based formulation, every feasible solution x corresponds to the vertex set S of a complete subgraph of G as follows: $x_i = 1$ if and only if $v_i \in S$.

F6: [NeTr74,NeTr75] Let x be an optimum $(0, \frac{1}{2}, 1)$-valued solution to the linear relaxation of the edge formulation, and let $J = \{j : \; x_j = 1\}$. Then there exists an optimal solution x^* to the edge-based formulation such that $x^*_j = 1$ for every $j \in J$.

REMARKS

R9: Unfortunately, the result in Fact 6 above appears to be of relatively minor computational value since optimal solutions of the linear relaxation of the edge-based formulation normally have only a small number of integer components, and the gap between optimal solutions of the edge-based formulation and its linear relaxation is usually too large. [BoBuPaPe99]

R10: The obvious alteration transforms the edge-based formulation to the corresponding formulation of the maximum-weight-independent-set problem, which is clearly equivalent to the following nonlinear optimization problem first studied in [Sh90].

Shor formulation:

$$\begin{aligned} \min w = &\sum_{i=1}^{n} w_i x_i \\ \text{subject to } \; & x_i x_j = 0 \;\; \forall \{v_i, v_j\} \in E \\ & x_i^2 - x_i = 0 \;\; i = 1, \ldots, n \end{aligned}$$

COMPUTATIONAL NOTE: Shor [Sh90] reported very good computational results using his formulation.

Two More Formulations of the Maximum-Clique Problem

While the formulations above are relatively straightforward, the following ones initiated by Motzkin and Straus [MoSt65] are less obvious.

DEFINITIONS

D10: The ***standard simplex*** Δ in R^n is defined as follows:

$$\Delta = \{x \in R^n : \; x_i \geq 0, \; i = 1, \ldots, n, \; \sum_{i=1}^{n} x_i = 1\}$$

NOTATION: For a graph G with adjacency matrix A_G and a vector $x \in R^n$, let $g(x) = x^T A_G x$.

D11: Let $G = (V, E)$ be a graph with vertices $v_1, \ldots, v_n$ and let $S \subseteq V$ be arbitrary. The ***characteristic vector*** $x^S \in R^n$ is defined as follows: $x^S = (x_1^S, x_2^S, \ldots, x_n^S)$, where $x_i^S = \frac{1}{|S|}$ if $v_i \in S$ and $x_i^S = 0$, otherwise.

FACTS

F7: **Motzkin-Straus Theorem** [MoSt65]: Let G be a graph and let $x^* = argmax\{g(x) : \; x \in \Delta\}$ (i.e., the $x \in \Delta$ for which $g(x)$ is maximum). Then

$$\omega(G) = \frac{1}{1 - g(x^*)} \geq \frac{1}{1 - g(x)} \;\; \forall x \in \Delta$$

Moreover, a subset S of vertices of G is a maximum clique if and only if

$$x^S = argmax\{g(x) : \; x \in \Delta\}$$

NOTATION: For a graph G with adjacency matrix A_G, let $f(x) = x^T A_G x + (x^T x)/2$.

F8: **Bomze Theorem** [Bo97]: Let S be a subset of vertices of a graph G. Then

(a) S is a maximum clique in G if and only if $x^S = argmax\{f(x) : \ x \in \Delta\}$.

(b) S is a clique in G if and only if x^S is a local maximizer of $\{f(x) : \ x \in \Delta\}$.

(c) All local maximizers x of $\{f(x) : \ x \in \Delta\}$ are characteristic vectors.

REMARK

R11: One drawback of the Motzkin-Straus formulation $\{g(x) : \ x \in \Delta\}$ is the fact that some solutions of this optimization problem are not characteristic vectors [PaPh90, PeJa95]. Thus, Bomze's variation of the Motzkin-Straus formulation is of interest.

5.3.3 Complexity and Approximation

The maximum-clique problem is one of the first shown to be NP-hard [Ka72]. Since then many researchers have tried to gain a more precise understanding of the difficulty of the problem (see, e.g., [AuCrGaKaMaPr99, BoBuPaPe99]). One of the strongest indicators of its considerable difficulty is given in Fact 9.

NOTATION: (a) If π is an algorithm for the maximum-independent-set problem, then $\pi(G)$ denotes the independent set produced by π when the input graph is G.

(b) If ψ is an algorithm for the maximum-clique problem, then $\psi(G)$ denotes the clique produced by ψ when the input graph is G.

FACTS

F9: [Ha99] Let ψ be any polynomial-time algorithm for the maximum-clique problem. Unless P = NP, for any $\epsilon \in (0, 1/2]$, there exists an n-vertex graph G such that $\psi(G)$ has fewer than $\omega(G)/n^{1/2-\epsilon}$ vertices.

F10: [BoHa92] There is a polynomial-time algorithm π such that for any n-vertex graph G, $ind(G)/|\pi(G)| = O(n/\log^2 n)$.

F11: [Fi] There is a polynomial-time algorithm π such that for any n-vertex graph G, $ind(G)/|\pi(G)| = O(n(\log\log n)^2/\log^3 n)$.

REMARK

R12: Fact 10 has a nice (and short) proof based on some ideas of Paul Erdős (see [Ha98]).

Some Results Involving Maximum Degree

Since the approximation in Fact 11 is still very weak, it is natural to consider the approximation for graphs for which certain parameters are restricted. One such parameter is the maximum degree $\Delta(G)$ of a graph G. For an overview of approximation algorithms whose performance is measured in terms of $\Delta(G)$, see [LaTi01]. Facts 12 and 13 are two of the currently best results.

FACTS

F12: [AlFeWiZu95] Unless P = NP, there exists a constant $\epsilon > 0$ such that there is no polynomial-time algorithm π for which $ind(G)/|\pi(G)| = O(\Delta(G)^{\epsilon})$ for every graph G.

F13: Vishvanathan (see [Ha98]) There is a polynomial-time algorithm π such that for every graph G, $ind(G)/|\pi(G)| = O(\Delta(G) \log\log \Delta(G)/ \log \Delta(G))$.

F14: [GuVaYe03] Let ψ be any polynomial-time algorithm for the maximum-clique problem and let $p(n)$ be any polynomial function of n. Unless P = NP, there exists an n-vertex graph G such that $\psi(G)$ has fewer vertices than at least $\frac{p(n)-1}{p(n)}$ of the complete subgraphs of G. The analogous fact holds for the minimum-vertex-cover problem.

REMARKS

R13: There is a simple polynomial-time algorithm for the minimum-vertex-cover problem that provides a 2-approximation (i.e., no worse than twice the optimum): find a maximum matching M in a given graph G and output the vertices of M as a vertex cover of G. For slightly better approximation results, see [AuCrGaKaMaPr99].

R14: The preceding remark and Fact 9 are, in a way, at odds with each other. The maximum-clique and the minimum-vertex-cover problems are dual, in a sense (via the maximum-independent-set problem as noted earlier). Nevertheless, while the former cannot be approximated to any good degree, the latter can be. This “strange” situation is somewhat resolved by Fact 14. Observe that a feasible solution of the maximum-clique problem is a set of vertices that induces a complete subgraph.

5.3.4 Bounds on Independence and Clique Numbers

Every maximum-clique or independent-set heuristic provides an “algorithmic” lower bound to the corresponding problem. In this subsection, we consider “analytical” ones that require only certain parameters of the input graph.

DEFINITIONS

D12: The ***Perron root***, denoted $\lambda_P(G)$, is the largest eigenvalue of A_G.

D13: The ***(open) neighborhood*** of a vertex v in a graph G, denoted $N(v)$, is the set of vertices adjacent to v, and the ***closed neighborhood*** $N[v] = N(v) \cup \{v\}$.

Lower Bounds

Caro and Wei obtained the lower bound given in Fact 15, and Alon and Spencer [AlSp92] gave an elegant probabilistic proof of that bound. Recently, Sakai, Togasaki, and Yamazaki generalized Fact 15 to vertex-weighted graphs (Fact 16).

FACTS

F15: [Ca79,We81] Let $G = (V, E)$ be a graph. Then $ind(G) \geq \sum_{v \in V} 1/(deg(v) + 1)$.

F16: [SaToYa03] Every vertex-weighted graph $G = (V, E)$ contains an independent set S of weight at least $\max\{\sum_{x \in V} w(x)/(deg(x) + 1), \sum_{x \in V} w(x)^2/[\sum_{y \in N[x]} w(y)]\}$. Moreover, such a set can be found in polynomial time.

F17: [Se94] Let $G = (V, E)$ be a graph. Then

$$ind(G) \geq \sum_{v \in V} \frac{1}{deg(v)+1}(1 + \max\{0, \frac{deg(v)}{deg(v)+1} - \sum_{u \in N(v)} \frac{1}{deg(u)+1}\})$$

F18: The adjacency matrix of a connected graph G is irreducible, symmetric, and has all non-negative entries; hence, all its eigenvalues are real (see, e.g., [HoJo85]).

F19: [Wi86] For a connected graph G on n vertices, $\omega(G) \geq \frac{n}{n - \lambda_P(G)}$. (This bound was recently improved by Budinich [Bu].)

Upper bounds

FACTS

F20: [Wi67] For a connected graph G, $\omega(G) \leq \lambda_P(G) + 1$. Equality holds if and only if G is complete.

NOTATION: For a connected graph G, let $\Lambda_{-1}(G)$ denote the number of eigenvalues of A_G that do not exceed -1.

F21: [AmHa72] For a connected graph G, $\omega(G) \leq \Lambda_{-1}(G) + 1$. Equality holds if and only if G is complete multipartite.

F22: [Bu] For a connected graph G on n vertices, $\omega(G) \leq n - \frac{1}{2} rank A_{\overline{G}}$.

F23: Each of the three upper bounds given above can be computed in time $O(n^3)$, and one can find examples that show the bounds are sharp [Bu].

REMARK

R15: Budinich [Bu] tested the upper bounds on a set of 700 random graphs of order 100 and 200. For these graphs, the smaller of the first two upper bounds was almost always better than the third upper bound.

5.3.5 Exact Algorithms

Clique Enumeration

Harary and Ross [HaRo57] initiated an algorithmic and theoretical study of the enumeration of all cliques in a graph. This topic has a variety of applications (see, e.g., [Bo64,HaRo57,PaUn59]). The first significant theoretical result (Fact 24) is due to Erdös, Moon, and Moser [MoMo65]. The currently best theoretical results regarding the performance of clique-enumeration algorithms are given in Facts 25 and 26.

DEFINITION

D14: Given a graph G, the ***arboricity*** is the minimum number of edge-disjoint acyclic subgraphs whose union is G.

FACTS

F24: [MoMo65] The maximum number of cliques in an n-vertex graph equals

$$\begin{cases} 3^{n/3} & \text{if } n \equiv 0 \ (mod\ 3) \\ 4 \cdot 3^{(n-4)/3} & \text{if } n \equiv 1 \ (mod\ 3) \\ 2 \cdot 3^{(n-2)/3} & \text{if } n \equiv 2 \ (mod\ 3) \end{cases}$$

(Some extensions of this result are discussed in [SaVa].)

F25: [ChNi85] There is an algorithm for listing all cliques of a graph $G = (V, E)$ in time $O(a(G)|E|\mu(G))$, where $a(G)$ is the arboricity of G and $\mu(G)$ is the number of cliques in G.

F26: [ToTaTa88] There is an algorithm for listing all cliques of an n-vertex graph in time $O(3^{n/3})$.

REMARKS

R16: The algorithm in Fact 25 improves slightly on the running time $O(|V||E|\mu(G))$ of the algorithm given in [TsIdAvSh77].

R17: The algorithm in Fact 26 is a modification of the backtracking algorithm of Bron and Kerbosch [BrKe73]. In light of the Erdős-Moon-Moser result in Fact 24, the result in Fact 26 is, in a sense, best possible.

COMPUTATIONAL NOTE: In computational experiments with graphs of order ranging from 30 to 220, Loukakis [Lo83] showed that his depth-first enumerative algorithm is much faster than the algorithms from [BrKe73], [TsIdAvSh77], and [LoTs81]. Loukakis's algorithm seems to be among the most efficient practical algorithms.

Maximum-Clique and Maximum-Weight-Clique Algorithms

Clearly, the algorithms mentioned previously can be used (directly or after simple modifications) to find a maximum clique in a graph. However, the maximum-clique problem has attracted more attention than the clique-enumeration problem and the use of certain procedures has made algorithms for the maximum-clique problem quite fast.

FACT

F27: [Ro86] There is an algorithm for solving the maximum-clique problem with time complexity $O(n^{0.276})$, where n is the number of vertices.

REMARKS

R18: Fact 27 was established by Robson [Ro86] by modifying a recursive algorithm of Tarjan and Trojanowski [TaTr77] and by using a detailed case analysis. The result seems to be the best time-complexity upper bound currently known for maximum-clique algorithms.

R19: Branch-and-cut algorithms were used with great success for several combinatorial optimization problems, see, e.g., §4.6. However, for the maximum-clique problem, branch-and-cut algorithms currently remain, in general, inferior to the state-of-the-art branch-and-bound algorithms [RoSm01].

R20: There are several quite efficient branch-and-bound algorithms that use fast coloring heuristics to produce upper and lower bounds. See, e.g., the new algorithm in [Os02], which uses a coloring heuristic by Biggs [Bi90]. Additional information and references may be found in [Os02].

R21: Branch-and-bound algorithms for the maximum-weight-clique problem are discussed in [BoBuPaPe99,Os01]. In [Os01], a new algorithm is compared with a few others.

R22: Some algorithms have been compared on random graphs and on special families of graphs inspired by applications. Frequently occurring special families of graphs for the maximum-clique problem may be found in [JoTr96] and are discussed in [HaPaVa93]. Östergård [Os01] introduces a special family of instances for the maximum-weight-clique problem.

R23: There are many algorithms for the maximum-clique and maximum-weight-clique problems. Some partial comparisons seem to suggest that the relative performance of various algorithms vary for different graph densities and instances (see, e.g., [Os01,Os2]).

COMPUTATIONAL NOTE: There are a few maximum-clique and maximum-weight-clique computer codes freely available for research purposes (see, e.g., [Di] and [NiOs03]).

5.3.6 Heuristics

When the instance of the maximum-weight clique problem under consideration is of large size or the data is not precise (which is the case in many applications) or a solution has to be obtained very quickly, one should resort to heuristics rather than exact algorithms. Moreover, heuristics form important parts of many exact algorithms.

Construction Heuristics and Local Search

DEFINITIONS

D15: A ***construction heuristic*** produces a feasible solution without any attempt to improve it.

D16: A ***local search (LS)*** heuristic starts from a feasible solution and, in each iteration until termination, chooses the next solution from a *neighborhood* of solutions that are, in some prescribed sense, close to the current solution.

D17: An ***improvement LS*** (or simply ***local improvement***) is a local search that always chooses a solution that is better than the current one and terminates when it cannot find one in the neighborhood.

REMARKS

R24: Construction heuristics are normally very fast, and they provide quick solutions and lower or upper bounds. However, their solutions cannot be expected to be of high quality.

R25: The simplest construction heuristic for the maximum-clique problem is to add one vertex at a time to an emerging clique. It is logical to choose in each iteration an eligible vertex of maximum degree [KoRu87]. Alternatively, one may delete vertices from the given graph one by one until a clique is obtained [KoRu87].

R26: Most approximation algorithms for various optimization problems are, in fact, construction heuristics. Examples of construction heuristics used in maximum-clique approximation algorithms may be found in [BoHa92] and [Fi].

COMPUTATIONAL NOTE: Local improvement does not appear to perform particularly well for the maximum-clique problem [GuGuMiZv] as it may terminate at a relatively small clique. Local-search algorithms that do not require monotonic improvement are much more flexible in that they can escape from local optima that are non-maximum cliques; perhaps the most flexible among them is briefly discussed below.

Tabu Search

This *metaheuristic* appears to provide a good trade-off between computational time and solution quality.

DEFINITION

D18: ***Tabu search*** is a local search in which solutions that are worse than the current one can be chosen provided that they are not in any of the so-called *tabu lists*.

REMARKS

R27: Tabu search was introduced independently by Glover [Gl89,Gl90] and Hansen and Jaumard [HaJa90]. "Pure" tabu search techniques for the maximum-clique problem were implemented in a number of papers (see, e.g., [FrHeWe90,SoGe96]).

COMPUTATIONAL NOTE: Tabu-search algorithms use parameters that have to be fine-tuned in order to achieve good results. This slows down the development and use of tabu-search computer codes. Battiti and Protasi [BaPr01] deal with this issue by adjusting the parameters using an internal learning loop.

R28: For brief descriptions and discussions of other metaheuristics applied to the maximum-clique problem, see [BoBuPaPe99].

References

[AlFeWiZu95] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman, Derandomized graph products, *Computational Complexity* 5 (1995), 60–75.

[AlSp92] N. Alon and J. H. Spencer, *The Probabilistic Method*, Wiley, 1992.

[AmHa72] A. T. Amin and S. L. Hakimi, Upper bounds on the order of a clique of a graph, *SIAM J. Appl. Math.* 22 (1972), 569–573.

[AuCrGaKaMaPr99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation*, Springer, 1999.

[BaPr01] R. Battiti and M. Protasi, Reactive local search for the maximum clique problem, *Algorithmica* 29 (2001), 610–637.

[BePe90] P. Berman and A. Pelc, Distributed fault diagnosis for multiprocessor systems, in *Proc. 20th Annual Int. Symp. Fault-Tolerant Comput.* (Newcastle, UK), (1990), 340–346.

[Bi90] N. Biggs, Some heuristics for graph coloring, in *Graph Colourings* (R. Nelson and R. J. Wilson, Eds.), Longman, 1990.

[Bo97] I. M. Bomze, Evolution towards the maximum clique, *J. Global Optim.* 10 (1997), 143–164.

[BoBuPaPe99] I. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, The Maximum Clique Problem, in *Handbook of Combinatorial Optimization (Supplement Volume A)* (D.-Z. Du and P. M. Pardalos, Eds.), Kluwer, 1999.

[Bo64] R. E. Bonner, On some clustering techniques, *IBM J. Res. Develop.* 8 (1964), 22–32.

[BrKe73] C. Bron and J. Kerbosch, Algorithm 457: Finding all cliques of an undirected graph, *Commun. ACM* 16 (1973), 575–577.

[Bu] M. Budinich, Bounds on the maximum clique of a graph, *Discrete Appl. Math.*, to appear.

[Ca79] Y. Caro, New results on the independence number, Tech. Report, Tel-Aviv University, 1979.

[ChNi85] N. Chiba and T. Nashizeki, Arboricity and subgraph listing algorithms. *SIAM J. Comput.* 14 (1985), 210–223.

[Di] C programs available at ftp://dimacs.rutgers.edu/pub/challenge/graph/solvers/

[Eg31] E. Egerváry, On combinatorial properties of matrices, *Math. Lapok* 31 (1931), 16–28.

[Fi] U. Fiege, Approximating maximum clique by removing subgraphs, Manuscript, 2002.

[FrHeWe89] C. Friden, A. Hertz, and M. de Werra, STABULUS: A technique finding stable sets in large graphs with tabu search. *Computing* 42 (1989), 35–44.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Gl89] F. Glover, Tabu search – Part I, *ORSA J. Comput.* 1 (1989), 190–260.

[Gl90] F. Glover, Tabu search – Part II, *ORSA J. Comput.* 2 (1990), 4–32.

[GuGuMiMa00] N. Gulpinar, G. Gutin, G. Mitra, and I. Maros, Detecting embedded network structures in linear programs, *Comput. Opt. Appl.* 15 (2000), 235–247.

[GuGuMiZv] N. Gulpinar, G. Gutin, G. Mitra, and A. Zverovitch, Extracting pure network submatrices in linear programs using signed graphs, *Discrete Appl. Math.*, to appear.

[GuVaYe03] G. Gutin, A. Vainshtein, and A. Yeo, Domination Analysis of Combinatorial Optimization Problems, *Discrete Appl. Math.* 129 (2003), 513–520.

[Ha98] M. Halldórsson, Approximation of independent sets in graphs, in *Proc. of APPROX'98*, (1998), 1–13.

[HaJa90] M.-H. Han and D. Jang, The use of maximum curvature points for the recognition of partially occluded objects, *Pattern Recognition* 23 (1990), 21–33.

[HaJa90] P. Hansen and B. Jaumard, Algorithms for the maximum satisfability problem, *Computing* 44 (1990) 279–303.

[HaPaVa93] J. Hasselberg, P. P. Pandalos, and G. Vairaktarakis, Test case generators and computational results for the maximum clique problem, *J. Global Optim.* 3 (1993), 463–482.

[HaRo57] F. Harary and I. C. Ross, A procedure for clique detection using the group matrix, *Sociometry* 20 (1957), 205–215.

[Ha99] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, *Acta Mathematica* 182 (1999), 105–142.

[HoJo85] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[HoSk89] R. Horaud and T. Skordas, Stereo correspondence through feature grouping and maximal cliques, *IEEE Trans. Pattern. Anal. Machine Intell.* 11 (1989), 1168–1180.

[JoTr96] D. S. Johnson, and M. Trick (Eds.), *Clique, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Vol. 26, AMS, 1996 (see also http://dimacs.rutgers.edu/Volumes/Vol26.html).

[Ka72] R. M. Karp, Reducibility among combinatorial problem, in *Complexity of Computer Computations* (R. E. Miller and J. W. Thatcher, Eds.), Plenum Press, 1972.

[Ko31] D. Kőnig, Graphen und matrizen, *Math. Lapok* 38 (1931), 116–119.

[KoRu87] R. Kopf and G. Ruhe, A computational study of the weighted independent set problem for general graphs, *Found. Control Eng.* 12 (1987), 167–180.

[LaTi01] H. Y. Lau and H. F. Ting, The greedier the better: an efficient algorithm for approximating maximum independent set, *J. Combin. Optim.* 5 (2001), 411–420.

[Lo83] E. Loukakis, A new backtracking algorithm for generating the family of maximal independent sets of a graph, *Comp. Math. Appl.* 9 (1983) 583–589.

[LoTs81] E. Loukakis and C. Tsouros, A depth first search algorithm to generate the family of maximal independent sets of a graph lexicographically, *Computing* 27 (1981), 249–266.

[MaSl79] J. MacWillimas and N. J. A. Slone, *The Theory of Error Correcting Codes*, North-Holland, 1979.

[MoMo65] J. W. Moon and L. Moser, On cliques in graphs, *Israel J. Math.* 3 (1965), 23–28.

[MoSt65] T. S. Motzkin and E. G. Straus, Maxima for graphs and a new proof of a theorem of Turán, *Canad. J. Math.* 17 (1965), 533–540.

[NeTr74] G. L. Nemhauser and L. E. Trotter, Properties of vertex packings and independence system polyhedra, *Math. Prog.* 6 (1974), 48–61.

[NeTr75] G. L. Nemhauser and L. E. Trotter, Vertex packings: Structural properties and algorithms, *Math. Prog.* 8 (1975), 232–248.

[NiOs03] S. Niskanen and P. R. J. Östergård, Cliquer User's Guide, Version 1.0, *Communications Lab., Helsinki Univ. Technology, Tech. Rep.* T48, 2003, The current release of the `Cliquer` code is available from `www.hut.fi/~pat/cliquer.html`

[Os01] P. R. Östergård, A new algorithm for the maximum-weight clique problem, *Nordic J. Comput.* 8 (2001), 424–436.

[Os02] P. R. Östergård, A fast algorithm for the maximum clique problem, *Discrete Appl. Math.* 120 (2002), 197–207.

[PaPh90] P. M. Pardalos and A. T. Phillips, A global optimization approach for solving the maximum clique problem, *Int. J. Comput. Math.* 33 (1990), 209–216.

[PaUn59] M. C. Paull and S. H. Unger, Minimizing the number of states in incompletely specified sequential switching functions, *IRE Trans. Electr. Comput.* EC-8 (1959), 356–367.

[PeJa95] M. Pelillo and A. Jagota, Feasible and infeasible maxima in a quadratic program for maximum clique, *J. Artif. Neural Networks* 2 (1995) 411–420.

[Ro86] J. M. Robson, Algorithms for maximum independent sets, *J. Algorithms* 7 (1986), 425–440.

[RoSm01] F. Rossi and S. Smriglio, A branch-and-cut algorithm for the maximum cardinality stable set problem, *Oper. Res. Lett.* 28 (2001), 63–74.

[SaVa] B. E. Sagan and V. R. Vatter, Maximal independent sets in graphs with at most r cycles, Preprint, 2002.

[SaToYa03] S. Sakai, M. Togasaki, and K. Yamazaki, A note on greedy algorithms for the maximum weighted independent set problem, *Discrete Appl. Math.* 126 (2003), 313–322.

[Se94] S. M. Selkow, A probabilistic lower bound on the independence number of graphs, *Discrete Math.* 132 (1994), 363–365.

[Sh90] N. Z. Shor, Dual quadratic estimates in polynomial and Boolean programming, in *Computational Methods in Global Optimization* (P. M. Pardalos and J. B. Rosen, Eds.), *Ann. Oper. Res.* 25 (1990), 163–168.

[SoGe96] P. Soriano and M. Gendreau, Tabu search algorithms for the maximum clique problem, in [JoTr96], 221–242, 1996.

[TaTr77] R. E. Tarjan and A. E. Trojanowski, Finding a maximum independent set, *SIAM J. Comput.* 13 (1977), 537–546.

[ToTaTa88] E. Tomita, A. Tanaka and H. Takahashi, The worst-case time complexity for finding all the cliques, *Tech. Report UEC-TR-C1* 1988.

[TsIdAvSh77] S. Tsukiyama, M. Ide, H. Aviyoshi and I. Shirakawa, A new algorithm for generating all maximum independent sets, *SIAM J. Comput.* 6 (1977), 505–517.

[We81] V. K. Wei, A lower bound on the stability number of a simple graph, Bell Lab. Tech. Memo., No. 81-11217-9, 1981.

[Wi67] H. S. Wilf, The eigenvalues of a graph and its chromatic number, *J. London Math. Soc.* 42 (1967), 330–332.

[Wi86] H. S. Wilf, The spectral bounds for the clique and independent numbers of graphs, *J. Combin. Theory B* 40 (1986), 113–117.

5.4 FACTORS AND FACTORIZATION

Michael Plummer, Vanderbilt University

5.4.1 Preliminaries
5.4.2 1-Factors
5.4.3 Degree Factors
5.4.4 Component Factors
5.4.5 Graph Factorization
References

Introduction

The vast body of work on factors and factorizations has much in common with other areas of graph theory. Indeed, factorization significantly overlaps the topic of *edge-coloring* (cf. §5.1), since any color class of a proper edge-coloring in a graph is just a matching. Moreover, the *Hamilton cycle problem* (cf. §4.5) can be viewed as the search for a connected 2-factor. Due to space constraints, we will treat factors of *finite undirected* graphs only. Nevertheless, several papers dealing with infinite graph factors and directed graph factors are included in our list of references.

5.4.1 Preliminaries

DEFINITIONS

D1: Given a graph (multigraph, general graph) G, we say that H is a ***factor*** of G if H is a spanning subgraph of G.

D2: A factor that is n-regular is called an ***n*-factor**.

D3: A factor defined only in terms of the degrees of its vertices is called a ***degree factor***.

D4: A factor described in terms of graph-theoretic properties other than its vertex degrees is called a ***component factor***.

D5: If a graph G can be represented as the edge-disjoint union of factors $F_1, F_2, \ldots F_k$, we shall refer to $\{F_1, F_2, \ldots F_k\}$ as a ***factorization*** of graph G.

FACTS

Today most workers in the field attribute the birth of graph factorization to two theorems of the Danish mathematician Julius Petersen. The analogous result of Bäbler for regular graphs of *odd degree* did not appear until almost fifty years later.

F1: [Pe1891] A 3-regular multigraph with at most one cutedge contains a 1-factor (and hence also a 2-factor).

F2: [Pe1891] Every $2k$-regular multigraph contains a 2-factor (and, hence, it has a factorization into 2-factors).

F3: [Bä38] Every 2-edge-connected $(2k+1)$-regular multigraph contains a 2-factor.

REMARKS

R1: The names *degree factors* and *component factors* for the two main categories of factors treated in the literature seem to be due to Akiyama and Kano [AkKan85a].

R2: These two main problem categories overlap. For example, finding a 1-factor and finding a factor each component of which is an edge amounts to the same thing.

R3: A thorough survey tracing the descendants of Petersen's factorization results for regular graphs may be found in [Vo95].

5.4.2 1-Factors

The most studied of degree factors are those in which each component is a single edge. We observe that §11.3 applies matchings to assignments. In our complementary approach here, we are principally interested in those properties of 1-factors that most naturally extend to analogous properties of more general factors.

Conditions for a Graph to Have a 1-Factor

DEFINITIONS

D6: A 1-***factor*** (or ***perfect matching***) of graph G is a set of vertex-disjoint edges in G which together span $V(G)$.

D7: The bipartite graph $K_{1,3}$ is often called a ***claw***. A graph containing no $K_{1,3}$ as an induced subgraph is said to be ***claw-free***.

D8: Graph G is said to have the ***odd-cycle property*** if every pair of odd cycles in G either have a vertex in common or are joined by an edge.

D9: The ***toughness*** of graph G, denoted by tough(G), is defined to be $+\infty$ when G is complete and otherwise to be

$$\min\{|S|/c(G-S)|S \subseteq V(G)\}$$

where the minimum is taken over all subsets $S \subseteq V(G)$ and $c(G-S)$ denotes the number of components of $G-S$.

D10: The ***binding number*** of a graph G, denoted bind(G), is defined to be

$$\min\left\{|N(X)|/|X| \;\middle|\; \emptyset \neq X \subseteq V(G) \text{ and } N(X) \neq V(G)\right\}$$

D11: We call a sequence of non-negative integers $d_1, \ldots, d_n$ ***graphical*** if there exists a graph G of order n the vertices of which have, in some order, degrees $d_1, \ldots, d_n$.

FACTS

Arguably, the most influential theorem in the study of 1-factors has been the seminal result called *Tutte's 1-factor Theorem.*

F4: [Tu47] **Tutte's 1-Factor Theorem**: A graph G has a 1-factor if and only if for each $S \subseteq V(G)$, $c_o(G-S) \leq |S|$, where $c_o(G-S)$ denotes the number of components of $G-S$ which have an odd number of vertices.

F5: [Pe1891] **Petersen's Theorem**: Every 2-edge-connected 3-regular multigraph has a 1-factor.

F6: [Bä38] Every $(r-1)$-edge-connected r-regular multigraph with an even number of vertices has a 1-factor. This generalizes Petersen's theorem.

F7: [Su74, Su76], [La75] If G is a connected claw-free graph of even order, then G has a 1-factor.

F8: [Su76] If G is an n-connected graph of even order, and if G has no induced subgraph isomorphic to the bipartite graph $K_{1,n+1}$, then G has a 1-factor.

F9: [FuHoMc65] If G is r-regular of even order and has the odd-cycle property, then G has a 1-factor.

F10: [Ni78, Ni79] If G is a k-connected graph ($k \geq 4$) of even order and if $\gamma(G) < k(k-2)/4$, then G has a 1-factor. (As in Chapter 7, $\gamma(G)$ denotes the (orientable) genus of G.)

F11: If G is of even order and tough$(G) \geq 1$, then G has a 1-factor. This follows immediately from Tutte's 1-factor Theorem.

F12: [Ku73], [Lo74] There exists a graph G having a 1-factor and degree sequence $d_1, d_2, \ldots, d_n$ if and only if both the sequences $d_1, \ldots, d_n$ and $d_1 - 1, \ldots, d_n - 1$ are graphical.

F13: (essentially due to Anderson [An73]) Let G be a graph of even order. If, for all $X \subseteq V(G)$,

$$|N(X)| \geq \min\left\{|V(G)|, \frac{4}{3}|X| - \frac{2}{3}\right\}$$

then G has a 1-factor. This theorem can be regarded as a binding number result.

F14: [LiGrHo75] If G is a connected graph of even order the automorphism group of which acts transitively on $V(G)$, then G has a 1-factor containing any given edge. Highly symmetric graphs of even order are guaranteed to have 1-factors by this result.

REMARKS

R4: A number of sufficient conditions quite similar to that of Anderson above are collected and compared in [Wo90]. A similar condition sufficient for a bipartite graph to have a k-factor (respectively, $[a,b]$-factor (see below)) may be found in [EnOtKan88] (respectively, [Kan90a]).

R5: There are now many papers investigating the existence of 1-factors containing or excluding specified edge sets. However, space does not permit us to treat these results and for the case of 1-factors, we direct the interested reader to two survey articles on the subject [Pl94, Pl96].

The Number of 1-Factors

DEFINITION

D12: A graph G is said to be ***bicritical*** if $G-x-y$ has a 1-factor for every choice of two different vertices x and y. (For further reading on bicritical graphs, see [LoPl86].)

NOTATION: $\Phi(G)$ denotes the number of 1-factors in graph G.

FACTS

F15: Let G be connected and have a unique 1-factor. Then:

(a) [Ko59] G has a cutedge belonging to the 1-factor;

(b) [LoPl86] G contains a vertex of degree $\leq \lfloor \log_2(p+1) \rfloor$; and

(c) [Hetyei (unpublished)] $|E(G)| \leq (|V(G)|/2)^2$.

F16: If G is k-connected and has a 1-factor, then either

(a) G has at least $k!$ 1-factors, or else

(b) G is bicritical.

It seems somewhat counterintuitive that bicritical graphs should be the exception here. It has proven much more difficult to bound $\Phi(G)$ in the bicritical case.

F17: If G is bicritical, then $\Phi(G) \geq |V(G)|/2+1$. Study of the *perfect matching polytope* of G, $PM(G)$, (see [LoPl86]) can be utilized to give the bound in this result.

F18: If the graph G is k-connected and contains a 1-factor, and if $|V(G)|$ is sufficiently large, then G has at least $k!$ 1-factors.

REMARKS

R6: Gabow, Kaplan and Tarjan [GaKaTa99, GaKaTa01] developed an $O(|E|\log^4|V|)$ algorithm to test whether a graph has a *unique* 1-factor and find it, if it exists.

R7: One can bound $\Phi(G)$ below by the matrix function called a *Pfaffian*. (For details, see [LoPl86; §8.3].) In the case when G is *planar*, the Pfaffian can be used to exactly compute $\Phi(G)$ in polynomial time.

R8: The connectivity of the graph G can also be employed to yield a lower bound on $\Phi(G)$ in some cases.

1-Factors in Bipartite Graphs

In the special case of bipartite graphs, the story of 1-factors has two principal historical roots, one in a result due to P. Hall [Ha35], and the other in a result due to König [Kö31, Kö33].

DEFINITIONS

D13: A ***vertex cover*** of a graph G is a subset $C \subseteq V(G)$ such that every edge of G has at least one endvertex in C.

D14: The ***vertex-covering number*** of a graph G is the size of any smallest vertex cover in G. Notation: $\tau(G)$.

D15: The ***matching number*** of a graph G is the size of any largest matching in G. Notation: $\nu(G)$

D16: The ***permanent*** of an $n \times n$ matrix A, denoted per A, is given by

$$\text{per } A = \sum a_{1\pi(1)} a_{2\pi(2)} \cdots a_{n\pi(n)}$$

where the sum extends over all permutations π of the set $\{1, \ldots, n\}$.

FACTS

F19: [Ha35] **Hall's Theorem**: Let G be a bipartite graph with vertex bipartition $V(G) = A \cup B$. Then G has a matching of A into B if and only if $|N(X)| \geq |X|$, for all $X \subseteq A$.

F20: [Fr12] **Marriage Theorem**: Let G be a bipartite graph with vertex bipartition $V(G) = A \cup B$. Then G has a 1-factor matching A onto B if and only if

(a) $|A| = |B|$ and

(b) $|N(X)| \geq |X|$, for all $X \subseteq A$. This earlier result of Frobenius is an immediate consequence of Hall's Theorem.

F21: It is clear that in any graph G, the matching number and the vertex-covering number are related by the inequality $\nu(G) \leq \tau(G)$.

F22: [Kö31, Kö33] **König's Theorem**: If G is bipartite, then $\nu(G) = \tau(G)$.

F23: [Ha48] Let G be a simple bipartite graph with bipartition $V(G) = A \cup B$, and assume that each vertex in A has degree at least k. If G has at least one 1-factor, then it has at least $k!$ 1-factors.

F24: Let G be a simple k-regular bipartite graph on $2n$ vertices. Then

$$n!\left(\frac{k}{n}\right)^n \leq \Phi(G) \leq (k!)^{n/k}$$

The first inequality is equivalent to the famous *van der Waerden Conjecture* [Wa26] on permanents, which was proved independently by [Fa81] and [Eg80, Eg81]. The second inequality was proved by [Br73].

F25: [Sc98b] If G is a k-regular bipartite graph of order $2n$, then

$$\Phi(G) \geq \left(\frac{(k-1)^{k-1}}{k^{k-2}}\right)^n$$

REMARKS

R9: In fact, it can be shown that Hall's Theorem and König's Theorem are equivalent.

R10: Since König's Theorem asserts the equality of the maximum of one quantity and the minimum of another, it is often referred to as a *minimax* theorem, especially in the study of linear programming. For an introduction to such ideas within the confines of graph theory, and for the associated *polytopal* ideas, see [LoPl86; Ch. 7 and 12].

5.4.3 Degree Factors

k-factors

DEFINITIONS

D17: A ***k-factor*** of a graph G is a k-regular subgraph that spans G.

D18: A graph G is ***hypohamiltonian*** (respectively, ***hypotraceable***) if G does not have a Hamilton cycle (respectively, path), but $G - v$ does, for all $v \in V(G)$.

FACTS

F26: [EnJaKatSa85] If tough $(G) \geq k$, then G has a k-factor. (This was conjectured by Chvátal [Ch73].)

F27: [Nis89a] Let G be a graph and k, an even non-negative integer. If

$$\kappa(G) \geq \max\{k(k+2)/2, (k+2)\alpha(G)/4\}$$

then G has a k-factor. ($\alpha(G)$ and $\kappa(G)$ are the independence number and the connectivity, respectively.)

F28: [IiNi91] Let k be a positive integer and let G be a graph of order $n \geq 4k - 5$, minimum degree at least k and kn even. Then if $\deg(u) + \deg(v) \geq n$, for each pair of non-adjacent vertices u and v, G has a k-factor. (The condition on degree sums is called an *Ore condition* after Ore who first introduced a condition of this type and showed it sufficient for the existence of a Hamilton cycle.)

F29: [IiNi97] Let $k \geq 2$ be an integer and let G be a connected graph of order n, minimum degree at least k and suppose kn is even. Suppose further that $n \geq 9k - 1 - 4\sqrt{2(k-1)^2 + 2}$. Then if $|N_G(u) \cup N_G(v)| \geq (1/2)(n + k - 2)$ for each pair of non-adjacent vertices u and v, G has a k-factor. (The sufficiency condition here is called a *neighborhood union condition.*)

F30: [Nis92] Let G be a connected graph of order n and let k be an integer ≥ 3 such that kn is even, $n \geq 4k - 3$ and $\delta(G) \geq k$. Then if $\max\{d(u), d(v)\} \geq n/2$, for all pairs of non-adjacent vertices u and v, G has a k-factor.

F31: [FaFaFlLiLi99] If a graph G is claw-free of order n with $\delta(G) \geq 4$, then G has a 2-factor with at most $[6n/(\delta(G) + 2)] - 1$ components. Moreover, there is an $O(n^3)$ algorithm to construct such a 2-factor.

F32: [Ku73] If k is a positive integer and the sequences $d_1, \ldots, d_n$ and $d_1 - k, \ldots, d_n - k$ are both graphical, then $d_1, \ldots, d_n$ can be realized by a graph G which contains a k-factor.

F33: [Kat83] If G is a graph and k is an even non-negative integer, then if $G - v$ has a k-factor for all $v \in V(G)$, G also has a k-factor.

F34: [Sa91] Suppose G is a graph with a 1-factor F and order at least four and let k be a positive integer. Then if $G - \{u, v\}$ has a k-factor for each edge $uv \in F$, G itself has a k-factor.

F35: [Kat83] If G is either hypohamiltonian or hypotraceable, then G has a 2-factor.

F36: [Nis90] Let G be a graph and m a non-negative integer. Then

(i) if $m \geq 2$ and m is even, and if $\deg(v) \geq m+1$, for all $v \in V(G)$, then $L(G)$ has a $2m$-factor;

(ii) if G is connected, $|E(G)|$ even, and $\deg(v) \geq m+2$ for all $v \in V(G)$, then $L(G)$ has a $(2m+1)$-factor.

F37: [Kat90b] Let G be a bipartite graph with bipartition $V(G) = X \cup Y$ and k be a positive integer. Then if:

(i) $|X| = |Y|$,

(ii) $\delta(G) \geq \lceil |X|/2 \rceil \geq k$, and

(iii) $|X| \geq 4k - 4\sqrt{k} + 1$, when $|X|$ is odd and $|X| \geq 4k - 2$, when $|X|$ is even,

then G has a k-factor.

F38: [Nis91] If $k \geq 2$ is an integer and G is a connected graph with $k|E(G)|$ even and if $\delta(L(G)) \geq (9k+12)/8$, then $L(G)$ has a k-factor.

F39: [Kan93] Let k be a positive integer and let G be a connected graph of order n and minimum degree at least k where kn is even and $n \geq 4k-3$. If for each pair of nonadjacent vertices u and v of G, $\deg(u) + \deg(v) \geq n$, then G has both a Hamilton cycle C and a k-factor F. Hence, G has a connected $[k, k+2]$-factor.

REMARKS

R11: [KatWo87, EgEn89] proved theorems similar to Fact 27, concerning the binding number.

R12: [KlWa73] gives an alternative proof of Fact 32 and also a polynomial algorithm for constructing the graph G containing the k-factor.

R13: Suppose that G is r-regular and has edge-connectivity λ. All values of k for which a multigraph G is guaranteed to have a k-factor are known [BoSaWo85]. Similarly, all such k are known when G is simple [NiesRa98].

R14: Hendry [He84] initiated the study of graphs with *unique* k-factors and his conjecture on the maximum number of edges that such a graph may have was proved by Johann [Joh00].

f-factors

Whereas a k-factor is a subgraph with the same degree at every vertex, an f-factor may have a prescription of different degrees.

DEFINITIONS

D19: Let G be a multigraph possibly with loops and f, a non-negative, integer-valued function on $V(G)$. Then a spanning subgraph H of G is called an ***f*-factor** of G if $\deg_H(v) = f(v)$, for all $v \in V(G)$.

D20: A set $S \subseteq V(G)$ such that $c_o(G-S) > |S|$ is called a ***1-barrier*** or ***antifactor set***. (Recall that by Tutte's 1-factor Theorem, a graph G with no 1-factor has a 1-barrier.)

NOTATION: Let $e_G(A, B)$ denote the number of edges in graph G joining vertex sets A and B.

FACTS

F40: [Tu54] **Tutte's f-Factor Theorem**: The graph G has an f-factor if and only if

$$f(D) - f(S) + d_{G-D}(S) - q_G(D, S, f) \geq 0$$

for all disjoint sets $D, S \subseteq V(G)$, where $q_G(D, S, f)$ denotes the number of components C of $G - (D \cup S)$ such that $e_G(V(C), S) + f(V(C)) \equiv 1(\bmod 2)$ and $f(D) - f(S) + d_{G-D}(S) - q_G(D, S, f) \equiv f(V(G))(\bmod 2)$, for all disjoint sets $D, S \subseteq V(G)$.

F41: [Tu81] A graph G has an f-factor if and only if it does not have an f-barrier. (An f-barrier is a generalization of a 1-barrier. We omit the details here.)

F42: [Ko01] Let G be a connected graph without multiple edges or loops and let p be an integer such that $0 < p < |V(G)|$. Let f be an integer-valued function on $V(G)$ such that $2 \leq f(v) \leq \deg_G(v)$ for all $v \in V(G)$. If every connected induced subgraph of order p of G has an f-factor, then G has an f-factor, or else $\sum_v f(v)$ is odd.

F43: [KatTs00] Let G be a graph and $a \leq b$, two positive integers. Suppose further that

(i) $\delta(G) \geq \frac{b}{a+b}|V(G)|$, and

(ii) $|V(G)| > \frac{a+b}{a}(b + a - 3)$.

Then if f is a function from $V(G)$ to $\{a, a+1, \ldots, b\}$ such that $\sum_v f(v)$ is even, G has an f-factor.

F44: [JaWh89] If G is a 2-edge-connected graph with a unique f-factor F, then some vertex has the same degree in F as in G.

REMARK

R15: There is a procedure for reducing the f-factor problem on a graph G to the 1-factor problem on a larger graph G'. See Chapter 10 of [LoPl86].

$[\mathbf{a}, \mathbf{b}]$-factors

DEFINITIONS

D21: Let a and b be integers such that $1 \leq a \leq b$. An $[a, b]$ ***factor*** of a graph G is a subgraph H such that $a \leq \deg_H(v) \leq b$, for all $v \in V(G)$. (Thus, it is an f-factor such that $a \leq f(v) \leq b$, for all $v \in V(G)$.)

D22: Let f be a function from $V(G)$ to the odd positive integers. A spanning subgraph F of graph G in which $\deg_F(v) \in \{1, 3, \ldots, f(v)\}$ is called a $(1, f)$-***odd-factor*** of G.

D23: A $[k, k+1]$-factor is sometimes called an ***almost regular*** (or ***semiregular***) factor.

D24: A graph G is an $[a, b]$-***graph*** if $a \leq \deg(v) \leq b$, for every vertex $v \in V(G)$.

FACTS

F45: [LiLi98] If G is a 2-connected claw-free graph, then G has a connected $[2,3]$-factor.

F46: [LiZhCh02] If G is a 2-connected claw-free graph containing a k-factor where $k \geq 2$, then G contains a connected $[k, k+1]$-factor.

F47: [KanSa83] Suppose that k, r, s and t are integers such that $0 \leq k \leq r$ and $1 \leq t$. If $ks \leq rt$, then an $[r, r+s]$-graph has a $[k, k+t]$-factor.

F48: [Nis94] If G is a graph, if a and b are integers such that $1 \leq a < b$, and if $\delta(G) \geq (\alpha(G)/2) + 1$, where $\alpha(G)$ is the independence number, then the line graph $L(G)$ has an $[a, b]$-factor.

F49: [La78, AmKan82] Let $n \geq 2$ be an integer, and let $i(G)$ denote the number of isolated vertices of graph G. The graph G has a $[1, n]$-factor if and only if $i(G-S) \leq n|S|$, for all $S \subset V(G)$. This is an analogue of Tutte's 1-factor theorem.

F50: [YuKan88] Let G be a graph and f a function from $V(G)$ to $\{1, 3, \ldots\}$. Then G has a $(1, f)$-odd-factor if and only if $c_o(G - S) \leq \sum_{v \in S} f(v)$, for all $S \subset V(G)$. This is another generalization of Tutte's 1-factor theorem. (See also [KaKa03].)

F51: [Lo70, Tu78] If G is r-regular, then G has a $[k, k+1]$-factor for all k, $0 \leq k \leq r$.

F52: [Th81] If G is an $[r, r+1]$-graph, then G has a $[k, k+1]$-factor for all k, $0 \leq k \leq r$.

$(\mathrm{g}, \mathbf{f})$-factors

DEFINITIONS

D25: Let G be a finite general graph, and let f, g be mappings of $V(G)$ into the non-negative integers. A ***(g, f)-factor*** of G is a spanning subgraph F such that $g(v) \leq \deg_F(v) \leq f(v)$ for all $v \in V(G)$.

D26: Let G be a finite general graph, and let f, g be mappings of $V(G)$ into the non-negative integers. Graph G is said to have ***all (g, f)-factors*** if and only if G has an h-factor for every h such that $g(v) \leq h(v) \leq f(v)$ for all $v \in V(G)$. Notice that if $f \equiv g \equiv 1$, then a (g, f)-factor (i.e., a $(1,1)$-factor) is just a 1-factor.

FACTS

F53: [Lo70] **(g, f)-factor theorem**: The graph G has a (g, f)-factor if and only if $e_G(V(C), S) + f(V(C)) \equiv 1(\bmod 2)$ and

$$f(D) - g(S) + \deg_{G-D}(S) - \hat{q}_G(D, S, g, f) \geq 0$$

for all pairs of disjoint sets $D, S \subseteq V(G)$, where $\hat{q}_G(D, S, g, f)$ denotes the number of components C of $G - (D \cup S)$ having $g(v) = f(v)$ for all $v \in V(C)$.

F54: [La78] let G be a graph and f and g two integer-valued functions defined on $V(G)$ such that $0 \leq g(x) \leq 1 \leq f(x)$. Then G contains a (g, f)-factor if and only if for every subset $X \subseteq V(G)$, the value $f(X)$ is at least equal to the number of connected components C of $G[V - X]$ such that either $C = \{x\}$ and $g(x) = 1$, or $|C|$ is odd and $|C| \geq 3$ and $g(x) = f(x) = 1$ for all $x \in C$.

F55: [EgKan96] Let G be a graph and f and g functions from $V(G)$ to the non-negative integers such that $g(v) \leq \deg_G(v), 0 \leq f(v)$ and $g(v) < f(v)$, for all $v \in V(G)$. If

$$\frac{g(x)}{\deg_G(x)} \leq \frac{f(y)}{\deg_G(y)}$$

for every pair of adjacent vertices x and y in G, then G has a (g, f)-factor.

F56: [Nies98] Let G be a multigraph and let g and f be as above. Then G has all (g, f)-factors if and only of

$$g(D) - f(S) + d_{G-D}(S) - q^*_G(D, S, g, f) \geq \begin{cases} -1, & \text{if } f \neq g; \\ 0, & \text{if } f = g, \end{cases}$$

for all disjoint sets $D, S \subset V(G)$, where $q^*_G(D, S, g, f)$ denotes the number of components C of $G - (D \cup S)$ such that there exists a vertex $v \in V(C)$ with $g(v) < f(v)$ or $e_G(V(C), S) + f(V(C)) \equiv 1(\bmod 2)$.

F57: [XuLiTo98] Let $n \geq 3$ be an integer and let G be a $K_{1,n}$-free graph. Let f and g be positive integer-valued functions on $V(G)$, such that $g(v) \leq f(v)$, for all $v \in V(G)$. If G has a (g, f)-factor, then G has a *connected* $(g, f + n - 1)$-factor.

F58: [An92] Let G be a graph with the odd-cycle property and with a k-regular spanning subgraph. Then

(i) for any even integer r, $1 \leq r < k$, or

(ii) when $|V(G)|$ is even, for any odd integer r satisfying $1 \leq r < k$,

G has an r-regular spanning subgraph.

REMARKS

R16: See also [AnNa98] for more sufficiency conditions for the existence of a (g, f)-factor, [An90, HeHeKiLi90] for simplified existence theorems for such factors, and [Kan84, Kan90b, Li89] for the existence of such a factor having additional properties such as including or excluding prescribed sets of edges.

R17: It is apparently unknown whether there is a polynomial algorithm to test if a graph G has all (g, f)-factors.

R18: For further information on the connections between network flows and graph factors see [FrJu99a, FrJu99b, FrJu99c, FrJu01, KoSt93].

Factors in Random Graphs

There are several popular models of so-called *random* graphs. We will be content to refer to only one of these.

DEFINITIONS

D27: Let $1, \ldots, n$ be a labelling of the vertices and let $\{e_{ij}\}, 1 \leq i < j \leq n$, be an array of independent random variables, where each e_{ij} assumes the value 1 with probability p and 0 with probability $1 - p$. This array determines a ***random graph*** on $\{1, \ldots, n\}$ where each (ij) is an edge if and only if $e_{ij} = 1$. It is denoted by $G_{n,p}$.

D28: An event E concerning a graph $G \in G_{n,p}$ is said to hold ***asymptotically almost surely*** (or **a.a.s.**), if $\lim_{n \to \infty}$ Prob $E = 1$.

FACTS

F59: [ErRé66] Let n be even and $p = (1/n)(\log n + w(n))$, with $\lim_{n\to\infty} w(n) = \infty$. Then $G \in G_{n,p}$ has a 1-factor a.a.s.

F60: [ShUp81] Let $p = (1/n)(\log n + (r-1)\log\log n + w(n))$, with $r \geq 1$ and suppose $\lim_{n\to\infty} w(n) = \infty$. Suppose further that f is a mapping from $V(G)$ into $\{1, \ldots, r\}$ with $\sum_{i=1}^{r} f(x_i)$ even. Then $G \in G_{n,p}$ has an f-factor a.a.s.

REMARKS

R19: For several excellent treatments of random graphs, including their factors, see [Ka82, Ka95, JaLuRu00, Bo85, Bo01, MoRe02, ShUp81, ShUp82].

R20: There are indeed infinite analogs of some of the matching and factor theorems for finite graphs. See [Ra49, Ah84a, Ah84b, Ah88, Ah91, AhMaSh92, AhNa84, AhNaSh83, Br71, HoPoSt87, St77, St85a, St85b, St89, Nied91, NiedPo94].

5.4.4 Component Factors

DEFINITIONS

D29: A ***path factor*** of graph G is a spanning subgraph of G each component of which is a path.

D30: An ***F-factor*** is a spanning subgraph in which each component is a single edge or an odd cycle.

FACTS

F61: [AnEgKaKawMa02] Let d be a non-negative integer and let G be a claw-free graph with $\delta(G) \geq d$. Then G has a path factor in which all paths have at least $d+1$ vertices.

F62: [St82] If G is a graph then G has an F-factor if and only if $|N(S)| \geq |S|$, for every independent $S \subseteq V(G)$. (This result can be viewed as a generalization of Hall's Theorem to the non-bipartite case.)

F63: [Mü79, HeKi81a] There is a polynomial algorithm for finding an F-factor or showing that none exists.

5.4.5 Graph Factorization

Roughly speaking, one could classify graph factorization problems as one of two kinds; those in which the edge set is partitioned and those in which the vertex set is partitioned.

Edge Partitions

DEFINITIONS

D31: A ***k*-*linear forest*** is a forest in which all components are paths of length at most k.

D32: The ***k*-*linear arboricity*** of a graph G is the minimum number of k-linear forests which partition $E(G)$.

CONJECTURES

The 1-Factorization Conjecture: Let G be a simple graph of even order n. If G is regular with $\Delta(G) \geq n/2$, then $\chi'(G) = \Delta(G)$; that is, G has a 1-factorization. (See [Wa97; Ch. 19].)

Conjecture [AkExHa80]: The linear arboricity of every d-regular graph is $\lceil (d+1)/2 \rceil$.

FACTS

F64: [ChHi89],[NiesVo90] If one replaces $(\sqrt{7}-1)n/2$ by $(\sqrt{7}-1)/2$, then the 1-Factorization Conjecture becomes true. This result is regarded as the best to date toward the conjecture.

F65: [PlTi91] Let G be a regular multigraph of even order n and multiplicity $\mu(G) \leq r$. Then if $\Delta(G) \geq r(5n/6+1)$, $\chi'(G) = \Delta(G)$. This result may be viewed as an extension of Fact F64 to the multigraph case.

F66: Given $\epsilon > 0$, there is a number $N = N(\epsilon)$ such that if G is a simple graph of even order greater than N, and $\Delta \geq (\frac{1}{2}+\epsilon)|V(G)|$, then G is 1-factorizable. This provides evidence in favor of the truth of the 1-Factorization Conjecture for "large" graphs. (See [PeRe97] and Häggkvist (unpublished).)

F67: [ZhZh92] Every k-regular graph of order $2n$ contains at least $\lfloor k/2 \rfloor$ edge-disjoint 1-factors, if $k \geq n$. This result represents another approach to the 1-Factorization Conjecture.

F68: [Kan85] Suppose a and b are integers such that $0 \leq a \leq b$. Then

(i) a graph G has a $[2a, 2b]$-factorization if and only if G is a $[2am, 2bm]$-graph, for some integer m; and

(ii) every $[8m+2k, 10n+2k]$-graph has a $[1, 2]$ factorization.

F69: [YaPaWoTo00] Let G be a multigraph and let g and f be two functions mapping $V(G)$ into the non-negative integers. Let m be a positive integer and ℓ an integer with $0 \leq \ell \leq 3$ and $\ell \equiv m(\bmod 4)$. If G is an $(mg + 2\lfloor m/4 \rfloor + \ell, mf - 2\lfloor m/4 \rfloor - \ell)$ graph, then G is (g, f)-factorizable. (See [Ya95] for other such results.)

F70: [Eg86] Let $k \geq 2$ be an integer. Then

(i) every r-regular graph G with $r \geq 4k^2$ has a $[2k, 2k+1]$-factorization; and

(ii) every $(k^2 - 4k + 2)$-regular graph G has a $[2k-1, 2k]$-factorization.

CONJECTURE

El-Zahár's Conjecture [El84]: If G is a graph with $n = n_1 + \cdots + n_k$ vertices and $\delta(G) \geq \lceil n_1/2 \rceil + \cdots + \lceil n_k/2 \rceil$, then G has a 2-factor in which the cycles have lengths $n_1, \ldots, n_k$ respectively.

El-Zahár himself proved the conjecture true in the case $k = 2$ and [CoHa63] when each $n_i = 3$. Further partial results can be found in [Johs00]. Abbasi [Ab98] proved the conjecture true in the case when $n = |V(G)|$ is sufficiently large.

REMARKS

R21: It is not difficult to show that the linear arboricity of a d-regular graph is at least the bound given above. It is the inequality in the opposite direction that has proved intractible so far. See [Al88, LiWo98] for further details.

R22: See [PlTi01] for extensions of 1-factorization to the multigraph case.

Vertex Partitions

When does a graph G admit a decomposition of its vertex set $V(G) = V_1 \cup \cdots V_k$ such that the induced subgraphs $G_i = G[V_i]$ have certain specified properties? We present several results of this genre.

FACTS

F71: [Gy78],[Lo77] Let G be a k-connected graph and suppose $v_1, \ldots, v_k$ are k distinct vertices of G. Suppose further that $|V(G)| = n = n_1 + \cdots + n_k$ is a partition of $|V(G)| = n$ into k positive parts. Then there exists a subgraph G' of G such that

(i) G' consists exactly of k components,

(ii) each of the components contains exactly one of the vertices v_i and

(iii) the component containing v_i contains exactly n_i vertices.

F72: [EnMa97b] Let G be a graph of order n and suppose $n = a_1 + \cdots + a_k$ is a partition of n where each $a_i \geq 2$. Suppose $\delta(G) \geq 3k - 2$. Then given any k distinct vertices $v_1, \ldots, v_k \in V(G)$, $V(G)$ can be partitioned as $V(G) = A_1 \cup \cdots \cup A_k$ such that $|A_i| = a_i$, $v_i \in A_i$ and $\delta(G[A_i]) > 0$, for all $1 \leq i \leq k$.

F73: (a) [Th83] For each pair of positive integers (s, t), there exist positive integers $f(s, t)$ and $g(s, t)$ such that each graph G with $\kappa(G) \geq f(s, t)$ (respectively $\delta(G) \geq g(s, t)$) admits a partition of its vertex set $V(G) = S \cup T$ such that the induced subgraphs $G[S]$ and $G[T]$ have connectivity (respectively minimum degree) at least s and t, respectively.

(b) [Ha83] Moreover, if $s \geq 3$ and $t \geq 2$, then $f(s, t) \leq 4s + 4t - 13$.

REMARK

R23: Fact F72 has been used to derive best known error bounds in certain branches of coding theory [CsKö81].

Factor Algorithms and Complexity

DEFINITIONS

D33: Let G be an arbitrary graph. A ***G-factor*** of a graph H is a set $\{G_1, \ldots, G_d\}$ of subgraphs of H such that each subgraph G_i is isomorphic to G and the sets $V(G_i)$ collectively partition $V(G)$.

D34: The ***G-factor recognition problem*** $FACT(G)$:

INSTANCE: A graph H.

QUESTION: Does H admit a G-factor?

D35: The ***clique partition number*** of a graph G is the smallest number $cp(G)$ such that there exists a set of $cp(G)$ cliques in G such that the cliques form a partition of $E(G)$.

D36: A graph G is ***chordal*** if every cycle in G of length greater than 3 has a chord.

D37: The ***H-decomposition Problem:*** Given a fixed graph H, can the edge set of an input graph G be partitioned into copies of H?

D38: If graph G admits a partition of its edge set into t isomorphic subgraphs, then we say that G ***is divisible by*** t. (An obvious necessary condition for G to be divisible by t is that the number of graphs in the partition must divide $|E(G)|$.)

D39: [El88a,El88b] A graph G is ***t-rational*** if G is divisible by t or if $t \not| |E(G)|$.

D40: The ***t-Rational Recognition Problem*** (or the ***Isomorphic Factorization Problem***): Given a graph G and a positive integer t, is G a t-rational graph? Note that t and G form the input to the problem. A candidate for subgraph H is not part of the input.

NOTATION: Throughout this subsubsection, $n = |V(G)|$ and $m = |E(G)|$.

FACTS

F74: The first polynomial algorithm for matching in an arbitrary graph was formulated by Edmonds [Ed65] and has come to be popularly known as the *blossom algorithm*. Its running time is $O(n^4)$.

F75: The fastest algorithm to date for maximum matching in a general (i.e., not necessarily bipartite) graph has complexity $O(m\sqrt{n})$ and is due to Micali and Vazirani [MiVa80]. (See also [PeLo88].) (Curiously, a proof of correctness of this algorithm was not published until fourteen years later! (See [Va94]).) Since the Micali-Vazirani algorithm was introduced, two other matching algorithms [GaTa91, Bl90] having the same complexity as Micali-Vazirani have been produced.

F76: Faster matching algorithms exist, however, in certain special cases. If the graph is 3-regular and has no cutedge, then by the classical result of Petersen [Pe1891], the graph must have a 1-factor. In this case, an $O(n \log^4 n)$ algorithm is given in [BiBoDeLu01] for finding a 1-factor. An $O(n)$ algorithm is also given, if, in addition, the graph is planar.

F77: The Gabow, Kaplan and Tarjan algorithm [GaTaKa99, GaTaKa01] cited above can be modified to test whether a graph has a *unique* f-factor and find it, if it exists, and to check whether a given f-factor is unique, all in polynomial time.

F78: Anstee [An85] gave algorithmic proofs of both the (g, f)-factor theorem and the f-factor theorem and his algorithms either return one of the factors in question or show that none exists, all in $O(n^3)$ time. Note that this complexity bound is independent of the number of edges in the graph and also independent of g and f.

F79: A polynomial algorithm for finding a 2-factor, if one exists, was first found by Edmonds and Johnson [EdJo70]. If one additionally demands that the 2-factor be triangle-free, the problem remains polynomially solvable. (See [CoPu80].) If one demands that the cycle lengths to be disallowed form a non-empty subset of $\{5, 6, \ldots\}$, the problem has been shown to be NP-hard [HeKiKrKr88]. The complexity in the two remaining cases, namely, where only 4-cycles are forbidden or where only triangles and 4-cycles are forbidden, remains unresolved.

F80: The problem of deciding whether or not a graph has a Hamilton cycle is one of first decision problems proved ([Ka72, Ka75]) to be NP-complete. The problem remains NP-complete, even if the graphs are restricted to be 3-regular and planar [GaJoTa76] or 4- or 5-regular and planar [Pi94].

F81: The answer to the factor recognition problem $FACT(K_1)$ is (trivially) always "yes" and so $FACT(K_1) \in P$. Problem $FACT(K_2)$ is just the question of the existence of a perfect matching in H and hence also lies in P. More generally, if G consists of a disjoint union of copies of K_1 and K_2, then $FACT(G)$ belongs to P.

F82: [KiHe83] If any component of G has more than two vertices, then $FACT(G)$ is NP-complete.

F83: [Kaw02c] Let G be a graph of order $4k$ with $\delta(G) \geq 5k/2$. Then G contains a K_4^--factor, where K_4^- denotes the complete graph K_4 with one edge removed.

F84: [ShWaJu88] The problem of determining $cp(G)$ is NP-hard, for the class of K_4-free graphs and for the class of chordal graphs. However, the problem is polynomial for the class of graphs which are both K_4-free and chordal.

F85: [Ho81a] The problem of determining the chromatic index of a graph is NP-complete. If G is bipartite, however, see Fact 86.

F86: **König's Edge-Coloring Theorem** [Kö16a, Kö16b]: If G is bipartite, then $\chi'(G) = \Delta(G)$. The proof yields an $O(mn)$ algorithm to produce an optimal edge-coloring.

F87: Presently, it seems that either an algorithm of Kapoor and Rizzi [KaRi00] or an algorithm of Schrijver [Sc98a] is best for edge-coloring a bipatite graph G, depending upon the relative sizes of $|V(G)|$ and $\Delta(G)$. If the bipartite graphs involved are regular, then even faster algorithms exist. (See [Ri02].)

F88: [Ho81b] Suppose $n \geq 3$. Then the problem of partitioning $E(G)$ into copies of K_n is NP-complete. Holyer used the above result to prove five other edge partition problems to be NP-complete in the same paper.

F89: [BrLo95] If H has no connected component with three or more edges, then the H-decomposition Problem is polynomial.

F90: [DoTa97] The H-decomposition Problem is NP-complete whenever H contains a connected component with three edges or more. (See also [AlCaYu98].)

F91: [BeHo97] There is a polynomial algorithm that finds a factorization of any given 4-regular graph into two triangle-free 2-factors or else shows that such a factorization does not exist.

F92: [Wo84] If $r > 2t$, then almost all labeled r-regular graphs cannot be factorized into $t \geq 2$ isomorphic subgraphs.

F93: But curiously, there is no known example of a regular non-factorizable graph as in Fact F92 which satisfies the obvious necessary divisibility condition: $t\big||E(G)|$.

F94: [ElWo88] Let G be a multigraph and suppose t is an integer such that $t \geq \chi'(G)$. Then G is t-rational.

F95: If G is r-regular and $t \geq r+1$, then G is t-rational. This follows from Vizing's theorem.

F96: [El88a] Let G be a $2k$-regular graph of even order that contains no 3-cycles or 5-cycles. Then $E(G)$ can be partitioned into $2k$ isomorphic subgraphs. Moreover, this factorization can be constructed in polynomial time.

F97: [ScBi78, HaRoWo78] Given the complete graph K_n, then there exists a graph H such that K_n is the edge-disjoint union of t copies of H if and only if $n(n-1) \equiv 0(\bmod 2t)$.

REMARKS

R24: [HaWa77] provides some observations about connections between the Isomorphic Factorization Problem and combinatorial designs. Even the subject of 1-factorizations of graphs (that is, where the isomorphs are 1-factors) is an enormous topic unto itself and quickly leads one into the discipline of combinatorial design theory. See the excellent surveys [StGo81, MeRo85] and the encyclopedic volume [Wa97].

R25: The so-called *packing* problems are closely allied to factor and factorization problems. Here instead of searching for a factor of a particular kind in a given graph G, one seeks a subgraph of G of maximum order which admits the factor. See [LoPo90] for a nice survey of the state of the art.

CONJECTURES

Conjecture [BeHo97]: The problems of recognizing

(a) which $2n$-regular graphs factor into two triangle-free n-factors, and

(b) which $2n$-regular graphs factor into n triangle-free 2-factors

are both NP-complete for all $n \geq 3$.

Conjecture [Hi85]: Let G be a d-regular simple graph of order $2n$ and let $d = p_1 + \cdots + p_r$ be a partition of d. If $d \geq n$, then G has a factorization into edge-disjoint subgraphs $H_1 \cup \cdots \cup H_r$, where H_i is regular of degree p_i. (The author proves the conjecture true in various special cases.)

Subgraph Problems

DEFINITIONS

D41: The ***k*-regular Subgraph Recognition Problem**: given a graph G, does it contain a k-regular subgraph? (Here we do not require that the k-regular subgraph span G.) If $k = 1$ or 2, clearly the problem takes only polynomial time.

D42: ***Berge Conjecture (solved)***: Every 4-regular simple graph G contains a 3-regular subgraph.

FACTS

F98: [Ga83] Let the graph B be k-regular and have order n.

(i) if the graph $\overline{G}$ has a 1-factor, then $v(G) = 0$.

(ii) if the graph $\overline{G}$ has no 1-factor and if n and k are of opposite parity, then $v(G) = 1$.

(iii) if the graph $\overline{G}$ has no 1-factor and n and k are of the same parity, then $n < 2k$ and $v(G) = k + 2$, where $v(G)$ denotes the minimum number of extra vertices needed in order that there exist a $(k + 1)$-regular supergraph of G.

F99: [Ta84] Berge's conjecture is true. (The proof does not provide an algorithm for finding the 3-regular subgraph.)

F100: [ChCo90] The k-regular Subgraph Recognition Problem is NP-complete for all $k \geq 3$.

F101: [EgOt99] If G is a graph with $|V(G)| \geq 4k+6$ and $\delta(G) \geq k+2$, then G contains k pairwise vertex-disjoint claws (i.e., copies of $K_{1,3}$). (The claws are *not* considered to be induced.)

REMARK

R26: There are hundreds of papers in the literature dealing with a wide variety of "subgraph problems" as well as "graph decompositions". The reader is referred to the survey papers [ChGr81, Di90b, Ro90, Be96] and to the books [Bo90, Di90a, CoRo99].

References

[Ab98] S. Abbasi, Spanning subgraphs of dense graphs and a combinatorial problem on strings, Ph.D. Thesis, Rutgers University, 1998.

[Ah84a] R. Aharoni, A generalization of Tutte's 1-factor theorem to countable graphs, *J. Combin. Theory Ser. B* 37 (1984), 199–209.

[Ah84b] R. Aharoni, König's duality theorem for infinite bipartite graphs, *J. London Math. Soc.* 29 (1984), 1–12.

[Ah88] R. Aharoni, Matchings in infinite graphs, *J. Combin. Theory Ser. B* 44 (1988), 87–125.

[Ah91] R. Aharoni, Infinite matching theory, *Discrete Math.* 95 (1991), 5–22.

[AhMaSh92] R. Aharoni, M. Magidor and R. Shore, On the strength of König's duality theorem for infinite bipartite graphs, *J. Combin. Theory Ser. B* 54 (1992), 257–290.

[AhNa84] R. Aharoni and C. Nash-Williams, Marriage in infinite societies, *Progress in Graph Theory*, Academic Press, 1984, 71–79.

[AhNaSh83] R. Aharoni, C. Nash-Williams and S. Shelah, A general criterion for the existence of transversals, *Proc. London Math. Soc.* 47 (1983), 43–68.

[AkExHa80] J. Akiyama, G. Exoo and F. Harary, Covering and packing in graphs III, cyclic and acyclic invariants, *Math. Slovaca* 30 (1980), 405–417.

[AkKan85a] J. Akiyama and M. Kano, Factors and factorizations of graphs – a survey, *J. Graph Theory* 9 (1985), 1–42.

[Al88] N. Alon, The linear arboricity of graphs, *Israel J. Math.* 62 (1988), 311–325.

[AlCaYu98] N. Alon, Y. Caro and R. Yuster, Packing and covering dense graphs, *J. Combin. Des.* 6 (1998), 451–472.

[AmKan82] A. Amahashi and M. Kano, On factors with given components, *Discrete Math.* 42 (1982), 1–6.

[An73] I. Anderson, Sufficient conditions for matching, *Proc. Edinburg Math. Soc.* 18 (1973), 129–136.

[AnEgKaKawMa02] K. Ando, Y. Egawa, A. Kaneko, K. Kawarabayashi, and H. Matsuda, Path factors in claw-free graphs, *Discrete Math.* 243 (2002), 195–200.

[An85] R. Anstee, An algorithmic proof of Tutte's f-factor theorem, *J. Algorithms* 6 (1985), 112–131.

[An90] R. Anstee, Simplified existence theorems for (g,f)-factors, *Discrete Math.* 27 (1990) 29–38.

[An92] R. Anstee, Matching theory: fractional to integral, *New Zealand J. Math.* 21 (1992), 17–32.

[AnNa98] R. Anstee and Y. Nam, More sufficient conditions for a graph to have factors, *Discrete Math.* 184 (1998), 15–24.

[Bä38] F. Bäbler, Über die Zerlegung regulärer Streckencomplexe ungerader Ordnung, *Comment. Math. Helv.* 10 (1938), 275–287.

[Be96] L. Beineke, Graph decompositions, *Congress. Numer.* 115 (1996), 213–226.

[BeLa78] C. Berge and M. Las Vergnas, On the existence of subgraphs with degree constraints, *Nederl. Akad. Wetensch. Indag. Math.* 40 (1978), 165–176.

[BeHo97] E. Bertram and P. Horák, Decomposing 4-regular graphs into triangle-free 2-factors, *SIAM J. Discrete Math.* 10 (1997), 309–317.

BiBoDeLu01] T. Biedl, P. Bose, E. Demaine and A. Lubiw, Efficient algorithms for Petersen's matching theorem, *J. Algorithms* 38 (2001), 110–134.

[Bl90] M. Blum, A new approach to maximum matchings in general graphs, *Languages and Programming* (Proc. of the 17th International Colloquium on Automata) (1990), 586–597.

[Bo85] B. Bollobás, *Random Graphs*, Harcourt Brace Jovanovich (1985).

[Bo01] B. Bollobás, *Random Graphs* (2nd Ed.), Cambridge University Press (2001).

[BoSaWo85] B. Bollobás, A. Saito and N. Wormald, Regular factors of regular graphs, *J. Graph Theory* 9 (1985), 97–103.

[Bo90] J. Bosák, *Decompositions of Graphs*, Kluwer Academic Publishers Group, 1990.

[Br73] L. Brègman, Certain properties of nonnegative matrices and their permanents, *Dokl. Akad. Nauk SSSR* 211 (1973), 27–30. (Russian); *Soviet Math. Dokl.* 14 (1973), 945–949. (English translation)

[Br71] R. Brualdi, Strong transfinite versions of König's duality theorem, *Monatsh. Math.* 75 (1971), 106–110.

[BrLo95] K. Bryś and Z. Lonc, A complete solution of a Holyer problem, *Proc. 4th Twente workshop on Graph and Combinatorial Optimization*, University of Twente, Enschede, The Netherlands, 1995.

[Ca91b] M. Cai, $[a,b]$-factorizations of graphs, *J. Graph Theory* 15 (1991), 283–302.

[ChCo90] F. Cheah and D. Corneil, The complexity of regular subgraph recognition, *Discrete Appl. Math.* 27 (1990), 59–68.

[ChHi89] A. Chetwynd and A. Hilton, 1-factorizing regular graphs with high degree: an improved bound, *Discrete Math.* 75 (1989), 103–112.

[ChGr81] F. Chung and R. Graham, Recent results in graph decompositions, *London Math. Soc. Lecture Note Ser.* 52 (1981), Cambridge University Press, 103–123.

[Ch73] V. Chvátal, Tough graphs and hamiltonian circuits, *Discrete Math.* 5 (1973), 215–228.

[CoRo99] C. Colbourn and A. Rosa, *Triple Systems*, Oxford University Press, 1999.

[CoPu80] G. Cornuéjols and W. Pulleyblank, Perfect triangle-free 2-matchings, *Combinatorial optimization II (Proc. Conf., Univ. East Anglia, Norwich, 1979)*, Math. Programming Stud. 13 (1980), 1–7.

[CoHa63] K. Corrádi and A. Hajnal, On the maximal number of independent circuits in a graph, *Acta Math. Acad. Sci. Hungar.* 14 (1963), 423–439.

[CsKö81] I. Csiszár and J. Körner, Graph decomposition: a new key to coding theorems, *IEEE Trans. Inform. Theory* 27 (1981), 5–12.

[Di90a] R. Diestel, *Graph Decompositions. A Study in Infinite Graph Theory*, Oxford University Press, 1990.

[Di90b] R. Diestel, Decomposing infinite graphs, *Contemporary Methods in Graph Theory*, Bibliographisches Inst., Mannheim, 1990, 261–289.

[DoTa97] D. Dor and M. Tarsi, Graph decomposition is NP-complete: a complete proof of Holyer's conjecture, *SIAM J. Comput.* 26 (1997), 1166–1187.

[Ed65] J. Edmonds, Paths, trees and flowers, *Canad. J. Math.* 17 (1965), 449–467.

[EdJo70] J. Edmonds and E. Johnson, Matching: a well solved class of integer linear programs, *Combinatorial Structures and their Applications*, Gordon and Breach, 1970, 89–92.

[Eg86] Y. Egawa, Era's conjecture on $[k, k+1]$-factorizations of regular graphs, *Ars Combin.* 21 (1986), 217–220.

[EgEn89] Y. Egawa and H. Enomoto, Sufficient conditions for the existence of k-factors, *Recent Studies in Graph Theory*, Vishwa International Publications (1989), 96–105.

[EgKan96] Y. Egawa and M. Kano, Sufficient conditions for graphs to have (g, f)-factors, *Discrete Math.* 151 (1996), 87–90.

[EgOt99] Y. Egawa and K. Ota, Vertex-disjoint claws in graphs, *Discrete Math.* 197/198 (1999), 225–246.

[Eg80] G. Egoryčev, Solution of the van der Waerden problem for permanents, *Sibirsk. Mat. Zh.* 22 (1981), 65–71, 225. (Russian)

[Eg81] G. Egoryčev, The solution of van der Waerden's problem for permanents, *Adv. in Math.* 42 (1981), 299–305.

[El88a] M. Ellingham, Isomorphic factorizations of regular graphs of even degree, *J. Austral. Math. Soc. Ser. A* 44 (1988), 402–420.

[El88b] M. Ellingham, Isomorphic factorization of r-regular graphs into r parts, *Discrete Math.* 69 (1988), 19–34.

[ElWo88] M. Ellingham and N. Wormald, Isomorphic factorization of regular graphs and 3-regular multigraphs, *J. London Math. Soc.* 37 (1988), 14–24.

[El84] M. El-Zahár, On circuits in graphs, *Discrete Math.* 50 (1984), 227–230.

[EnJaKatSa85] H. Enomoto, B. Jackson, P. Katerinis and A. Saito, Toughness and the existence of k-factors, *J. Graph Theory* 9 (1985), 87–95.

[EnMa97b] H. Enomoto and S. Matsunaga, Graph decompositions without isolated vertices III, *J. Graph Theory* 24 (1997), 155–164.

[EnOtKan88] H. Enomoto, K. Ota and M. Kano, A sufficient condition for a bipartite graph to have a k-factor, *J. Graph Theory* 12 (1988), 141–151.

[ErRé66] P. Erdős and A. Rényi, On the existence of a factor of degree one of a connected random graph, *Acta math. Acad. Sci. Hungar.* 17 (1966), 359–368.

[Fa81] D. Falikman, Proof of the van der Waerden conjecture on the permanent of a doubly stochastic matrix, *Mat. Zametki* 29 (1981), 931–938, 957. (Russian) *Math. Notes* 29 (1981), 475–479.

[FaFaFlLiLi99] R. Faudree, O. Favaron, E. Flandrin, H. Li and Z. Liu, On 2-factors in claw-free graphs, *Discrete Math.* 206 (1999), 131–137.

[FrJu99a] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows. I. A unifying framework for design and analysis of matching algorithms, *Networks* 33 (1999), 1–28.

[FrJu99b] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows. II. Simple augmentation algorithms, *Networks* 33 (1999), 29–41.

[FrJu99c] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows. III. Strongly polynomial augmentation algorithms, *Networks* 33 (1999), 43–56.

[FrJu01] C. Fremuth-Paeger and D. Jungnickel, Balanced network flows. IV. Duality and structure theory, *Networks* 37 (2001), 194–201.

[Fr12] G. Frobenius, Über Matrizen aus nicht negativen Elementen, *Sitzungsber. König. Preuss. Akad. Wiss.* 26 (1912), 456–477.

[FuHoMc65] D. Fulkerson, A.Hoffman and M. McAndrew, Some properties of graphs with multiple edges, *Canad. J. Math.* 17 (1965), 166–177.

[GaKaTa99] H. Gabow, H. Kaplan and R. Tarjan, Unique maximum matching algorithms, *Annual ACM Symp. on Theory of Comput.*, ACM New York 1999, 70–78.

[GaKaTa01] H. Gabow, H. Kaplan and R. Tarjan, Unique maximum matching algorithms, *J. Algorithms* 40 (2001), 159–183.

[GaTa91] H. Gabow and R. Tarjan, Faster scaling algorithms for general graph-matching problems, *J. Assoc. Comput. Mach.* 38 (1991, 815–853.

[Ga83] A. Gardiner, Embedding k-regular graphs in $k+1$-regular graphs, *J. London Math. Soc.* 28 (1983), 393–400.

[GaJoTa76] M. Garey, D. Johnson and R. Tarjan, The planar hamiltonian circuit problem is NP-complete, *SIAM J. Comput.* 5 (1976), 704–714.

[Gy78] E. Györi, On division of graphs to connected subgraphs, *Combinatorics (Proc. Fifth Hungarian Colloq., Keszthely, 1976) I*, North-Holland, 1978, 485–494.

[Ha83] A. Hajnal, Partition of graphs with condition on the connectivity and minimum degree, *Combinatorica* 3 (1983), 95–99.

[Ha48] M. Hall, Distinct representatives of subsets, *Bull. Amer. Math. Soc.* 54 (1948), 922–926.

[Ha35] P. Hall, On representatives of subsets, *J. London Math. Soc.* 10 (1935), 26–30.

[HaRoWo78] F. Harary, R. Robinson and N. Wormald, Isomorphic factorizations. I. Complete graphs, *Trans. Amer. Math. Soc.* 242 (1978), 243–260.

[HaWa77] F. Harary and W. Wallis, Isomorphic factorizations. II. Combinatorial designs, *Congress. Numer.* XIX (1977), 13–28.

[HeHeKiLi90] K. Heinrich, P. Hell, D. Kirkpatrick and G. Liu, Communication: A simplified existence criterion for $(g < f)$-factors, *Discrete Math.* 85 (1990), 313–317.

[HeKi81a] P. Hell and D. Kirkpatrick, On generalized matching problems, *Inform. Process. Lett.* 12 (1981), 33–35.

[HeKiKrKr88] P. Hell, D. Kirkpatrick, J. Kratochvíl and I. Kříž, On restricted two-factors, *SIAM J. Discrete Math.* 1 (1988), 472–484.

[He84] G. Hendry, Maximum graphs with a unique k-factor, *J. Combin. Theory Ser. B* 37 (1984), 53–63.

[Hi85] A. Hilton, Factorizations of regular graphs of high degree, *J. Graph Theory* 9 (1985), 193–196.

[Ho81a] I. Holyer, The NP-completeness of edge-coloring, *SIAM J. Comput.* 10 (1981), 718–720.

[Ho81b] I. Holyer, The NP-completeness of some edge partition problems, *SIAM J. Comput.* 10 (1981), 713–717.

[HoPoSt87] M. Holz, K-P. Podewski and K. Steffens, Injective choice functions, Lecture Notes in Mathematics, 1238, Springer-Verlag, 1987.

[IiNi91] T. Iida and T. Nishimura, An Ore-type condition for the existence of k-factors in graphs, *Graphs Combin.* 7 (1991), 353–361.

[IiNi97] T. Iida and T. Nishimura, Neighborhood conditions and k-factors, *Tokyo J. Math.* 20 (1997), 411–418.

[JaWh89] B. Jackson and R. Whitty, A note concerning graphs with unique f-factors, *J. Graph Theory* 13 (1989), 577–580.

[JaLucRu00] S. Janson, T. Łuczak and A. Rucinski, *Random Graphs*, Wiley Interscience (2000).

[Joh00] P. Johann, On the structure of graphs with a unique k-factor, *J. Graph Theory* 35 (2000), 227–243.

[Johs00] R. Johansson, On the bipartite case of El-Zahár's conjecture, *Discrete Math.* 219 (2000), 123–134.

[Kan84] M. Kano, Graph factors with given properties, *Graph Theory, Singapore 1983*, Lecture Notes in Math., 1073 (1984) 161–168.

[Kan85] M. Kano, $[a, b]$-factorization of a graph, *J. Graph Theory* 9 (1985), 129–146.

[Kan90a] M. Kano, A sufficient condition for a graph to have $[a, b]$-factors, *Graphs Combin.* 6 (1990), 245–251.

[Kan90b] M. Kano, Sufficient conditions for a graph to have factors, *Discrete Math.* 80 (1990), 159–165.

[Kan93] M. Kano, Current results and problems on factors of graphs, *Combinatorics, Graph Theory, Algorithms and Applications* (Beijing, 1993), World Scientific Publishers, 1994, 93–98.

[KaKa03] M. Kano and G. Katona, Note: Odd subgraphs and matchings, *Discrete Math.* (2003) (to appear).

[KanSa83] M. Kano and A. Saito, Note: $[a, b]$-factors of graphs, *Discrete Math.* 47 (1983), 113–116.

[KaRi00] A. Kapoor and R. Rizzi, Note: Edge-coloring bipartite graphs, *J. Algorithms* 34 (2000), 390–396.

[Ka82] M. Karoński, A review of random graphs, *J. Graph Theory* 6 (1982), 349–389.

[Ka95] M. Karoński, Random Graphs, *Handbook of Combinatorics*, Elsevier (1995), 351–380.

[Ka72] R. Karp, Reducibility among combinatorial problems, *Complexity of Computer Compututations*, Plenum Press, 1972, 85–103.

[Ka75] R.Karp, On the computational complexity of combinatorial problems, *Networks* 5 (1975), 45–68.

[Kat83] P. Katerinis, Some results on the existence of $2n$-factors in terms of vertex deleted subgraphs, *Ars Combin.* 16-B (1983), 271–277.

[Kat90b] P. Katerinis, Minimum degree of bipartite graphs and the existence of k-factors, *Graphs Combin.* 6 (1990), 253–258.

[KatTs00a] P. Katerinis and N. Tsikopoulos, Minimum degree and F-factors in graphs, *New Zealand J. Math.* 29 (2000), 33–40.

[KatWo87] P. Katerinis and D. Woodall, Binding numbers of graphs and the existence of k-factors, *Quart. J. Math.* 38 (1987), 221–228.

[Kaw02c] K. Kawarabayashi, K_4^--factor in a graph, *J. Graph Theory* 39 (2002), 111–128.

[KiHe83] D. Kirkpatrick and P. Hell, On the complexity of general graph factor problems, *SIAM J. Comput.* 12 (1983), 601–609.

[KlWa73] D. Kleitman and D. Wang, Algorithms for constructing graphs and digraphs with given valences and factors, *Discrete Math.* 6 (1973), 78–88.

[KoSt93] W. Kocay and D. Stone, Balanced network flows, *Bull. Inst. Combin. Appl.* 7 (1993), 17–32.

[Kö16a] D. König, Über Graphen und ihre Andwendung auf Determinantentheorie und Mengenlehre, *Math. Ann.* 77 (1916), 453–465.

[Kö16b] D. König, Graphok és alkalmazásuk a determinánsok és a halmazok elméletére, *Math. Termész. Ért.* 34 (1916), 104–119.

[Kö31] D. König, Graphs and matrices, *Mat. Fiz. Lapok* 38 (1931), 116–119. (Hungarian)

[Kö33] D. König, Über trennende Knotenpunkte in Graphen (nebst. Anwendungen auf Determinanten und Matrizen), *Acta Sci. Math. (Szeged)* 6 (1933), 155–179.

[Ko01] K. Kotani, Factors and connected induced subgraphs, *Graphs Combin.* 17 (2001), 511–515.

[Ko59] A. Kotzig, On the theory of finite graphs with a linear factor II, *Mat.-Fyz. Časopis Slovensk. Akad. Vied* 9 (1959), 136–159.

[Ku73] S. Kundu, The k-factor conjecture is true, *Discrete Math.* 6 (1973), 367–376.

[La75] M. Las Vergnas, A note on matchings in graphs, *Colloque sur la Théorie des Graphes (Paris, 1974)*, Cahiers Centre Étude Rech. Opér., 17 (1975), 257–260.

[La78] M. Las Vergnas, An extension of Tutte's 1-factor theorem, *Discrete Math.* 23 (1978), 241–255.

[LiLi98] G. Li and Z. Liu, On connected factors in $K_{1,3}$-free graphs, *Acta Math. Appl. Sinica* 14 (1998), 43–47.

[LiZhCh02] G. Li, B. Zhu and C. Chen, On connected $[k, k+1]$-factors in claw-free graphs, *Ars Combin.* 62 (2002), 207–219.

[LiWo98] T. Lindquester and N. Wormald, Factorisation of regular graphs into forests of short paths, *Discrete Math.* 186 (1998), 217–226.

[LiGrHo75] C. Little, D. Grant and D. Holton, On defect d-matchings in graphs, *Discrete Math.* 13 (1975), 41–54.

[Li88b] G. Liu, On (g, f)-covered graphs, *Acta Math. Sci. (English ed.)* 8 (1988), 181–184.

[Li89] G. Liu, On $[a, b]$-covered graphs, *J. Combin. Math. Combin. Comput.* 5 (1989), 14–22.

[LoPo90] M. Loebl and S. Poljak, Subgraph packing — a survey, *Topics in Combinatorics and Graph Theory (Oberwolfach, 1990)*, Physica, Heidelberg, 1990, 491–503.

[Lo70] L. Lovász, Subgraphs with prescribed valencies, *J. Combin. Theory* 8 (1970), 391–416.

[Lo74] L. Lovász, Valencies of graphs with 1-factors, *Period. Math. Hungar.* 5 (1974), 149–151.

[Lo77] L. Lovász, A homology theory for spanning trees of a graph, *Acta Math. Acad. Sci. Hungar.* 30 (1977), 241–251.

[LoPl86] L. Lovász and M. Plummer, *Matching Theory*, North-Holland, 1986.

[MeRo85] E. Mendelsohn and A. Rosa, One-factorizations of the complete graphs — a survey, *J. Graph Theory* 9 (1985), 43–65.

[MiVa80] S. Micali and V. Vazirani, An $O(|V|^{1/2}|E|)$ algorithm for finding maximum matchings in general graphs, *Proc. 21st Ann. Sympos. Found. Comput. Sci. (Syracuse)* (1980), 17–27.

[MoRe02] M. Molloy and B. Reed, *Graph Colouring and the Probabilistic Method*, Springer-Verlag (2002).

[Mü79] J. Mühlbacher, F-factors of graphs: a generalized matching problem, *Inform. Process. Lett.* 8 (1979), 207–214.

[Nied91] F. Niedermeyer, f-optimal factors of infinite graphs, *Discrete Math.* 95 (1991), 231–254.

[NiedPo94] F. Niedermeyer and K-P. Podewski, Matchable infinite graphs, *J. Combin. Theory Ser. B* 62 (1994), 213–227.

[Nies98] T. Niessen, Note: A characterization of graphs having all (g, f)-factors, *J. Combin. Theory Ser. B* 72 (1998), 152–156.

[NiesRa98] T. Niessen and B. Randerath, Regular factors of simple regular graphs and factor-spectra, *Discrete Math.* 185 (1998), 89–103.

[NiesVo90] T. Niessen and L. Volkmann, Class 1 conditions depending on the minimum degree and the number of vertices of maximum degree, *J. Graph Theory* 14 (1990), 225–246.

[Nis89a] T. Nishimura, Independence number, connectivity and degree factors, *SUT J. Math.* 25 (1989), 79–87.

[Nis90] T. Nishimura, Note: Regular factors of line graphs, *Discrete Math.* 85 (1990), 215–219.

[Nis91] T. Nishimura, Regular factors of line graphs. II, *Math. Japon.* 36 (1991), 1033–1040.

[Nis92] T. Nishimura, A degree condition for the existence of k-factors, *J. Graph Theory* 16 (1992), 141–151.

[Nis94] T. Nishimura, Degree factors of line graphs, *Ars Combin.* 38 (1994), 149–159.

[Ni78] T. Nishizeki, Lower bounds on the cardinality of the maximum matchings of graphs, *Proc. Ninth Southeastern Conf. on Combinatorics, Graph Theory and Computing* Utilitas Mathematica (1978), 527–547.

[Ni79] T. Nishizeki, On the relationship between the genus and the cardinality of the maximum matchings of a graph, *Discrete Math.* 25 (1979), 149–156.

[Pe1891] J. Petersen, Die Theorie der regulären Graphen, *Acta Math.* 15 (1891), 193–220.

[PeLo88] P. Peterson and M. Loui, The general maximum matching algorithm of Micali and Vazirani, *Algorithmica* 3 (1988), 511–533.

[Pi94] C. Picouleau, Note: Complexity of the hamiltonian cycle in regular graph problem, *Theoret. Comput. Sci.* 131 (1994), 463–473.

[PlTi91] M. Planthold and S. Tipnis, Regular multigraphs of high degree are 1-factorizable, *Proc. London Math. Soc.* 44 (1991), 393–400.

[PlTi01] M. Plantholt and S. Tipnis, All regular multigraphs of even order and high degree are 1-factorable, *Electron. J. Combin.* 8 (2001), #R41.

[Pl94] M. Plummer, Extending matchings in graphs: a survey, *Discrete Math.* 127 (1994), 277–292.

[Pl96] M. Plummer, Extending matchings in graphs: an update, *Utilitas Math.* 116 (1996), 3–32.

[Ra49] R. Rado, Factorization of even graphs, *Quart. J. Math. Oxford Ser.* 20 (1949), 94–104.

[Ri02] R. Rizzi, Finding 1-factors in bipartite regular graphs and edge-coloring bipartite graphs, *SIAM J. Discrete Math.* 15 (2002), 283–288.

[Ro90] C. Rodger, Graph decompositions, *Matematiche (Catania)* 45 (1990), 119–139.

[Sa91] A. Saito, One-factors and k-factors, *Discrete Math.* 91 (1991), 323–326.

[ScBi78] J. Schönheim and A. Bialostocki, Decomposition of K_n into subgraphs of prescribed type, *Arch. Math. (Basel)* 31 (1978/79), 105–112.

[Sc83] A. Schrijver, Bounds on permanents, and the number of 1-factors and 1-factorizations of bipartite graphs, *Surveys in Combinatorics*, London Math. Soc. Lecture Note Ser. 82 (1983), 107–134.

[Sc98a] A. Schrijver, Bipartite edge colouring in $O(\Delta m)$ time, *SIAM J. Comput.* 28 (1998), 841–846.

[Sc98b] A. Schrijver, Counting 1-factors in regular bipartite graphs, *J. Combin. Theory Ser. B* 72 (1998), 122–135.

[ScVa80] A. Schrijver and W. Valiant, On lower bounds for permanents, *Nederl. Akad. Wetensch. indag. Math.* 42 (1980), 425–427.

[ShUp81] E. Shamir and E. Upfal, On factors in random graphs, *Israel J. Math.* 39 (1981), 296–302.

[ShUp82] E. Shamir and E. Upfal, One-factor in random graphs based on vertex choice, *Discrete Math.* 41 (1982), 281–286.

[ShWaJu88] M. Shaohan, W. Wallis and W. Ju-Lin, The complexity of the clique partition number problem, Nineteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing (Baton Rouge, LA, 1988). *Congress. Numer.* 67 (1988), 59–66.

[StGo81] R. Stanton and I. Goulden, Graph factorization, general triple systems, and cyclic triple systems, *Aequationes Math.* 22 (1981), 1–28.

[St77] K. Steffens, Matchings in countable graphs, *Canad. J. Math.* 29 (1977), 165–168.

[St85a] K. Steffens, Maximal tight sets and the Edmonds-Gallai decomposition for matchings, *Combinatorica* 5 (1985), 359–365.

[St85b] K. Steffens, Faktoren in unendlichen Graphen, *Jahresber. Deutsch. Math.-Verein.* 87 (1985), 127–137.

[St89] K. Steffens, The f-factors of countable graphs, *Grüne Reihe Preprint Series* 233 Univ. Hannover (1989).

[St82] F. Steinparz, On the existence of F factors, *Conference on Graphtheoretic Concepts in Computer Science (7th: 1981: Linz, Austria)* Hanser, 1982, 61–73.

[Su74] D. Sumner, On Tutte's factorization theorem, *Graphs and Combinatorics*, Lecture Notes in Math. Vol. 406, Springer (1974), 350–355.

[Su76] D. Sumner, 1-factors and anti-factor sets, *J. London Math. Soc.* 13 (1976), 351–359.

[Ta84] V. Tashkinov, 3-regular subgraphs of 4-regular graphs (Russian), English transl.: *Math. Notes* 6 (1984), 612–623.

[Th81] C. Thomassen, A remark on the factor theorems of Lovász and Tutte, *J. Graph Theory* 5 (1981), 441–442.

[Th83] C. Thomassen, Graph decomposition with constraints on the connectivity and minimum degree, *J. Graph Theory* 7 (1983), 165–167.

[Tu47] W. Tutte, The factorization of linear graphs, *J. London Math. Soc.* 22 (1947), 107–111.

[Tu54] W. Tutte, A short proof of the factor theorem for finite graphs, *Canad. J. Math.* 6 (1954), 347–352.

[Tu78] W. Tutte, The subgraph problem, *Advances in Graph Theory (Cambridge Comb. Conf., Trinity College, Cambridge, 1977)*, Ann. Discr. Math. 3 (1978), 289–295.

[Tu81] W. Tutte, Graph factors, *Combinatorica* 1 (1981), 79–97.

[Va79a] L. Valiant, The complexity of computing the permanent, *Theoret. Comput. Sci.* 8 (1979), 189–201.

[Va79b] L. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* 8 (1979), 410–421.

[Va94] V. Vazirani, A theory of alternating paths and blossoms for proving correctness of the $O(\sqrt{V}E)$ general graph maximum matching algorithm, *Combinatorica* 14 (1994), 71–109.

[Vo95] L. Volkmann, Regular graphs, regular factors, and the impact of Petersen's theorems, *Jahresber. Deutsch. Math.-Verein.* 97 (1995), 19–42.

[Wa26] B. van der Waerden, Problem 45, *Jahresber. Deutsch. Math.-Verein.* 35 (1926), 117.

[Wa97] W. Wallis, *One-Factorizations*, Kluwer Academic Publishers, 1997.

[Wo90] D. Woodall, k-factors and neighbourhoods of independent sets in graphs, *J. London Math. Soc.* 41 (1990), 385–392.

[Wo84] N. Wormald, Isomorphic factorizations. VII. Regular graphs and tournaments, *J. Graph Theory* 8 (1984), 117–122.

[XuLiTo98] B. Xu, Z. Liu and T. Tokuda, Connected factors in $K_{1,n}$-free graphs containing a (g, f)-factor, *Graphs Combin.* 14 (1998), 393–395.

[Ya95] G. Yan, Some new results on (g, f)-factorizations of graphs, *J. Combin. Math. Combin. Comput.* 18 (1995), 177–185.

[YaPaWoTo00] G. Yan, J. Pan, C. Wong and T. Tokuda, Decompositions of graphs into (g, f)-factors, *Graphs Combin.* 16 (2000), 117–126.

[YuKan88] C. Yuting and M. Kano, Some results on odd factors of graphs, *J. Graph Theory* 12 (1988), 327–333.

[ZhZh92] C-Q. Zhang and Y. Zhu, Factorizations of regular graphs, *J. Combin. Theory Ser. B* 56 (1992), 74–89.

5.5 PERFECT GRAPHS

Alan Tucker, SUNY at Stony Brook

5.5.1 Cliques and Independent Sets
5.5.2 Graph Perfection
5.5.3 Motivating Applications
5.5.4 Matrix Representation of Graph Perfection
5.5.5 Efficient Computation of Graph Parameters
5.5.6 Classes of Perfect Graphs
5.5.7 The Strong Perfect Graph Theorem
References

Introduction

The family of perfect graphs includes a number of important classes of graphs such as bipartite graphs and triangulated graphs. Moreover, perfect graphs have special properties that are of great theoretical interest and can be of considerable value in applications. For instance, although it is an NP-complete problem to find the chromatic number of an arbitrary planar graph, the chromatic number of a perfect graph can be determined in polynomial time.

[Go80] is an older text that provides a nice survey of classes of perfect graphs. [BeCh84] reprints key results in the first 25 years of perfect graph research. [AlRe01] surveys more recent developments about perfect graphs.

All graphs discussed in this section are assumed to be simple, that is, they have no self-loops or multi-edges.

5.5.1 Cliques and Independent Sets

The property of graph perfection is associated with two fundamental concepts in graph theory, a clique and an independent set.

DEFINITIONS

D1: A ***clique*** is a subset of vertices in a graph that are mutually adjacent to one another. (Elsewhere, the term “clique” is reserved for maximal subsets of mutually adjacent vertices.)

D2: The ***clique number*** of a graph G, denoted $\omega(G)$, is the size of a largest clique in G.

D3: A ***clique cover*** of a graph G is a collection of cliques that contains every vertex of G.

D4: The ***partition number*** of a graph G, denoted $\rho(G)$, is the size of a smallest clique cover of G.

D5: An ***independent set***, also called a *stable set*, is a subset of vertices in a graph that are mutually non-adjacent.

D6: The ***independence number*** of a graph G, denoted $\alpha(G)$, is the size of a largest independent set in G.

D7: An ***independent-set cover*** of a graph G is a collection of independent sets that contains every vertex of G. The size of a smallest independent-set cover of G is denoted $\chi(G)$.

D8: The ***chromatic number*** of a graph G, denoted $\chi(G)$, is the smallest number of colors needed to partition the vertices of G into color classes. Since color classes (see §5.1) are independent sets, the chromatic number is the same as the size of the smallest independent-set cover.

D9: The ***complement*** (or ***edge-complement***) $G^c = (V, E^c)$ of a simple graph $G = (V, E)$ has the same vertex set V as G and edges defined: (x, y) is in E^c if and only if (x, y) is not in E.

FACTS

F1: Since the independent sets in a graph G correspond bijectively to the cliques in the complement G^c, it follows that

$$\alpha(G) = \omega(G^c) \quad \text{and} \quad \omega(G) = \alpha(G^c) \qquad (*)$$

F2: For any graph G,

$$\chi(G) = \rho(G^c) \quad \text{and} \quad \rho(G) = \chi(G^c) \qquad (**)$$

F3: In a independent-set cover of the vertices of G, each member of a clique is in a different independent set. Similarly, each vertex in a independent set must be in a different clique. Thus,

$$\omega(G) \leq \chi(G) \quad \text{and} \quad \alpha(G) \leq \rho(G) \qquad (***)$$

EXAMPLE

E1: The largest clique in the graph G of Figure 5.5.1 is of size 2, i.e., any edge. The largest clique in the complement G^c is of size 3, e.g., the triangle b, d, f. The largest independent set in G is of size 3, e.g., b, d, f. The largest independent set in G^c is of size 2, i.e., and non-adjacent pair. The minimum independent-set cover (a, c), (b, d), (e, f) of the graph G, corresponds to the minimum clique cover (a, c), (b, d), (e, f) of the complement G^c.

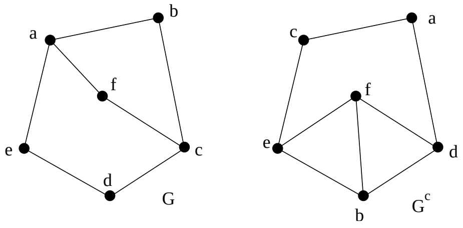


Figure 5.5.1 **Graph G and its complement G^c.**

5.5.2 Graph Perfection

Based on a communication theory problem presented in the next section, Claude Berge [Be61] defined a graph G to be *α-perfect* if $\alpha(G) = \rho(G)$ and also if for every (non-empty) vertex-induced subgraph H of G, $\alpha(H) = \rho(H)$. Based on the complementary relationship between independent sets and cliques captured in (*) and (**) above, Berge simultaneously was studying ω-perfection. See Berge's personal recollections of the history of perfect graphs in [Be97].

Along with conjecturing that α-perfection and ω-perfection are equivalent, Berge also conjectured a structure theorem for perfect graphs called the Strong Perfect Graph Conjecture. There is no obvious way in which these two types of perfection are related, and so perfect graphs are interesting and important because the inequalities in (***) are equalities and because the quantities involved in (*) and (**) – cliques, independent sets, and coverings with them – are basic concepts in graph theory.

DEFINITIONS

D10: A ***vertex-induced subgraph*** $H = (V', E')$ of a graph $G = (V, E)$ has the property that for any two vertices $x, y \in V'$, $(x, y) \in E'$ if and only if $(x, y) \notin E$.

D11: A graph G is ***α-perfect*** if for every (non-empty) vertex-induced subgraph H of G (including $H = G$), $\alpha(H) = \rho(H)$.

D12: A graph G is ***ω-perfect*** if for every (non-empty) vertex-induced subgraph H of G (including $H = G$), $\omega(H) = \chi(H)$.

D13: A graph G is ***perfect*** if it is α-perfect and ω-perfect.

D14: A ***hole*** in a graph is a circuit of odd length ≥ 5 with no chords, i.e., no edges between non-consecutive vertices on the circuit.

D15: An ***antihole*** is the complementary graph of a hole.

D16: A ***Berge graph*** is a graph with no hole or antihole.

FACTS

F4: **Perfect Graph Theorem**: (Lovász [Lo72a]) A graph is α-perfect if and only if its complement is α-perfect.

F5: The Perfect Graph Theorem can be restated: A graph is α-perfect if and only if it is ω-perfect. (Hence, these two types of perfection are equivalent, and it is standard simply to call a graph *perfect* when it is α-perfect and ω-perfect.)

F6: (Lovász) [Lo72b]) A graph is perfect if and only if $\alpha(H)\omega(H) \geq |V(H)|$ for every (non-empty) vertex-induced subgraph H of G.

F7: **The Strong Perfect Graph Theorem**: (Chudnovsky, Robertson, Seymour, Thomas [ChRoSeTh03]) A graph is perfect if and only if it is a Berge graph.

EXAMPLE

E2: Every bipartite graph is perfect. For any bipartite graph $G = (V_1, V_2, E)$, observe that $\chi(G) = \omega(G) = 2$. G is 2-colorable, since all the vertices in V_1 can be given one color and all the vertices in V_2 the second color. Since bipartite graphs contain no triangles, all cliques have size 2.

5.5.3 Motivating Applications

In this subsection, we present two applications related to perfect graphs. The first, in communication theory, was the motivation for Berge's initial exploration that led to the definition of perfect graphs. The second illustrates the roles that graph perfection can play in applied coloring problems. These two examples try to give the reader some sense of the ways that the graph parameters $\alpha(G)$, $\omega(G)$, $\chi(G)$ and $\rho(G)$, and their efficient computation, can arise in diverse applications.

DEFINITIONS

D17: The ***graph*** $G(P)$ ***of a communication channel*** P has a vertex for each different symbol that can be communicated over the channel and an edge between two vertices if they correspond to symbols that can be confused with each other due to noise during transmission over the channel.

D18: The ***direct product*** $G \times H$ of two graphs G, H is defined as follows. For each vertex x of G and vertex y of H, there is a vertex xy in $G \times H$, and there is an edge between xy and $x'y'$ in $G \times H$ if and only if either x is adjacent to x' in G and $y = y'$ or y is adjacent to y' in H and $x = x'$.

D19: The ***n-th power*** G^n of a graph G is the direct product of the graph with itself n times.

D20: (Shannon [Sh56]) A communication channel P is said to be ***perfect*** if for all exponents n, we have $\alpha(G^n(P)) = \alpha(G(P))^n$.

D21: (Shannon [Sh56]) The ***Shannon capacity*** of a communication channel P is defined to be sup $\alpha(G^n(P))^{1/n}$.

EXAMPLE

E3: *Transmission in a Noisy Communication Channel.* To send a single-symbol message through a noisy communications channel P so that there can be no confusion about what message is sent, one chooses a set of symbols corresponding to a set of vertices in $G(P)$ that form a independent set. To maximize the number of non-confusable symbols that can be sent, one would use a maximum independent set of size $\alpha(G(P))$. Note that in this situation, if a symbol is received that is not in this independent set, one knows that an error occurred in transmission and one would ask the transmitter to re-send the message. For example, suppose one has the five symbols $1, 2, 3, 4, 5$ and $G(P)$ is the 5-circuit $1 - 2 - 3 - 4 - 5 - 1$, for which $\alpha(G(P)) = 2$. Any pair of non-consecutive integers forms a maximum independent set, e.g., $1, 3$.

In a multi-symbol message, one assumes that at most one of the symbols might be changed in transmission by noise. The associated graph is now $G^n(P)$, and the maximum number of different non-confusable n-digit messages is $\alpha(G^n(P))$. Observe that for a 2-symbol message using the channel P whose graph was a 5-circuit, one can do better than composing the maximum independent set of $1, 3$ with itself to get four different 2-symbol messages. For example, the vertices $12, 24, 31, 45, 51$ form a larger independent set. The quantity of interest is $\alpha(G^n(P))^{1/n}$, since if one can send $\alpha(G^n(P))$ non-confusable messages of length n, each position in the message contributes a factor of $\alpha(G^n(P))^{1/n}$ to the total of $\alpha(G^n(P))$. Thus, one wants to determine the *Shannon capacity* of a channel, sup $\alpha(G^n(P))^{1/n}$.

REMARKS

R1: Lovász [Lo79] proved that the Shannon capacity of the 5-circuit communication channel above is $\frac{1}{2}log_2(5)$.

R2: The inequality $\alpha(G(P))^n \leq \alpha(G^n(P))$ generalizes the situation for 2-symbol messages in Example 3. If P is a perfect communication channel, then $\alpha(G(P))^n = \alpha(G^n(P))$, and the Shannon capacity is simply $\alpha(G(P))$.

R3: One can show that

$$\alpha(G(P))^n \leq \alpha(G^n(P)) \leq \rho(G^n(P)) \leq \rho(G(P))^n$$

from which it follows that if the graph $G(P)$ is perfect, then $\alpha(G^n(P)) = \alpha(G(P))^n$ and the channel P is perfect.

R4: This noisy channel problem has been generalized to allow a probability distribution governing the likelihood of the different symbols being transmitted. This leads to the concept of graph entropy. See [Si01] for more details about graph entropy.

EXAMPLE

E4: *Application of Graph Coloring in Vehicle Routing.* There is a set of sites S_i that must be serviced (visited) k_i times each week, where $1 \leq k_i \leq 6$. One seeks a minimal set of day-long truck tours for the week such that each site is visited on k_i of the tours. Further, these tours must be partitioned among the days of the week in a manner so that no site is visited twice on one day. This is an extremely difficult problem that can only be solved by approximate, heuristic algorithms. Consider a simplified situation with a 3-day week and suppose the tour-building algorithm has generated the set of 5 tours shown in Figure 5.5.2. Can these five tours be partitioned among the three workdays so that tours visiting a common site are assigned to different days? This is the type of constraint handled by a coloring model.

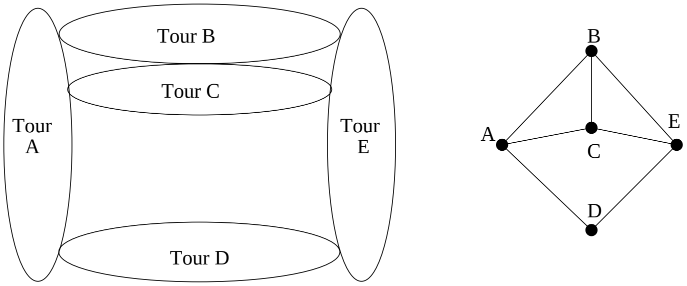


Figure 5.5.2 A set of truck tours and the associated tour graph.

Given a set of tours, one forms an associated tour graph with one vertex for each tour. Two vertices are adjacent if they correspond to two tours that visit a common site. We assign colors, representing the different workdays, to the vertices (tours) of the tour graph. For the tour graph in the figure, the desired 3-coloring does exist. Continual improvements in the solution are sought by combining two tours or other refinements. Determining the chromatic number of the tour graph is often the most time-consuming step in algorithm. The chromatic number of a perfect graph can be determined fairly efficiently, and so one heuristic for graph coloring would be to require that the tour graph be perfect, that is, refinements of tours would be blocked that make the tour graph imperfect. By the now-proved Strong Prefect Graph Conjecture, this means blocking the creation of holes or antiholes.

5.5.4 Matrix Representation of Graph Perfection

Graph perfection can be couched in terms of matrices and polyhedral optimization. This approach to perfect graphs builds on the following two matrices, the clique incidence matrix $\mathbf{A}(G)$ of a graph G and the independent-set incidence matrix $\mathbf{B}(G)$ of G. Without loss of generality, one can assume that only maximal cliques are listed in $\mathbf{A}(G)$, where a clique is maximal if it is not a proper subset of another clique. Similarly, one can assume the independent sets in $\mathbf{B}(G)$ are maximal.

DEFINITIONS

D22: The ***clique incidence matrix*** $\mathbf{A}(G)$ of a graph G is a 0-1 matrix whose rows are incidence vectors of the cliques of G; that is, entry (i, j) in this matrix is 1 if and only if the j-th vertex is in the i-th clique.

D23: A 0-1 matrix $\mathbf{A}$ is called ***perfect*** if for every integer-valued vector $\mathbf{c}$, there exists an integer-value vector $\mathbf{x}$ that achieves the maximum value of the linear program:

$$\text{Max } \mathbf{cx} \text{ subject to } \mathbf{Ax} \leq \mathbf{1}$$

D24: The ***independent-set incidence matrix*** $\mathbf{B}(G)$ of a graph G is a 0-1 matrix whose rows are incidence vectors of the independent sets of G; that is, entry (i, j) in this matrix is 1 if and only if the j-th vertex is in the i-th independent set.

FACTS

F8: From the complementary nature of cliques and independent sets, it follows that $\mathbf{A}(G) = \mathbf{B}(G^c)$ and that $\mathbf{B}(G) = \mathbf{A}(G^c)$.

F9: $\alpha(G)$ is the maximum value of the following integer program:

$$\text{Max } \mathbf{1x} \text{ subject to } \mathbf{A}(G)\mathbf{x} \leq \mathbf{1},\ \mathbf{x} \text{ is a 0-1 vector} \qquad (1)$$

F10: $\rho(G)$ is the minimum value of the following integer program:

$$\text{Min } \mathbf{1y} \text{ subject to } \mathbf{y}^T\mathbf{A}(G) \geq \mathbf{1},\ \mathbf{y} \text{ is a 0-1 vector} \qquad (2)$$

F11: If the constraints that $\mathbf{x}$ and $\mathbf{y}$ be integer valued are dropped and are replaced by the typical constraints of $\mathbf{x} \geq \mathbf{0}$, $\mathbf{y} \geq \mathbf{0}$, then (1) and (2) become a standard pair of dual linear programs:

$$\text{Max } \mathbf{1x} \text{ subject to } \mathbf{A}(G)\mathbf{x} \leq \mathbf{1},\ \mathbf{x} \geq \mathbf{0} \qquad (3)$$

and

$$\text{Min } \mathbf{1y} \text{ subject to } \mathbf{y}^T\mathbf{A}(G) \geq \mathbf{1},\ \mathbf{y} \geq \mathbf{0} \qquad (4)$$

F12: Von Neumann's famous max-min theorem for dual linear programs says that the maximum value in (3) equals the minimum value in (4).

REMARKS

R5: In (1), $\mathbf{x}$ is the incidence vector for a set of vertices in G. Any such $\mathbf{x}$ satisfying $\mathbf{A}(G)\mathbf{x} \leq \mathbf{1}$ is the incidence of an independent set in G.

R6: The objective function $\mathbf{1x}$ in (1) counts the number of vertices in the independent set, and so Max $\mathbf{1x}$ is the size of a maximum independent set.

R7: In (2), $\mathbf{y}$ is the incidence vector of cliques of G. The scalar product of $\mathbf{y}$ with the j-th column of $\mathbf{A}(G)$ is ≥ 1 when at least one clique in the set of cliques represented by $\mathbf{y}$ contains the j-th vertex of G. Then $\mathbf{y}^T\mathbf{A}(G) \geq \mathbf{1}$ says that the set of cliques represented by y must contain collectively all the vertices of G, i.e., the set is a clique cover of G.

R8: One can repeat the preceding discussion using $\mathbf{B}(G)$, the independent-set incidence matrix of G, instead the clique incidence matrix $\mathbf{A}(G)$. Then, $\omega(G)$ is the maximum value of the linear program:

$$\text{Max } \mathbf{1x} \text{ subject to } \mathbf{B}(G)\mathbf{x} \leq \mathbf{1},\ \mathbf{x} \text{ is a 0-1 vector}$$

and $\rho(G)$ is the minimum value of the dual integer program:

$$\text{Min } \mathbf{1y} \text{ subject to } \mathbf{y}^T\mathbf{B}(G) \geq \mathbf{1},\ \mathbf{y} \text{ is a 0-1 vector}$$

FACT

F13: **Chvátal's Theorem** [Ch75]: A graph G is perfect if and only if its clique incidence matrix $\mathbf{A}(G)$ is perfect.

REMARKS

R9: By Chvátal's Theorem, when G is perfect, the linear programs (3) and (4) achieve their optima at integer-valued $\mathbf{x}$ and $\mathbf{y}$. Of course, by the perfection of G, we already knew that $\alpha(G) = \rho(G)$.

R10: Since $\mathbf{B}(G) = \mathbf{A}(G^c)$ and, by the Perfect Graph Theorem, G^c is perfect when G is perfect, it follows that if G is perfect, then $\mathbf{B}(G)$ is a perfect matrix.

5.5.5 Efficient Computation of Graph Parameters

It might seem that for a perfect graph G, $\alpha(G)$ as a solution to the linear program (3) can be determined in polynomial time using the ellipsoid method or Karmarkar's algorithm. Unfortunately, there can be an exponential number of (maximal) cliques in G, and so the number of rows of $\mathbf{A}(G)$ can be an exponential function of the number of vertices of G.

DEFINITION

D25: [Lo79] The ***Lovász parameter*** $\theta(G)$ is formulated as the solution to a linear program. Its definition is very complicated.

FACTS

F14: Linear programs (without integer constraints) can be solved in polynomial time in the number of rows and columns of the constraint matrix.

F15: The independence number $\alpha(G)$ and the partition number $\rho(G)$, the size of a smallest clique cover of a graph G, are NP-hard to compute for arbitrary graphs.

F16: (Grotschel, Lovász, and Schrijver) [GrLoSc81] $\alpha(G) \leq \theta(G) \leq \alpha^*(G)$ (5)

where $\alpha^*(G)$ is the maximum value of the objective function in (3), the linear program relaxation for the integer program (1) for finding the size of a maximum independent set in a graph.

F17: The Lovász parameter $\theta(G)$ can be calculated in polynomial time in the number of vertices in G.

REMARKS

R11: Since $\alpha(G) = \alpha^*(G)$ for perfect graphs, (5) implies that $\alpha(G) = \theta(G)$. Thus, for perfect graphs, $\theta(G)$ can be used to compute $\alpha(G)$ and $\rho(G)$ (since in perfect graphs $\alpha(G) = \rho(G)$), in polynomial time, or looking at the complementary graphs, to compute $\omega(G)$ and $\chi(G)$ in polynomial time.

R12: The motivation for defining $\theta(G)$ arose out of Lovász's study of the Shannon capacity of communications channel (see §5.4.3).

5.5.6 Classes of Perfect Graphs

The best known class of perfect graphs is bipartite graphs. However, there is an immense literature of theorems that are true for bipartite graphs but not for perfect graphs generally. In this section we briefly survey properties of two of the better known other classes of graphs that are perfect.

DEFINITIONS

D26: A graph G is ***bipartite*** if its vertex set V can be partitioned into two sets V_1 and V_2 such that every edge joins a vertex in V_1 with a vertex in V_2.

D27: The ***line graph*** $L(G)$ of a graph G has a vertex for each edge of G; two vertices of $L(G)$ are adjacent if and only if they correspond to two edges of G with a common endvertex.

FACT

F18: The line graph of a perfect graph is a perfect graph.

Interval Graphs

Interval graphs were first studied by Hajos [Ha57]. They have interesting applications in biology and computer science, as indicated in examples below.

DEFINITIONS

D28: The ***intersection graph*** $G(\mathcal{F})$ of a family $\mathcal{F}$ of subsets of a given set has as its vertices the members of F; two vertices are adjacent if and only if the corresponding subsets of F have non-empty intersection.

D29: A graph G is an ***interval graph*** if it is isomorphic to the intersection graph of a family of intervals of a line.

D30: An ***interval model*** for an interval graph G is a family of intervals on the line for which G is an intersection graph.

D31: A graph G is a ***proper interval graph*** if it is an interval graph with the property that there is an interval model $\mathcal{F}$ for G in which no interval of $\mathcal{F}$ is a proper subinterval of another interval of $\mathcal{F}$.

D32: The ***adjacency matrix*** of a graph G is a 0-1 matrix with a row and a column for each vertex and entry (i, j) is 1 if and only if the i-th vertex is adjacent to the j-th vertex.

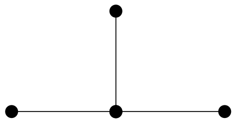
D33: A 0-1 matrix has the ***consecutive 1's property*** if the rows can be rearranged so that the 1's in each column occur consecutively.

FACTS

F19: Interval graphs are perfect graphs.

F20: [BoLu76] Using a data structure called a PQ-tree, one can test whether a graph G is an interval graph in linear time in the number of edges of G.

F21: [Ro69] An interval graph is a proper interval graph if and only if it does not contain $K_{1,3}$, illustrated below, as a vertex-induced subgraph.



F22: [Ro69] A graph G is a proper interval graph if and only if its adjacency matrix has the consecutive 1's property.

F23: [FuGr65] G is an interval graph if and only if its clique incidence matrix (see §5.4.5) has the consecutive 1's property.

EXAMPLES

E5: Consecutive Retrieval Problem. We have to store a collection of items, say, large data sets, on a linear storage device such as a tape drive. We know that subsets S_i of these items will need to be retrieved from time to time. The retrieval time can be minimized if it is possible to arrange the items in an order with the property that each subset S_i of items occurs consecutively in the storage order. If we create a 0-1 matrix $\mathbf{M}$ with a row from each item and a column for each subset and entry $(i, j) = 1$ if and only if item i is in subset S_j, then such a minimizing order exists if and only if $\mathbf{M}$ has the consecutive 1's property.

E6: Genetic Fine Structure [Be59]. Benzer isolated over 100 mutant forms of the virus T4, each known to correspond to different mutation of a connected segment of the DNA in this gene (the nucleotides in this segment of DNA were either missing or were all altered). Benzer wanted to know whether this overlap information of the mutations was consistent with a linear fine structure of genes. That is, could each mutated segment be viewed as an interval along linearly organized DNA. One could represent the overlap information with a graph having a vertex for each mutation and an edge between vertices corresponding to overlapping mutations. Then Benzer's question could be rephrased as, is this mutation graph an interval graph? The overlap data of the 100+ mutations were readily seen to be consistent with an interval model. This was the first persuasive evidence that DNA had a linear structure within individual genes.

REMARKS

R13: Biology background for preceding gene example. In the early stages of studying genes, it was known that genes occur in a locally linear order along a chromosome, i.e., a DNA molecule, but within genes it was unknown whether the DNA subsequence was linear or has a more complicated structure that mirrored the topology of the proteins that a gene's DNA encoded. Benzer performed a series of recombination experiments on a virus called phage T4 that examined the structure of mutations of a particular gene that encoded a protein on the surface of the virus. Different mutant forms T4 had different physical characteristics, e.g., they grew in presence of different nutrients. The recombination experiments determined which mutants corresponded to overlapping mutations of the DNA.

R14: A natural generalization of interval graphs are circular-arc graphs (see Golumbic [Go80] for information about circular-arc graphs), which in general are not perfect.

Chordal Graphs

Chordal graphs were first studied by Hajnal and Surányi [HaSu58] who proved that they were α-perfect. Two interesting subclasses of chordal graphs are comparability graphs and permutation graphs.

DEFINITIONS

D34: A graph is a ***chordal graph*** if every circuit of length ≥ 4 has a chord, i.e., an edge joining non-consecutive vertices on the circuit. Chordal graphs are elsewhere called *triangulated graphs, rigid-circuit graphs, monotone transitive graphs* and *perfect elimination graphs.*

D35: A graph is a ***comparability graph*** if its edges can be directed so that directed adjacency becomes a transitive relation, that is, whenever there exist directed edges (a, b) and (b, c) there must also exist the directed edge (a, c).

D36: A graph G with n vertices is a ***permutation graph*** if there exists a permutation π of the numbers 1 to n such that (x_i, x_j), for $i < j$, is an edge of G if and only if $\pi(j) < \pi(i)$.

D37: A vertex x of graph G is called ***simplicial*** if its neighboring vertices form a clique.

D38: An ***asteroidal triple*** is a set of three vertices in a graph such that there is a path P between any pair of vertices in the triple with no vertex on P adjacent to the third vertex in the triple.

FACTS

F24: Any vertex-induced subgraph of a chordal graph is a chordal graph.

F25: Chordal graphs are perfect.

F26: [GiHo64) G is an interval graph if and only if it is chordal and its complement is a comparability graph. (Thus, interval graphs are a subclass of chordal graphs.)

F27: [LeBo62] G is an interval graph if and only if it is chordal and has no asteroidal triple.

F28: [Di61] The connected graph G is a chordal graph if and only if every (set-theoretically) minimal subset of vertices that disconnects the graph G forms a clique.

F29: [FuGr65] Every chordal graph that is not a complete graph has at least two simplicial vertices.

F30: [FuGr65] A recognition algorithm for chordal graphs is to repeatedly find a simplicial vertex, remove it and repeat the process until the graph is reduced to a single vertex. If at any stage a simplicial vertex cannot be found, then the graph is not chordal. This recognition algorithm can be implemented to run in linear time in the number of edges [RoTaLu76].

F31: [Ga74] The graph G is a chordal graph if and only if it is the intersection graph of subtrees of some tree.

5.5.7 The Strong Perfect Graph Theorem

In the spring of 2002, Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas [ChRoSeTh03] presented a (lengthy) proof of the Strong Perfect Graph Theorem, namely, that a graph is perfect if and only if it is a Berge graph, i.e., it contains no hole or antihole. Note that perfect graphs must be Berge graphs since any vertex-induced subgraph of a perfect graph are perfect but holes and antiholes are not perfect. Thus the challenge was to prove that every Berge graph is perfect. A huge amount of effect over the previous 40 years had been invested in verifying the Strong Perfect Graph Theorem for special families of graphs and by developing machinery for an inductive proof of the theorem, that is, looking at properties of what are called critical imperfect graphs. The successful proof of this theorem took a very different line of reasoning, following a conjecture by Conforti, Cornuejols, and Vuskovic [CoCoVu03] that characterized the structure of Berge graphs.

DEFINITIONS

D39: A graph G is ***critical imperfect*** if G is not perfect, but all proper vertex-induced subgraphs of G are perfect.

D40: A 2-***join*** in a graph $G = (V, E)$ is a partition of V into two sets X_1 and X_2 such that there exist disjoint non-empty $A_i, B_i \subseteq X_i, i = 1, 2$, satisfying the following three conditions.

(i) The pairs (A_1, A_2) and (B_1, B_2) form complete bipartite subgraphs, and there are no other edges between X_1 and X_2.

(ii) For $i = 1, 2$, every component of the subgraph induced by X_i contains at least one member of A_i and of B_i.

(iii) For $i = 1, 2$, if A_i and B_i each consist of a single vertex and the subgraph induced by X_i consists of a path, then that path has length ≥ 3.

D41: An ***M-join*** in a graph $G = (V, E)$ is a partition of V into six nonempty sets, A, B, C, D, E, F satisfying the following three conditions:

(i) Every vertex in A has a neighbor in B and a nonneighbor in B, and vice versa.

(ii) The pairs (A, C), (A, F), (B, D) and (B, F) form complete bipartite subgraphs.

(iii) There are no edges between the following pairs of vertex sets: (A, D), (A, E), (B, C) and (B, E).

D42: A ***skew partition*** in a graph $G = (V, E)$ is a partition of V into two sets X and Y such that the subgraph induced by the vertices of X is disconnected and the complement of the subgraph induced by the vertices of Y is disconnected.

FACTS

F32: Structure Theorem for Berge Graphs [ChRoSeTh03]: A graph G is Berge graph if and only if

(i) it is a bipartite graph, the line graph of a bipartite graph, or the complements of one of these two types of graphs; or

(ii) it has one of the following four vertex partitions: a 2-join of G, a 2-join of its complement G^c, an M-join of G, or a skew partition of G.

F33: Corollary – The Strong Perfect Graph Theorem [ChRoSeTh03]: Every perfect graph is a Berge graph.

REMARK

R15: Chudnovsky [Ch03] subsequently proved that M-joins can be dropped in the above characterization of Berge graphs.

References

[AlRe01] J. L. R. Alfonsin and B. A. Reed, Eds., *Perfect Graphs*, John Wiley and Sons, 2001.

[Be59] S. Benzer, On the topology of the genetic fine structure, *Proc. Natl. Acad. Sci. U.S.A.* 45 (1959), 1607–1620.

[Be61] C. Berge, Färbung von Graphen deren sämtlliche bzw. Deren ungerade Kreise starr sind (Zusammenfassung), *Wiss. Z. Martin-Luther Univ. Halle-Wittenberg, Math.-Natur. Reihe* 10 (1961), 114–115.

[Be97] C. Berge, Motivations and history of some of my conjectures, *Discrete Mathematics* 165/166 (1997), 61–70.

[BeCh84] C. Berge and V. Chvátal, *Topics on Perfect Graphs*, North-Holland, Amsterdam, 1984.

[BoLu76] K. Booth and G. Lueker, Testing for the consecutive ones property, interval graph, and graph planarity using *PQ*-tree algorithms, *J. Comp. Syst. Sci.* 13 (1976), 335–379.

[Ch03] M. Chudnovsky, Berge Trigraphs and Their Applications, Ph.D. Dissertation, Princeton University Mathematics Department, 2003.

[ChRoSeTh03] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, The Strong Perfect Graph Theorem, to appear.

[Ch75] V. Chvátal, On certain polytopes associated with graphs, *J. Combin. Theory Ser. B* 18 (1975), 138–154.

[CoCoVu03] M. Confoti, G. Cornuéjols, and K. Vuskovic, Squaring perfect graphs, *J. Combin. Theory Ser. B*, to appear.

[Di61] G. A. Dirac, on rigid circuit graphs, *Abl. Math. Sem. Univ. Hamburg* 25 (1961), 71–76.

[FuGr65] D. R. Fulkerson and O. Gross, Incidence matrices and interval graphs, *Pacific J. Math.* 15 (1965), 835–855.

[Ga74] F. Gavril, The intersection graphs of substrees in trees are exactly the chordal graphs, *J. Combin. Theory B* 16 (1974), 47–56.

[GiHo64] P. Gilmore and A. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math.* 16 (1964), 539–548.

[Go80] M. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[GrLoSc81] M. Grötschel, L. Lovász, and A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1 (1981), 169–197.

[HaSu58] A. Hajnal and J. Surányi, Über die auflösung von Graphen in vollständige Teilgraphen, *Ann. Univ. Sci. Budapest Eötvös, Sect. Math.* 1 (1958), 113–121.

[LeBo62] C. G. Lekkerkerker and J. C. Boland, Representation of a finite graph by a set of intervals on the real line, *Fund. Math.* 51 (1992), 45–64.

[Lo72a] L. Lovász, Normal hypergraphs and the weak perfect graph conjecture, *Discrete Mathematics* 2 (1972), 253–267.

[Lo72b] L. Lovász, A characterization of perfect graphs, *J. Combin. Theory Ser. B* 13 (1972), 95–98.

[Lo79] L. Lovász, On the Shannon capacity of a graph, *IEEE Trans. Inform. Theory* 25 (1979), 1–7.

[Ro69] F. Roberts, Indifference graphs, pp. 139–146 in *Proof Techniques in Graph Theory*, ed. by F. Harary, Academic Press, 1969.

[RoTaLu76] D. Rose, R. Tarjan, and G. Lueker, Algorithmic aspects of vertex elmination on graphs, *SIAM J. Computing* 5 (1976), 266–283.

[Sh56] C. Shannon, The zero error capacity of a noisy channel, *IRE Trans. Inform. Theory IT-2* (1956), 8–19.

[Si01] G. Simonyi, Perfect graphs and graph entropy, in *Perfect Graphs*, ed. by J. L. R. Alfonsin and B. A. Reed, John Wiley and Sons, 2001.

5.6 APPLICATIONS TO TIMETABLING

Edmund Burke, University of Nottingham, UK
Dominique de Werra, École Polytechnique Fédérale de Lausanne, Switzerland
Jeffrey Kingston, University of Sydney, Australia

5.6.1 Specification of Timetabling Problems
5.6.2 Class-Teacher Timetabling
5.6.3 University Course Timetabling
5.6.4 University Examination Timetabling
5.6.5 Sports Timetabling
References

Introduction

The construction of timetables for educational institutions and other organizations is a rich area of research with strong links to graph theory, especially to node- and edge-coloring, bipartite matching, and network flow problems. A significant amount of recent research has developed powerful hybrids of graph coloring/meta-heuristic methods. The purpose of this section is to demonstrate how graph theory plays a pivotal role in timetabling research today and to provide insight into the close relationship between graph coloring and a range of timetabling problems. We concentrate on four timetabling problems: class-teacher timetabling, university course timetabling, university examination timetabling and sports timetabling, and we illustrate some of the key points that have underpinned graph-theoretical approaches to timetabling over the years. We aim to highlight the role of graph theory in modern timetabling research and to provide some pointers to the relevant literature for the interested reader.

Automated Timetabling: Historical Perspective

The problem of developing computer programs and systems to solve timetabling problems has been addressed by the scientific community for over 40 years. Bardadym in his 1995 survey [Ba96] examines the distribution of educational timetabling publications from 1960 to 1995. This shows a significant growth in educational timetabling research throughout the 1960s and into the 1970s. There is a lowering of interest in the late 1970s, which picks up again in the 1980s and reaches a peak of over 60 published papers in 1995 alone, the year of the 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT) [BuRo96]. In 1996 the European Association of Operational Research Societies Working Group on Automated Timetabling was launched, and today it has over 300 members from more than 60 countries.

REMARKS

R1: The broad definition of the timetabling problem covers a wide variety of important scheduling problems, which include school timetabling, university course timetabling, examination timetabling, sports timetabling, transport timetabling and a wide variety of employee timetabling and rostering problems.

R2: Welsh and Powell [WePo67] observed the relationship between the graph-coloring problem and timetabling in 1967. This relationship has been a significant feature of timetabling research ever since. A broad generation of timetabling algorithms were based upon graph-coloring methods.

R3: It is not our purpose to survey all of these approaches. Carter's 1986 survey paper [Ca86] on examination timetabling provides an excellent review of the early examination timetabling methods, and Carter and Laporte updated this survey paper in 1995 [CaLa96]. There are a number of other timetabling survey papers that cover the field (e.g., [de85-b], [Ba96], [Wr96], [BuJaKiWe97], [CaLa98], and [Sc99]).

TERMINOLOGY: Throughout this section, *node* is used instead of vertex.

5.6.1 Specification of Timetabling Problems

Timetabling problems are complex and vary widely in structure, and methods for their formal specification are under active research [BuKiPe97], [Ki00], [DeVa02]. Our definition is general enough to cover most cases.

The General Problem

DEFINITION

D1: A ***timetabling problem*** is a problem with four parameters: T, a finite set of times; R, a finite set of resources; M, a finite set of meetings; and C, a finite set of constraints. The problem is to assign times and resources to the meetings so as to satisfy the constraints as far as possible. The parts of this definition are elaborated below.

Times

Although it is possible to allow arbitrary time intervals for meetings, in practice time is usually discretized by dividing it into a fixed finite set of intervals of equal length.

DEFINITIONS

D2: A ***time*** t is an element of the set of times T of an instance of the timetabling problem.

D3: A ***time slot*** is a variable constrained to contain one time.

FACTS

F1: Time slots are occasionally preassigned (fixed to a particular value in advance).

F2: In practice, constraints involving time often use information about the actual time intervals being represented. For example, a constraint could specify that two time slots must contain times whose underlying time intervals are directly adjacent, or that a set of time slots must contain times that are spread fairly uniformly through the week, and so on.

F3: Some timetables recur: they are repeated every week, or every two weeks, etc. School and university course timetables recur. Other timetables are used only once (e.g., examination timetables).

Resources

Meetings contain teachers, rooms, items of special equipment, students (or groups of students), and so on, which we call *resources.*

DEFINITIONS

D4: A ***resource*** r is an element of the set of resources R of an instance of the timetabling problem.

D5: A ***resource slot*** is a variable constrained to contain one resource.

FACT

F4: Resource slots are often preassigned (fixed to a particular value in advance). Student group slots are usually preassigned.

EXAMPLE

E1: The basic constraint of timetabling, that no resource appear in two meetings that share a time, applies equally to teachers, students, and rooms, and hence, these items are often treated together as part of a meeting. Other constraints may be specialized for different resources. For example, if a meeting contains several times, it may be required that a particular teacher be present in that meeting for all of those times, whereas in filling the room slot it may be acceptable to use a split assignment, that is, to assign different rooms at different times.

Meetings

DEFINITION

D6: A ***meeting*** m is a named collection of time slots and resource slots. Assigning values to these slots means that all of the assigned resources attend this meeting at all of the assigned times.

EXAMPLES

E2: In examination timetabling, one meeting will usually represent one examination and contain: one time slot, a large number of preassigned students (those students enrolled in the corresponding course), and one or more room slots.

E3: In school timetabling, one meeting will usually represent one subject studied through one week, and will contain some small number of time slots, one preassigned student group slot, one teacher slot (often preassigned) and one room slot.

E4: In staff rostering, one meeting will represent the total staff requirements for one time interval, and will contain one preassigned time and a number of staff slots, not preassigned.

Constraints

Timetabling practitioners have documented dozens of different constraints in the many organizations they have investigated, so it is not possible to give a comprehensive list of constraints in such a general setting. When evaluating constraints against solutions it is convenient to assign a value of 0 to perfectly acceptable outcomes, and to assign progressively higher values to less acceptable outcomes.

DEFINITIONS

D7: Let S be the set of all solutions to a given timetabling problem. A ***hard constraint*** is a constraint that must be satisfied. Associated with each hard constraint is a binary-valued function $h : S \rightarrow \{0, 1\}$, defined for each solution $w \in S$ by

$$h(w) = \begin{cases} 1, & \text{if w does not satisfy the constraint} \\ 0, & \text{otherwise} \end{cases}$$

D8: A ***feasible*** solution is any solution $w \in S$ that satisfies all the hard constraints, i.e., $h(w) = 0$ for all h.

D9: Let S be the set of all solutions to a given timetabling problem. A ***soft constraint*** is a constraint that it is desirable, but not necessary, to satisfy. Associated with each soft constraint is a function $s : S \rightarrow Z^+$. The interpretation is that a solution $w \in S$ for which $s(w)$ is small is preferred.

D10: Let S be the set of all solutions to a given timetabling problem. The ***badness function*** of that problem is a function $b : S \rightarrow Z^+$ that encapsulates in a single number $b(w)$ an overall rating for a solution $w \in S$.

D11: The ***completeness constraint*** requires that every time slot receive a value.

D12: The ***no-clashes constraint*** (or ***no-conflicts constraint***) requires that each resource not participate in any two meetings that share a time.

D13: The ***availability constraint*** specifies that a particular resource is only available for a certain subset of the times T. For example, a part-time teacher might be available only on Thursdays and Fridays.

EXAMPLE

E5: Let the hard constraints for a given problem be $h_1, h_2, \ldots, h_n$ and the soft constraints be $s_1, s_2, \ldots, s_m$. A common approach is to choose a badness function that is a weighted sum of these values:

$$b(S) = \sum_{i=1}^{n} v_i h_i(S) + \sum_{j=1}^{m} w_j s_j(S)$$

where the weights v_i and w_j are nonnegative integers chosen to reflect the importance of the corresponding constraints, with the v_i much larger than the w_j.

REMARKS

R4: In university course timetabling, the no-clashes constraint would typically be a hard constraint for lecturers but a soft constraint for students as far as optional courses are concerned (since it is usually impossible to satisfy every student).

R5: When a resource slot is not preassigned, it almost always carries a resource type constraint, which specifies that the value is constrained to some subset of R. For example, a slot may require one English teacher or one science laboratory. Within the basic categories (rooms, teachers, etc.) these subsets are typically not disjoint; for example, some English teachers may also teach history. Preassignment can be viewed as a type constraint that constrains a slot to a subset of size 1.

R6: The availablity constraint for a particular resource may also be expressed by creating an artificial meeting that contains just that resource and those times when the resource is to be unavailable for actual meetings.

R7: Examples of other constraints often considered are: each teacher is to have at least one hour free each day; each student is to have a lunch hour; large gaps between classes during any one day should be minimized; walking time between classes is to be minimized; etc.

5.6.2 Class-Teacher Timetabling

Class-teacher timetabling is a special case of the general problem in which each meeting contains one preassigned student-group slot, one preassigned teacher slot, and any number of time slots. We first consider this basic version of the problem, and then generalize it to *school problems* (pre-college), in which students are timetabled in groups rather than individually. School problems are characteristically dominated by hard constraints, since constraint violations that might be acceptable when they affect one individual are unacceptable when they affect an entire student group.

The Basic Class-Teacher Timetabling Problem

DEFINITIONS

D14: The ***basic class-teacher timetabling problem*** [Go62] is a timetabling problem in which each meeting contains one preassigned student-group slot, one preassigned teacher slot, and one completely unconstrained time slot. The no-clashes constraint is a hard constraint and applies to every resource.

D15: A ***proper edge-coloring*** in a graph G is a mapping of the edge-set $E(G)$ to a set of colors such that adjacent edges are assigned different colors.

D16: The ***edge-chromatic number*** of a graph G, denoted $\chi'(G)$, is the minimum number of different colors required for a proper edge-coloring of G.

FACTS

F5: There is no requirement that each student group and teacher meet exactly once, or indeed at most once. We could allow each meeting to contain any number k of unconstrained time slots, since that would be equivalent to having k meetings between the given student group and teacher.

F6: The class-teacher timetabling problem can be modeled as an **edge-coloring problem in a bipartite graph** [Be83, de85-a]. Each student group is represented by a left node, each teacher is represented by a right node, and each meeting m is represented by an edge between the nodes corresponding to the student group and teacher preassigned to m. If a student group and teacher meet k times, there will be k parallel edges between the two corresponding nodes. Assigning a time to a meeting corresponds to assigning a color to the corresponding edge; the no-clashes constraint is equivalent to requiring a proper edge-coloring.

F7: An obvious lower bound on the edge-chromatic number of a graph, and hence, on the number of different times needed to timetable an instance of the basic class-teacher problem, is the maximum vertex degree. König's theorem (Fact 8) asserts that for the basic class-teacher problem, this is an upper bound as well.

NOTATION: the maximum vertex degree in a graph G is denoted $\Delta(G)$. Sometimes, when the context is clear, we use Δ.

F8: [König, 1916] Let G be a bipartite graph. Then $\chi'(G) = \Delta$. (See [GrYe99], §10.3, for a proof.)

REMARKS

R8: A timetable using Δ different times can be constructed in low-order polynomial time [Be83]. The algorithm is based on finding maximum matchings in a bipartite graph. Matchings are discussed in §11.3 of the *Handbook*.

R9: The connection between class-teacher timetabling and edge-coloring in a bipartite graph was first made by Csima [Cs65], according to [ScSt80].

Extensions to the Basic Class-Teacher Problem

We give some examples of extensions to the basic class-teacher timetabling problem.

EXAMPLES

E6: Some teachers may be available for only certain subsets of the full set of times. This was the first timetabling problem, identified as such, shown to be NP-complete [EvItSh76]. Allowing some times to be preassigned is essentially the same case, since meetings with preassigned times reduce the availability of the teachers within them.

E7: Multiple time slots within meetings may be constrained to be contiguous. There is an easy reduction from the bin packing problem [GaJo79], where the bins are days, showing that this problem is NP-complete.

E8: Some meetings may be "group meetings" involving several student groups coming together for a large lecture. This problem is NP-complete, but there is a good approximation algorithm [Asde02].

E9: Room slots may be added to the meetings. In most school-timetabling problems, each student group attends some class at every time, and therefore there must be at least as many rooms as there are student groups. In that case, if rooms are not differentiated into different types, each student group can be permanently allocated to some room. If rooms are typed or preassigned, we have an NP-complete problem equivalent to the basic problem with teacher unavailabilities described in Example 6.

Graph Models for Subproblems of the Class-Teacher Problem

It is frequently the case that intractable timetabling problems have tractable subproblems that may be useful to solve within a larger framework. If the subproblem has no solution, then the entire problem is infeasible (and analysis of the model can uncover the deficiency). If the subproblem reveals that there is only one feasible assignment for some slot, then that assignment might as well be made immediately [Go62].

MODELING EXAMPLES

E10: Suppose we need to determine whether a set of meetings can be scheduled to run simultaneously. First we must check that preassignments or other constraints on their time slots do not preclude this. Then we must check that the combined resource slots of all these meetings can be covered by the complete set of resources R. This is trivial if all the resource slots are preassigned (simply check that no resource is used twice), but in general these slots will be constrained to overlapping subsets of R.

Bipartite Graph Model: Each resource slot becomes a left node, each available resource in R becomes a right node, and an edge joins slot s to resource r whenever r is an acceptable resource for slot s. The meetings may run simultaneously if a matching touching every left node exists [CoKi93].

E11: Example 10 generalizes to multiple times in a way that allows us to check whether the resources and times available can cover all the meetings.

Bipartite Graph Model: There is one left node for each possible triple (m, ts, rs), where m is a meeting, ts is a time slot from m, and rs is a resource slot from m. These triples represent indivisible units of demand for one resource at one time. There is one right node for each possible pair (r, t), where r is a resource and t is a time when r is available. These pairs represent indivisible units of supply (of resources). An edge joining a triple to a pair means that the given constraints are not violated by the implied time and resource assignment. For example, if ts is preassigned we would join triples containing it only to pairs containing its preassigned time; if rs requires an English teacher we would join triples containing it only to pairs containing resources r that are teachers whose capabilities include English. Clearly, if there is no matching that touches every triple, then the problem is infeasible.

E12: We may have a partial solution in which some time slots have been assigned times and others have not. We ask whether we can extend this set of time assignments by assigning workable times to all currently unassigned time slots in the set M_r of all meetings containing a particular fixed resource r (e.g., a student group). These time slots must be assigned distinct times, otherwise there will be a clash involving r.

Bipartite Graph Model: The left nodes are the time slots of M_r, and the right nodes are all the times of T. Create an edge between each time slot that is already assigned and the time it has been assigned. For each time slot that has not been assigned, create an edge between it and each of its allowable times. A time is allowable for a time slot if, when the time slot's meeting is added to those meetings that already contain this time, the resulting collection of meetings can run simultaneously. The meetings may be assigned allowable times if there exists a matching in the resulting graph that touches every time slot node [de85-a, CoKi93].

E13: If the times of all meetings are preassigned it may be possible to create models for assigning teachers. For example, suppose that all meetings occupy one time and may be taught by all teachers, but that each teacher is available for a limited set of times and for a limited total number of classes. This problem, which arises in allocating staff to university tutorials, can be modeled as a *network-flow problem*.

Network Flow Model: From the source there is one edge directed to a *teacher-node* for each teacher, with capacity equal to the maximum number of classes for that teacher. From each teacher-node there is one edge with capacity 1 for each time that that teacher is available. Each such edge is directed to a *time-node* that represents the set of all meetings assigned that time. This time-node receives edges from all teachers available at that time. From each time-node, there is an edge directed to the sink, with capacity equal to the number of simultaneous classes allowed at that time. A *minimum-cost network flow model* would allow the inclusion of soft constraints such as teacher preferences for certain times.

REMARK

R10: For a discussion of minimum-cost network flow, see, for example, [Pa82] or §11.2 of the *Handbook*.

5.6.3 University Course Timetabling

University course timetabling differs from the basic class-teacher timetabling problem essentially by the fact that each student may in principle choose the courses of his program, and that there are no other classes of students that are given beforehand and that follow exactly the same program.

Basic Model

The following notation will be used for the rest of this subsection.

NOTATION: Let $\mathcal{C} = \{C_1, \ldots, C_n\}$ denote a collection of courses to be offered during the week W, where W is viewed as a set of time periods. We assume that each course C_i consists of c_i one-period lectures, that is, $C_i = \{C_i^1, C_i^2, \ldots, C_i^{c_i}\}$. For each student s_t, let $\overline{S}_t$ be the collection of courses chosen by student s_t.

DEFINITIONS

D17: A ***course timetable*** is an assignment to each course C_i a set $\overline{C_i} \subset W$ of c_i time periods, one for each of its c_i lectures.

D18: Given a course timetable, a ***conflict*** occurs if for some student s_t, there exist two courses $C_i, C_j \in \overline{S}_t$ such that $\overline{C_i} \cap \overline{C_j} \neq \emptyset$. In other words, there are two courses chosen by student s_t that have at least one lecture at the same time.

D19: The ***university course timetabling problem*** is to produce a conflict-free (or ***feasible***) course timetable.

REMARKS

R11: For the moment we assume that there are no capacity obstacles (i.e., the classrooms are large enough and a course may accommodate any number of students).

R12: It may occur that with a given set of data, no feasible timetable can be found. In such a case we may need to relax our requirements and consider allowing certain conflicting lectures to occur. The resulting timetabling problem becomes one of minimizing the severity of the conflicts. A measure of the severity of a conflict is the number of students who have elected to take both of these lectures. This is formalized in Definition 21 below.

A Graph Formulation

Our graph model consists of nodes representing lectures, and edges joining pairs of these nodes, where the edges are weighted according the severity of the conflicts they represent.

DEFINITIONS

D20: The ***(penalty) weight***, w_{ij}, of a conflict between two lectures C_i^r and C_j^s is the number of students who have to take both of these lectures, i.e.,

$$w_{ij} = |\{t | C_i, C_j \in \overline{S}_t\}|$$

D21: A ***conflict graph*** G is an edge-weighted graph defined as follows: for each course C_i, there are c_i nodes, $C_i^1, C_i^2, \ldots, C_i^{c_i}$, representing its lectures. For each pair i, j, $i \neq j$, if $w_{ij} > 0$, then an edge with weight w_{ij} is created between nodes C_i^r and C_j^s for each pair r, s, $r \neq s$. In addition, an edge with weight ∞ is created between nodes C_i^r, C_i^s for each possible pair r, s, representing a *prohibitive penalty* corresponding to two lectures of the same course.

NOTATION: The edge joining nodes x and y is denoted $[x, y]$. This causes no ambiguity here because conflict graphs have no multi-edges. Some other sections of the *Handbook* use (x, y) or xy to denote simple adjacency between x and y.

D22: In a graph G, a subset of mutually non-adjacent nodes is called a ***stable*** (or ***independent***) set of nodes.

D23: A ***proper node-coloring*** of a graph G is an assignment of colors to the nodes of G such that adjacent nodes receive different colors. A ***proper node k-coloring*** is a proper node-coloring that uses k different colors.

FACTS

F9: The timetabling problem reduces to finding a ***partition*** $\mathcal{P}$ of the node-set $V(G)$ into $k = |W|$ subsets, $S_1, \ldots, S_k$, that minimizes the total penalty

$$z(\mathcal{P}) = \sum_{u=1}^{k} (w_{ij} \mid C_i^r, C_j^s \in S_u)$$

F10: It is easy to see that there is a one-to-one correspondence between feasible (conflict-free) timetables and partitions $\mathcal{P}$ with $z(\mathcal{P}) = 0$: given such a partition, $C_i^r \in S_u$ means that lecture r of course C_i is scheduled at period $u \in W$.

F11: A partition $\mathcal{P}$ for which $z(\mathcal{P}) = 0$ gives rise to a proper node-coloring, obtained by assigning the same color to each node in one cell of the partition so that different cells get different colors. Conversely, given a proper node-coloring, the node-subsets receiving the same color (called *color classes*) form a partition with $z(\mathcal{P}) = 0$. Thus, there exists a feasible timetable in $k = |W|$ periods if and only if G has a proper node k-coloring.

F12: Node-coloring models are more general than edge-coloring models: one can always transform an edge-coloring instance into a node-coloring instance in an auxiliary graph, but the converse is not true.

F13: For some classes of graphs, the determination of the smallest k for which there exists a node k-coloring (the chromatic number) is easy; it is in particular the case for perfect graphs (see [Be83]). But in general the problem is NP-hard. Node-coloring (vertex-coloring) is discussed in detail in §5.1 and §5.2.

EXAMPLE

E14: Figure 5.6.1 gives an example of a university timetabling problem. In the basic model introduced above, we have not mentioned the teachers giving the various courses. We have assumed that all courses are to be taught by different teachers. Should this not be the case, we would simply introduce edges with a prohibitively large weight between lectures (of different courses) that have to be given by the same teacher. This would not change the nature of the problem, which remains a node-coloring problem in a graph or a weighted extension as shown above.

$$\begin{array}{lll}
C_1 = 3 \text{ lectures} & \overline{S}_1 = C_1, C_2 & w_{12} = 1 \\
C_2 = 2 \text{ lectures} & \overline{S}_2 = C_2, C_3 & w_{13} = 1 \\
C_3 = 2 \text{ lectures} & \overline{S}_3 = C_1, C_3, C_4 & w_{14} = 1 \\
C_4 = 1 \text{ lecture} & \overline{S}_4 = C_3, C_4 & w_{23} = 1 \\
 & \overline{S}_5 = C_2 & w_{24} = 0 \\
 & & w_{34} = 2
\end{array}$$

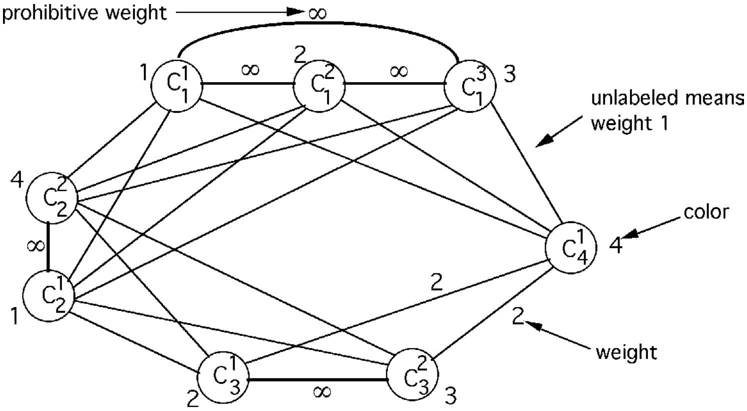


Figure 5.6.1 **An example of university timetabling.**

$$\begin{array}{lcllcl}
C_1^1, C_2^1 & : & \text{period } 1 & C_1^3, C_3^2 & : & \text{period } 3 \\
C_1^2, C_3^1 & : & \text{period } 2 & C_2^2, C_4^1 & : & \text{period } 4
\end{array}$$

$$z(\mathcal{P}) = 1 \quad (edge[C_1^1, C_2^1])$$

Scheduling Multi-Section Courses

Suppose that a collection of m courses, $\{C_1, C_2, \ldots, C_m\}$, has to be scheduled. Assume, for notational simplicity, that each course consists of a single weekly lecture and that there are exactly h_i sections of course C_i, where $h_1 \geq h_2 \geq \ldots \geq h_m$. The following four-step strategy produces a timetable for all sections of all m courses, *in advance*, that can accommodate any collection of student groups, $\{g_1, g_2, \ldots, g_n\}$, as long as no more than h_i of those groups need course C_i. The strategy is followed by an example illustrating each step on a sample problem.

Step 1: Construct bipartite graph $G^* = (L^*, R^*, E^*)$, where the left set $L^* = \{C_1, C_2, \ldots, C_m\}$, the right set $R^* = \{1, 2, \ldots, h_1\}$, and for each $i = 1, 2, \ldots, m$, $[C_i, j] \in E^*$ for $j = 1, 2, \ldots h_i$.

Step 2: Produce a set of *feasible colors* for the sections of each course.

- If Δ be the maximum degree of nodes in G^*, then clearly, $\Delta = \max\{m,\ h_1\}$.
- From König's theorem (Fact 8), G^* has a proper edge Δ-coloring, which can be constructed easily.
- For this edge-coloring of bipartite graph G^*, let $p(C_i)$, $i = 1, 2, \ldots, m$, denote the set of colors used for the edges incident on node (course) C_i. Observe that $|p(C_i)| = h_i$, $i = 1, 2, \ldots, m$.

Step 3: Given an actual collection $\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ of student groups, construct a bipartite graph $G^{**} = (L^{**}, R^{**}, E^{**})$, where the left set $L^{**} = \{C_1, C_2, \ldots, C_m\}$, the right set $R^{**} = \{g_1, g_2, \ldots, g_n\}$, and for each pair i, j, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, edge $[C_i, g_j] \in E^{**}$ if and only if student group g_j needs course C_i.

Step 4: Assign the collection $\mathcal{G}$ of student groups to the sections of courses $C_1, C_2, \ldots, C_m$ without changing the time-period of any section.

NOTATION: Let $g(C_i)$ denote the set of student groups needing course C_i.

FACT

F14: [Hä83] Given the bipartite graph G^{**} defined in Step 3, if $deg_{G^{**}}(C_i) = h_i$ for each i, then there is a proper edge Δ-coloring of G^{**} such that the edges incident on node C_i are assigned the feasible colors of C_i obtained from Step 2. (See also [AsDeHä98].)

REMARK

R13: In terms of the timetabling problem, Fact 14 says that if the number of student groups that need course C_i equals the number of sections that have been scheduled for C_i, i.e., $|g(C_i)| = |p(C_i)|$ $(= h_i)$, $i = 1, 2, \ldots, m$, then there exists an assignment of the student groups to sections such that each student group gets the courses it needs and the original set of time-periods for the sections of each course is unchanged.

EXAMPLE

E15: (Step 1) The graph G^* with $m = 5$ and $(h_1, h_2, \ldots, h_5) = (6, 4, 4, 2, 1)$ is shown in Figure 5.6.2.

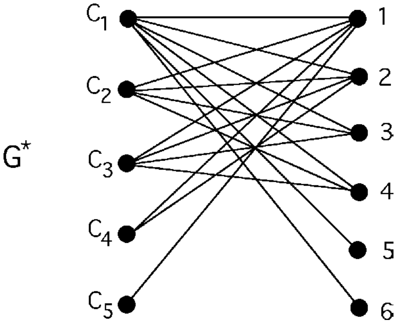


Figure 5.6.2 Bipartite graph G^* for $m = 5$ and $(h_1, h_2, \ldots, h_5) = (6, 4, 4, 2, 1)$.

(Step 2) For the graph in Figure 5.6.2 above, $\Delta = 6$, and a proper edge 6-coloring using colors $\{a, b, c, d, e, f\}$ is represented by the following matrix whose (i, j)th entry is the color assigned to edge $[C_i, j]$.

$$\begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
C_1 & a & b & c & d & e & f \\
C_2 & b & e & d & a & & \\
C_3 & c & d & a & e & & \\
C_4 & d & a & & & & \\
C_5 & e & & & & &
\end{array}$$

Thus, the corresponding timetable for the courses is given by the following sets $p(C_i)$ of feasible colors (time-periods) for the sections of course C_i, $i = 1, 2, \ldots, m$:

$$\begin{aligned}
p(C_1) &= \{a, b, c, d, e, f\} \\
p(C_2) &= \{a, b, d, e\} \\
p(C_3) &= \{a, c, d, e\} \\
p(C_4) &= \{a, d\} \\
p(C_5) &= \{e\}
\end{aligned}$$

(Step 3) The bipartite graph G^{**} shown in Figure 5.6.3 below represents the specific requirements of seven student groups, $g_1, g_2, \ldots, g_7$. For instance, group g_1 needs courses C_1, C_2, and C_4, and group g_6 needs courses C_1 and C_4.

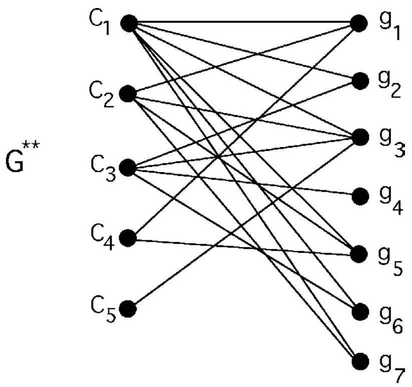


Figure 5.6.3 **Bipartite graph G^{**}.**

(Step 4) A proper edge 6-coloring for bipartite graph G^{**} in Figure 5.6.3 is represented by the matrix below. Observe that the colors used for the edges incident on a given course-node are precisely that node's feasible colors determined in Step 2.

$$\begin{array}{c|ccccccc}
 & g_1 & g_2 & g_3 & g_4 & g_5 & g_6 & g_7 \\
\hline
C_1 & f & e & d & & a & b & c \\
C_2 & d & & b & & e & & a \\
C_3 & & d & c & a & & e & \\
C_4 & a & & & & d & & \\
C_5 & & & e & & & &
\end{array}$$

5.6.4 University Examination Timetabling

Basic Model

Examination timetabling differs from university course timetabling in a number of ways. However, the very core of the problem can be considered to be the same. We have a collection of exams $E_1, \ldots, E_n$ that have to be assigned time slots (periods) and rooms. The number of periods that are available can play a crucial role. In many universities the number of periods extends over a time length of two to four weeks. The constraints that characterize the examination timetabling problem are quite different from constraints that are important in course timetabling.

DEFINITION

D24: Given an examination timetable, a ***conflict*** occurs if two exams taken by the same student are scheduled in the same time period.

FACTS

F15: In examination timetabling it is often desirable (or necessary) to have several exams allocated to the same room. It would, of course, not be very sensible to assign a number of lectures to the same room!

F16: In examination timetabling, it is usually considered desirable to spread exams out over the number of periods so that students do not have exams in succession. On the other hand, for course timetabling it is often considered undesirable to spread the lectures out. Students tend to prefer to have lectures in contiguous blocks. The prototype problem given in Example 16 below has seven exams $(E_1, \ldots, E_7)$ that it has to allocate to five time periods $P_1, \ldots, P_5$. It only attempts to satisfy the constraint that no student can attend more than one examination at the same time. We say that there is a conflict in the timetable if that constraint is not satisfied.

EXAMPLE

E16: For our prototype problem, there are seven exams $(E_1, \ldots, E_7)$ to assign to five time periods $P_1, \ldots, P_5$ such that there are no conflicts. In our graph model, nodes represent examinations, and edges join two nodes whose corresponding exams have at least one student in common. Weights on the edges between two nodes (exams) can represent the number of students who have to take both of those exams. The graph model is shown in Figure 5.6.4 below. Exam 1 only conflicts with Exam 4 (seven students need to take both exams). However, Exam 2 has one, seven, and three students in common with Exam 4, Exam 5, and Exam 7, respectively. This simplified examination-timetabling problem is directly analogous to the node-coloring problem where the colors are represented by the periods and is very similar to the graph-theoretical models discussed earlier. The following solution to this simplified problem uses all five colors (periods).

Period 1	Period 2	Period 3	Period 4	Period 5
Exam-1 Exam-5	Exam-3 Exam-4	Exam-2	Exam-6	Exam-7

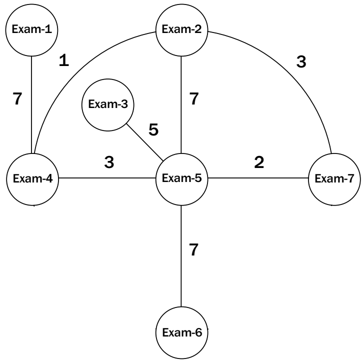


Figure 5.6.4 **A graph model for our prototype problem.**

A More Compact Schedule

Is the solution given in the table above a good solution? If we consider the problem purely as producing a coloring that uses the fewest possible colors, then the answer is clearly no. An alternative coloring can be seen in Figure 5.6.5 below.

Period 1	Period 2	Period 3	Period 4	Period 5
Exam-1 Exam-5	Exam-3 Exam-4 Exam-6 Exam-7	Exam-2		

Figure 5.6.5 **A solution to the problem that uses only three colors.**

REMARK

R14: The solution in Figure 5.6.5 uses only three colors (periods), rather than five, so it is clearly better in terms of the number of colors used. However, if we consider it as an examination timetabling problem, then a student who has to take Exam 1, Exam 2 and Exam 7 would (almost certainly) consider the solution in Figure 5.6.4 to be better. On the other hand, a university administrator might think that the solution in Figure 5.6.5 is better than the one in Figure 5.6.4 because it gets the exams completed more quickly.

FACTS

F17: The node-coloring problem can be considered to be an underlying model of examination timetabling, but for realistic applications there are a number of other constraints, both hard and soft, that need to be considered. For instance, room capacity is a hard constraint, but avoiding having students take consecutive exams is a soft constraint.

F18: In situations where no proper node-coloring exists, the objective might be to minimize the number of students having conflicts. But a further complication is weighing this consideration against other features of an examination timetable: how spread out the exams are, how many days are used, etc.

DEFINITION

D25: The ***quality of a solution*** to an examination timetabling problem can be defined as a measure of the level of satisfaction of the soft constraints (provided all the hard constraints are satisfied).

The Breadth and Variation of Exam Timetabling Constraints

In 1996, Burke, Elliman, Ford, and Weare [BuElFoWe96] published a paper that analyzed and discussed the results of a questionnaire completed by examination-timetabling administrators from 56 British universities. The aim of this exercise was to determine the nature of the problem as it occurred in British universities (circa 1995). The questionnaire concentrated upon 13 constraints but also asked administrators to include other constraints thought to be important to their institution. An additional 19 constraints were listed. The 32 constraints demonstrate the breadth and variation of requirements and priorities among British universities.

REMARK

R15: The message for developers of examination-timetabling decision-support software is that if a system is to be generic and widely applicable, then it has to be flexible in designating which soft constraints are important and which are not. It has to allow the user to weight soft constraints according to the needs and requirements of the user's own institution.

Heuristic Methods

Early approaches to solving the examination-timetabling problem [Br64], [Co64] employed heuristic construction methods. As mentioned in the introduction, the analogy with graph coloring was observed by Welsh and Powell in 1967 [WePo67]. This observation has underpinned the development of examination-timetabling methods. The survey paper [Ca86] and its sequel by Carter and Laporte [CaLa96] provide an excellent overview of the development of graph-coloring-based heuristic approaches for the examination-timetabling problem.

FACTS

F19: One of the basic approaches for solving the examination-timetabling problem is to construct the timetable by sequentially placing exams into periods according to some measure (heuristic) of how difficult the exams are to schedule (see [Ca86]). The early approaches mentioned above used this strategy.

F20: The strategy of scheduling the most troublesome exams first corresponds to coloring the nodes in the graph model that are expected to be the most difficult. Examples of four of the most common node-coloring heuristics used in examination-timetabling systems are presented below.

HEURISTICS

H1: **Largest Degree**: This heuristic takes the nodes with the largest degree (number of edges) and schedules them first. This corresponds to the exams that have the most conflicts with other exams.

H2: **Largest Weighted Degree**: This heuristic is similar to Largest Degree except that the edges are weighted by the number of students who are involved in the conflict.

H3: **Color Degree**: Here, we first schedule the exams that have the largest number of conflicts (degree) with the other exams that have already been placed into the timetable.

H4: **Saturation Degree**: This heuristic chooses first those exams that have the least number of available periods in the timetable that can be selected without violating hard constraints.

REMARKS

R16: While these four heuristics do not form an exhaustive list, they do cover the key node-coloring-based heuristics used in examination-timetabling. Examples of these and other similar approaches include [Br64], [Co64], [WePo67], [Wo68], [Me81], [Me82], and [BuElWe94]. For further discussion, see the survey papers [de85-b], [Ca86], [Ba96], [Wr96], [BuJaKiWe97], [CaLa96], [CaLa98], and [Sc99].

R17: There is an obvious limitation with the simple timetable construction method outlined above (independent of the heuristics used). Exams scheduled early in the process might make certain other exams impossible to schedule later on. This can be addressed by adding a ***backtracking*** component to the process. An algorithm that gets stuck can unschedule or re-schedule exams. Examples of this kind of approach can be found in [CaLaCh94] and [CaLaLe96].

R18: Consistent with the philosophy of scheduling the most troublesome exams first, Carter and his colleagues investigated methods based on finding a ***maximum clique*** of the conflict graph. A maximum clique is a largest subgraph where each node is adjacent to every other node. See [CaJo01] for more details about the role of cliques in examination timetabling.

Two Different Random-Selection Strategies

Burke, Newall, and Weare [BuNeWe98-a] use a random element in the process of selecting the next exam to schedule. This approach produced good results quickly and can be seen as a compromise between the relatively simple coloring-heuristic-based methods discussed earlier and the more complex *meta-heuristic* methods (discussed briefly below), which generally require much more computational time. The two randomization approaches that are considered in [BuNeWe98-a] are described in the next two examples.

(1) A random subset of exams is selected, and the most difficult from within the subset is selected (according to some heuristic).

(2) The x most difficult exams to schedule are selected (according to some heuristic), and then one of those x exams is selected at random.

Hybrid Graph-Coloring/Meta-Heuristic Approaches

Throughout the 1990s, meta-heuristic approaches, such as *simulated annealing*, *evolutionary methods*, and *tabu search*, were investigated and developed for various timetabling problems. Significant progress has been made by combining the more modern meta-heuristic methods with some of the older graph-coloring-based methods. For a discussion of the advantages and disadvantages of using such approaches for timetabling, see [de85-b], [Ca86], [Ba96], [Wr96], [BuJaKiWe97], [CaLa96], [CaLa98], and [Sc99]. To find out more about the meta-heuristics themselves, see [GlKo03].

EXAMPLES

E17: Dowsland and Thompson [ThDo96-a, ThDo96-b] implemented a ***simulated annealing***/graph-coloring hybrid approach for solving the examination-timetabling problem at the University of Wales Swansea. Their method works in two phases. The first phase satisfies the hard (*binding*) constraints:

(a) all exams to be scheduled within 24 time slots,

(b) no student clashes to be allowed,

(c) certain pairs of exams to be scheduled at the same time,

(d) certain pairs of exams to be scheduled at different times,

(e) certain groups of exams to be scheduled in order,

(f) certain exams to be scheduled within time windows,

(g) no more than 1200 students to be involved in any one session.

The second phase of this simulated annealing approach attempts to optimize the soft constraints of the problem:

(a) minimize the number of exams with over 100 students scheduled after period 10,

(b) minimize the number of occurrences of students having exams in consecutive periods.

E18: Burke, Newall and Weare in 1998 [BuNeWe98-b] used graph-coloring heuristics (Largest Degree, Color Degree, and Saturation Degree) to construct initial solutions that were then fine-tuned by ***memetic algorithms***. Memetic algorithms refer to evolutionary methods (often genetic algorithms) combined with local search (often hill-climbing). The memetic algorithm that they investigated was based upon one that had already been shown to work well on benchmark examination-timetabling problems [BuNeWe96].

E19: Burke and Newall used the heuristics outlined above in conjunction with a ***decomposition*** approach [BuNe99]. The authors investigated ways of decomposing large problems into smaller subproblems, which were then solved using memetic algorithms. However, the authors noted that the decomposition approach is independent of the method that is used to solve each of the subproblems. Decomposition had been previously addressed by Carter [Ca83].

E20: Di Gaspero and Schaerf [DiSc01] presented an approach, based on the work of Hertz and de Werra [Hede87], that combined graph-coloring heuristics and ***tabu search***. They employed weights on the edges to represent the number of students who were involved in the conflicts between the corresponding pairs of exams, and they also employed weights on the nodes to indicate the number of students taking the exams. For a range of benchmark problems, their method was competitive with (and in some cases, better than) state-of-the-art methods in 2001.

REMARK

R19: A potential drawback with the decomposition approach described in Example 19 is that exams can be assigned time slots in earlier subproblems that then lead to the infeasibility of later subproblems. Burke and Newall employed graph coloring heuristics to build the subproblems in order to tackle this difficulty. For the problems they considered in [BuNe99], the approach that used the saturation-degree heuristic along with using a subproblem size of 50 exams for the smaller problems and 100 exams for the larger problems was the most effective one. They also employed a look-ahead approach to try and detect difficulties. It considered two subproblems together and fixed the solution to the ith one only after it had solved the $(i + 1)$th one.

5.6.5 Sports Timetabling

This section focuses on modeling and solving some basic problems occurring in the construction of season schedules for sports leagues. We show how the design of some round-robin tournaments can be modeled as an edge-coloring problem in a digraph. Such a model should then be extended to handle more general constraints that arise when a season schedule involves travel that should be optimized. This ***traveling-tournament problem*** (TTP) is described in [EaNeTr03]. Instead of discussing the general problem here, we concentrate on a simple model using elementary properties of graphs. We use the terminology of [Be83] for general graphs and that of [de81] for sports scheduling.

DEFINITIONS

D26: A ***(single) round-robin tournament*** for a set of l teams is a collection of games such that each team plays each other team exactly once. Each game is played in one of the two teams' home city.

D27: If the game between teams i and j is played in the home city of team j, then the game is a ***home game***, H, for team j and an ***away game***, A, for team i.

D28: Given a sports league consisting of l teams, a ***basic sports timetable (schedule)*** (for a round-robin tournament) has two components for each pair of teams i and j:

- designating the day on which the game between i and j is played;
- designating the home city for that game.

REMARK

R20: For the rest of this subsection, we assume that the sports league consists of $2n$ teams for some integer n.

A Simple Graph Model

The league of $2n$ teams are identified with the nodes of a graph G, and an edge joining node i and node j corresponds to a game between team i and team j. Observe that if a round-robin tournament is to be scheduled, graph G is the complete graph K_{2n}.

NOTATION: An undirected edge between node i and node j is denoted $[i, j]$. A directed edge from i to j is denoted (i, j) and indicates that the game is a home game for team j and an away game for team i.

DEFINITIONS

D29: Let G be the $2n$-node graph representing a league of $2n$ teams. An ***oriented d-coloring*** of graph G is a proper edge-d-coloring together with an assignment of a direction to each edge. This oriented d-coloring results in a digraph, each of whose arcs is assigned one of the d colors. This arc-colored digraph specifies a sports timetable using d days for the $2n$ teams: the arcs that are assigned color c_k correspond to those games that are scheduled for day k; and the arc (i, j) indicates that the game between teams i and j is a home game for j and an away game for i.

NOTATION: The digraph created from an oriented coloring of a graph G is denoted $\vec{G}$.

TERMINOLOGY: When a round-robin tournament is to be scheduled (i.e., G is K_{2n}), the digraph $\vec{G}$ is a ***tournament***. This family of digraphs is covered in detail in §3.3.

D30: Let $\{c_1, c_2, \ldots, c_d\}$ be the colors used for an edge d-coloring of a graph G. For each color c_k, $k = 1, 2, \ldots, d$, the ***color class*** M_k is the set of edges assigned color c_k. For a given oriented d-coloring of graph G, $\vec{M}_k$ denotes the set of arcs assigned color c_k.

D31: A ***factor*** of a graph (digraph) G is a subset F of edges (arcs) such that every node of G is incident on exactly one edge (arc) in F.

D32: A ***d-factorization*** of a graph G is a partition, $\{F_1, F_2, \ldots, F_d\}$, of the edge-set of G such that each F_i is a factor of G. A graph G is ***d-factorizable*** if there exists a d-factorization of G. A d-factorization $\{\vec{F}_1, \vec{F}_2, \ldots, \vec{F}_d\}$ of the arcs of a digraph is defined analogously.

TERMINOLOGY NOTE: A factor in an undirected graph is also called a ***perfect matching*** and is actually a 1-factor, where an ***r-factor*** is an r-regular, spanning subgraph of G. Matchings are discussed in §11.3, regular graphs are introduced in §1.2, and graph factors and factorization are discussed in §5.4.

FACTS

F21: A d-factorization $\{F_1, \ldots, F_d\}$ of graph G induces a proper edge d-coloring of G, obtained by assigning color c_k to each of factor F_k, $k = 1, 2, \ldots, d$. Thus, if a graph G is d-factorizable, then there exists a proper edge d-coloring of G.

F22: Analogous to Fact 21, an oriented d-coloring of a graph G induces a d-factorization of the digraph $\vec{G}$.

F23: The d-day schedule that corresponds to a d-factorization of a graph G has the property that each team plays a game on each of the d days, i.e., no team has a day off.

TERMINOLOGY: Sometimes, an oriented d-coloring, its induced d-factorization of the arc-set of the resulting digraph, and the corresponding schedule will all be regarded as the same thing.

EXAMPLE

E21: Figure 5.6.6 shows an oriented 5-coloring (using colors $1, 2, \ldots, 5$) of a complete graph $G = K_6$ representing a league of $2n = 6$ teams.

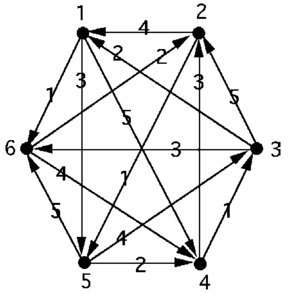


Figure 5.6.6 An oriented 5-coloring of K_6.

The corresponding 5-factorization, $(\vec{F}_1, \ldots, \vec{F}_5)$, of the digraph $\vec{G}$ is shown in table below (Figure 5.6.7). The ith row of the table lists the arcs assigned color i. This factorization specifies the complete 5-day schedule for the six teams. In particular, the arcs in the ith row indicate the games scheduled on the ith day. For instance, on day 2, team 3 plays team 1, and team 1 is at home.

$$\begin{array}{l|ccc|} & \text{game } 1 & \text{game } 2 & \text{game } 3 \\ \vec{F}_1\ (\text{day}1) & (1,6) & (2,5) & (4,3) \\ \vec{F}_2\ (\text{day}2) & (6,2) & (3,1) & (5,4) \\ \vec{F}_3\ (\text{day}3) & (3,6) & (4,2) & (1,5) \\ \vec{F}_4\ (\text{day}4) & (6,4) & (5,3) & (2,1) \\ \vec{F}_5\ (\text{day}5) & (5,6) & (1,4) & (3,2) \end{array}$$

Figure 5.6.7 A compact schedule.

Observe that each team plays a game on each of the five days, which illustrates Fact 23. This kind of *compact* schedule always exists for a round-robin tournament of $2n$ teams because the complete graph K_{2n} has a $(2n-1)$-factorization, where each factor has n edges.

Profiles, Breaks, and Home-Away Patterns of a Schedule

DEFINITIONS

D33: Let S be a schedule for a league of $2n$ teams. The ***home-away pattern*** (HAP) associated with S (see [de81]), denoted $H(S)$, is a $2n \times (2n-1)$ array defined by

$$h_{ik}(S) = \left\{ \begin{array}{c} \text{A} \\ \text{H} \\ \emptyset \end{array} \right\} \text{ if team } i \text{ has } \left\{ \begin{array}{c} \text{an away game} \\ \text{a home game} \\ \text{no game} \end{array} \right\} \text{ on day } k$$

D34: For a given an $(2n-1)$-day schedule S for a league of $2n$ teams, the ***profile*** of team i is the ith row of $H(S)$. Thus, the profile is the sequence of H's and A's indicating when team i is home and away for the $2n-1$ days.

D35: For a given schedule S, the profiles of two teams are ***complementary*** if for each day in the schedule, one of the teams is at home and the other is away.

D36: For a given schedule S, team i has a ***break*** on day $(k+1)$ if $h_{ik}(S) = h_{i,k+1}(S)$. In other words, the profile of team i has two consecutive H's or two consecutive A's, with the second one falling on day $(k+1)$.

EXAMPLE

E22: Figure 5.6.8 shows the HAP associated with the oriented 5-coloring of Figure 5.6.6 and its corresponding schedule in Figure 5.6.7. The breaks are indicated by underlining. Notice that teams 4 and 5 have complementary profiles.

				days		
		1	2	3	4	5
	1	A	H	A	H	A
teams	2	A	H	$\underline{\text{H}}$	A	H
	3	H	A	$\underline{\text{A}}$	H	A
	4	A	H	A	H	$\underline{\text{H}}$
	5	H	A	H	A	$\underline{\text{A}}$
	6	H	A	H	A	H

Figure 5.6.8 **The HAP associated with the schedule of Figure 5.6.7.**

A Lower Bound on the Number of Breaks

Often in round-robin tournaments, one tries to construct schedules in which for each team, home games and away games alternate as regularly as possible (i.e., the number of breaks is minimized).

DEFINITION

D37: A subset of mutually non-adjacent nodes in a graph G is called a ***stable*** (or ***independent***) set. The ***independence number*** of G, denoted $\alpha(G)$, is the maximum size of stable set. Some other sections of the *Handbook* use $ind(G)$ instead of $\alpha(G)$.

FACTS

F24: [de88] Let G be a d-factorizable graph on $2n$ nodes, and let $(\vec{F}_1, \vec{F}_2, \ldots, \vec{F}_d)$ be a d-factorization arising from an oriented d-coloring of G. Then the corresponding schedule has at least $2(n - \alpha(G))$ breaks.

F25: Since the independence number of a complete graph equals 1, any oriented $(2n-1)$-coloring, $(\vec{F}_1, \ldots, \vec{F}_{2n-1})$ of K_{2n} has at least $2n-2$ breaks.

REMARK

R21: Fact 25 implies that the schedule given in Figure 5.6.7 has a minimum number of breaks.

Irreducible and Compact Schedules

DEFINITIONS

D38: A schedule is ***irreducible*** if, whenever two teams play against each other, at most one of them has a break on that day.

D39: A schedule is ***compact*** if each team plays one game on each day (i.e., its HAP has no ϕ symbols).

FACTS

F26: A compact d-day schedule corresponds to a d-factorization of the associated graph.

F27: In a compact schedule, if there is a team with an A in its profile for days k and $k+1$, there must be another team with an H in its profile for days k and $k+1$. Thus, in a compact schedule, *breaks occur in pairs.* (In Figure 5.6.8, teams 2 and 3 and teams 4 and 5 are two such pairs.)

F28: By reversing the orientation of some arcs (corresponding to games with a break for each one of its teams), one may always generate from a schedule S an irreducible schedule that does not have more breaks than S. For the rest of this section, we assume (without loss of generality) that the schedules we consider are irreducible.

NOTATION: Fact 29 below uses the following notation. Given a compact schedule S constructed on a d-regular graph, b_i denotes the ith day on which breaks occur, and γ_i is the number of breaks occurring on day b_i (where $2 \leq b_1 < b_2 < \cdots < b_p \leq d$). In addition, we define $b_0 = 1$ and $b_{p+1} = d+1$.

F29: Let G be a d-regular graph. Then the following conditions are equivalent:

(1) there exists a compact schedule S constructed on G, where $2 \cdot \gamma_i$ breaks occur on day $b_i, i = 1, \ldots, p$.

(2) the edge-set of G can be partitioned into subsets $E_1, \ldots, E_{p+1}$ such that

 (a) The edge subset E_i induces a $(b_i - b_{i-1})$-regular bipartite graph with vertex bipartition $\{X_i, \overline{X_i}\}$ for $i = 1, \ldots, p+1$

 (b) $|\overline{X}_{i+1} \cap X_i| = |\overline{X}_i \cap X_{i+1}| = \gamma_i$ for $i = 1, \ldots, p$.

EXAMPLE

E23: For the compact schedule S in Figure 5.6.7 (and its corresponding HAP in Figure 5.6.8), $(b_0, b_1, b_2, b_3) = (1, 3, 5, 6)$ and $\gamma_1 = \gamma_2 = 1$, and it is easy to see that condition (1) of Fact 29 is satisfied.

To show that condition (2) is satisfied, let $E_1 = F_1 \cup F_2$; $E_2 = F_3 \cup F_4$; and $E_3 = F_5$. Then the vertex bipartitions of the induced subgraphs are:

$$X_1 = \{1,2,4\},\ \overline{X_1} = \{3,5,6\}$$
$$X_2 = \{1,3,4\},\ \overline{X_2} = \{2,5,6\}$$
$$X_3 = \{1,3,5\},\ \overline{X_3} = \{2,4,6\}$$

It is now straightforward to verify that condition (2) is also satisfied.

Complementarity

Another property of compact schedules that is of interest in practice is *complementarity.*

DEFINITION

D40: A compact schedule S for K_{2n} has the ***complementarity property*** if the $2n$ teams can be grouped into n disjoint pairs $T_1, \ldots, T_n$ such that the two teams in each T_i have complementary profiles.

FACTS

F30: [de88] If S is a compact schedule (for K_{2n}) such that each team has at most one break, then S has the complementarity property (by Fact 27).

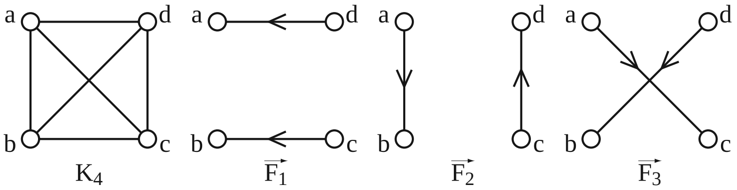
F31: If S is a compact schedule (for K_{2n}) with exactly $2n-2$ breaks, then S has the complementarity property.

F32: There are compact schedules with the complementarity property where some teams have more than one break.

EXAMPLES

E24: Consider the compact schedule S in Figure 5.6.7 and its corresponding HAP, given in Figure 5.6.8. The three pairs $T_1 = \{1,6\}$, $T_2 = \{2,3\}$, and $T_3 = \{4,5\}$ show that S has the complementarity property.

E25: Figure 5.6.9 shows a 3-factorization of $G = K_4$ that corresponds to an irreducible compact schedule S. Its HAP shows that S does not have the complementarity property (team a has two breaks).



$$\begin{array}{r} \\ \text{team a} \\ \text{team b} \\ \text{team c} \\ \text{team d} \end{array} \begin{array}{c} \begin{array}{ccc} \text{day 1} & \text{day 2} & \text{day 3} \end{array} \\ \left(\begin{array}{ccc} H & A & \underline{A} \\ H & \underline{H} & \underline{H} \\ A & \underline{A} & H \\ A & H & A \end{array} \right) \end{array}$$

Figure 5.6.9 **An irreducible compact schedule of K_4.**

Constructing a Compact Schedule with a Minimum Number of Breaks

We restrict our attention to the most common case, when $G = K_{2n}$. Algorithm 5.6.1 below gives a simple construction that produces an oriented coloring (and hence, a schedule) having exactly $2n-2$ breaks, which, by Fact 25, is the minimum.

EXAMPLE

E26: Figure 5.6.10 illustrates the Algorithm 5.6.1 for K_6. Observe that the schedule reproduces the oriented 5-coloring given in Figure 5.6.6.

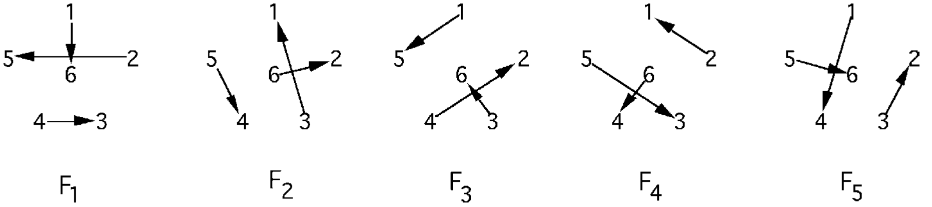


Figure 5.6.10 **The 5-day schedule of K_6 produced by Algorithm 5.6.1.**

Algorithm 5.6.1: A $(2n-1)$-Day Schedule of K_{2n} With $2n-2$ Breaks

```
Input: Complete graph $K_{2n}$.
Output: $(2n-1)$-day schedule with $2n-2$ breaks.
    Step 1. Construct a $(2n-1)$-factorization of $K_{2n}$:
        For $i = 1$ to $2n-1$
            $F_i = \{[2n, i]\} \cup \{[i+k, i-k] \pmod{2n-1} : k = 1, 2, \ldots, n-1\}$
    Step 2. Orient the edges:
        For $i = 1$ to $2n-1$
            If $i$ is odd
                orient edge $[2n, i]$ as $(i, 2n)$
            Else
                orient edge $[2n, i]$ as $(2n, i)$
            For $k = 1$ to $n-1$
                If $k$ is odd
                    orient edge $[i+k, i-k]$ as $(i+k, i-k)$
                Else
                    orient edge $[i+k, i-k]$ as $(i-k, i+k)$
```

REMARK

R22: The factorization specified in Step 1 of the algorithm is discussed in [Be83, Chapter 5]. It is called a ***canonical*** factorization [de88].

An Alternate View of the Canonical Factorization

Let $\alpha_n, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \ldots, \alpha_{n-1}, \beta_{n-1}, \beta_n$ denote the nodes $1, 2, \ldots, 2n$, respectively, and let $(F_1, \ldots, F_{2n-1})$ be the canonical factorization produced by Algorithm 5.6.1. Consider the partition $(E_1, E_2, \ldots, E_n)$ of the edge-set of K_{2n}, defined by $E_n = F_{2n-1}$ and $E_i = F_{2i-1} \cup F_{2i}$, $i = 1, \ldots, n-1$.

FACTS

F33: For $i = 1, 2, \ldots, n-1$, E_i defines a 2-regular bipartite graph on node sets $\mathrm{X}_i = \{\alpha_i, \alpha_{i+1}, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_{i-1}\}$ and $\overline{X}_i = \{\beta_i, \beta_{i+1}, \ldots, \beta_n, \alpha_1, \alpha_2, \ldots, \alpha_{i-1}\}$. In addition, $X_i \cap \overline{X}_{i+1} = \alpha_i$ and $\overline{X}_i \cap X_{i+1} = \beta_i$. Thus, by Fact 29, it defines a schedule where nodes α_i and β_i have a simultaneous break on day $b_i = 2i + 1$, $i = 1, \ldots, n-1$.

F34: Fact 33 implies that the canonical factorization produces a compact schedule having exactly $2n-2$ breaks and satisfying the complementarity property.

EXAMPLE

E27: For K_6, we have $\alpha_1 = 2, \alpha_2 = 4, \alpha_3 = 1, \beta_1 = 3, \beta_2 = 5, \beta_3 = 6$, and from the HAP of Figure 5.6.8, one sees that teams $\alpha_1 = 2$ and $\beta_1 = 3$ have a break on day 3, while teams $\alpha_2 = 4$ and $\beta_2 = 5$ have a break on day 5, while teams $\alpha_3 = 1$ and $\beta_3 = 6$ have no break.

Some Characterization Results

FACTS

F35: Let S_1 and S_2 be two compact schedules for K_{2n}, each with exactly $2n-2$ breaks. If both schedules have the same sequence $b_1, b_2, \ldots, b_{n-1}$ of days where breaks occur in pairs, then their HAPs, $H(S_1)$ and $H(S_2)$, are the same (up to a permutation of rows).

F36: Equivalently, by setting $b_0 = 1$ and $b_n = 2n$, we could start with the sequence $D = (b_1 - b_0, b_2 - b_1, \ldots, b_n - b_{n-1})$, which is the sequence of degrees of the $(b_i - b_{i-1})$-regular bipartite graphs appearing in the partition of the edge-set of K_{2n} defined in Fact 33. (For instance, the schedule of Figure 5.6.7 has $D = (2,2,1)$ since $(b_0, b_1, b_2, b_3) = (1,3,5,6)$.)

F37: Given a sequence $D = (d_1, d_2, \ldots, d_n)$ with $d_1 + \cdots + d_n = 2n-1$, we can reconstruct a unique HAP as follows: for $i \leq n-1$ the profile of α_i starts with an A and has a unique break on day $d_1 + \cdots + d_i + 1$; the profile of α_n starts also with an A and has no break. For each $i \leq n$ the profile of β_i is the complement of the profile of α_i.

EXAMPLE

E28: The HAP in Figure 5.6.11 illustrates Fact 37 for $2n = 6$ and $D = (3,1,1)$.

$$\begin{array}{cc} & \begin{array}{ccccc} \text{day 1} & \text{day 2} & \text{day 3} & \text{day 4} & \text{day 5} \end{array} \\ \begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{array} & \begin{pmatrix} H & A & H & \underline{H} & A \\ H & A & H & A & \underline{A} \\ H & A & H & A & H \\ A & H & A & \underline{A} & H \\ A & H & A & H & \underline{H} \\ A & H & A & H & A \end{pmatrix} \end{array}$$

Figure 5.6.11 **A HAP corresponding to $D = (3,1,1)$.**

REMARK

R23: A sequence $D = (d_1, \ldots, d_n)$ of positive integers with $d_1 + \cdots + d_n = 2n-1$ does not in general give a HAP that corresponds to a compact schedule for K_{2n} with $2n-2$ breaks. For instance, the HAP in Figure 5.6.11 does not correspond to any compact schedule for K_6.

Feasible Sequences

DEFINITION

D41: A sequence $D = (d_1, \ldots, d_n)$ and its corresponding HAP are ***feasible*** if they correspond to a compact schedule for K_{2n} with $2n-2$ breaks.

FACTS

F38: [de88] If $D = (d_1, d_2, \ldots, d_n)$ is feasible for K_{2n}, then $\overline{D} = (d_n, d_{n-1}, \ldots, d_1)$ and all sequences obtained by a cyclic permutation of D or $\overline{D}$ are also feasible.

F39: No complete characterization of the feasible sequences has been obtained yet; however, for $n \leq 13$ the feasible sequences have been characterized (see [MiIwMa02]).

F40: Given a sequence $(d_1, d_2, \ldots, d_n)$, we can reconstruct the associated HAP such that: the rows are ordered $\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \beta_2, \ldots, \beta_n$, where α_i and β_i have complementary profiles; α_i and β_i have their break on day $d_1 + \cdots + d_i + 1$ $(i = 1, \ldots, n-1)$; and the profiles of $\alpha_1, \ldots, \alpha_n$ start with an A.

NOTATION: For a given HAP and any subset T of teams, we define the quantity $\alpha_k(T)$ for each day k by $\alpha_k(T) = \min\{|\{i \in T | h_{ik} = A\}|,\ |\{i \in T | h_{ik} = H\}|\}$.

F41: [MiIwMa02] If a given HAP is feasible, then for any subset T of teams, $\alpha_k(T)$ is an upper bound on the number of games between teams in T that can be scheduled at period k. Moreover, since all these teams have to play against each other over the $2n-1$ days, we have

$$\sum_{k=1}^{2n-1} \alpha_k(T) \geq \binom{|T|}{2}$$

F42: [MiIwMa02] Instead of checking explicitly all possible subsets T, it is sufficient to assume that the $2n$ teams are cyclically ordered $(\alpha_1, \alpha_2, \ldots, \alpha_n, \beta_1, \ldots, \beta_n, \alpha_1, \ldots)$, and to examine only subsets T that are intervals of at most n consecutive teams in the cyclic order. Thus, the number of inequalities to check is $O(n^2)$. Using these observations, the authors were able to eliminate, as infeasible, a number of sequences $D = (d_1, \ldots, d_n)$; it turned out that for $2n \leq 26$, all sequences D that were not eliminated did correspond to feasible HAPs.

CONJECTURE

[MiIwMa02] The inequalities given in Fact 41 are necessary and sufficient conditions for a sequence D to correspond to a feasible HAP.

REMARKS

R24: There are also season schedules where each pair of teams has to meet several times. The schedule consists of rounds that have to satisfy additional requirements.

R25: Also, there are often constraints that require more breaks in the schedule (for instance, some teams might have away games or home games on prespecified days), so we cannot use all the properties of schedules with a minimum number of breaks.

R26: Some references on various types of sports-scheduling problems are given in [EaNeTr03].

R27: Finally, one should observe that canonical factorizations are not the only factorizations that should be considered for constructing the schedules. There are other types of factorization that are of interest (in particular when a league is divided into several subleagues in which internal games have also to be played). Such constraints are considered in [de82] and [de85-c].

References

[AsDeHä98] A. S. Asratian, T. M. J. Denley, and R. Häggkvist, *Bipartite Graphs and their Applications*, Cambridge University Press, 1998.

[Asde02] Asratian and D. de Werra, A Generalized Class-Teacher Model for Some Timetabling Problems, *European J. of Operational Research* 143 (2002), 531–542.

[Ba96] V. A. Bardadym, Computer Aided School and University Timetabling: The New Wave, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*. Springer, (1996), 22–45.

[Be83] C. Berge, *Graphes*, Gauthiers-Villars, Paris, 1983.

[Br64] S. Broder, Final Examination Scheduling, *Communications of the ACM* 7 (1964), 494–498.

[BuCa98] E. Burke and M. Carter (Eds.), *Practice and Theory of Automated Timetabling II (PATAT 1997, Toronto, Canada, August, selected papers) (Lecture Notes in Computer Science, Vol. 1408)*, Springer, 1998.

[BuDe03] E. Burke and P. De Causmaecker (Eds.), *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent, Belgium, August, selected papers) (Lecture Notes in Computer Science, Vol. 2740)*, Springer, 2003.

[BuElWe94] E. K. Burke, D. G. Elliman, and R. F. Weare, A University Timetabling System Based on Graph Colouring and Constraint Manipulation, *J. of Research on Computing in Education* 27 (1994), 1–18.

[BuElFoWe96] E. K. Burke, D. G. Elliman, P. H. Ford and R. F. Weare, Examination Timetabling in British Universities: a Survey, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*, Springer, (1996), 76–90.

[BuEr01] E. Burke and W. Erben (Eds.), *Practice and Theory of Automated Timetabling III (PATAT 2000, Konstanz, Germany, August, selected papers) (Lecture Notes in Computer Science, Vol. 2079)*, Springer, 2001.

[BuJaKiWe97] E. K. Burke, K. S. Jackson, J. H. Kingston, and R. F. Weare, Automated Timetabling: The State of the Art, *The Computer J.* 40 (1997), 565–571.

[BuKiPe97] E. K. Burke, J. H. Kingston and P. W. Pepper, A Standard Data Format for Timetabling Instances, in E. Burke and M. Carter (Eds.), *Practice and Theory of Automated Timetabling II (PATAT 1997, Toronto, Canada, August, selected papers) (Lecture Notes in Computer Science, Vol. 1408)*, Springer, 1998.

[BuNe99] E. K. Burke and J. P. Newall, A Multi-stage Evolutionary Algorithm for the Timetable Problem, *IEEE Transactions on Evolutionary Computation* 3 (1999), 63–74.

[BuNeWe96] E.K.Burke, J.P.Newall, and R.F.Weare, A Memetic Algorithm for University Exam Timetabling, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*. Springer, (1996), 241–250.

[BuNeWe98-a] E. K. Burke, J. P. Newall, and R. F. Weare, A Simple Heuristically Guided Search for the Timetable Problem, *Proceedings of the International ICSC Symposium on Engineering of Intelligent Systems (EIS'98)*, ICSC-Academic, New York (1998), 575–579.

[BuNeWe98-b] E. K. Burke, J. P. Newall, and R. F. Weare, Initialization Strategies and Diversity in Evolutionary Timetabling, *Evolutionary Computation* 6 (1996), 81–103.

[BuRo96] E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*, Springer, 1996.

[Ca83] M. W. Carter, A Decomposition Algorithm for Practical Timetabling Problems, *Working Paper 83-06*, Department of Industrial Engineering, Univ. of Toronto (1983).

[Ca86] M. W. Carter, A Survey of Practical Applications of Examination Timetabling Algorithms, *Operations Research* 34 (1986), 193–201.

[CaJo01] M. W. Carter and D. G. Johnson, Extended Clique Initialization in Examination Timetabling, *J. of the Operations Research Society* 52, No. 5 (2001), 538–544.

[CaLa96] M. W. Carter and G. Laporte, Recent Developments in Practical Examination Timetabling, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*, Springer, (1996), 3–21.

[CaLa98] M. W. Carter and G. Laporte, Recent Developments in Practical Course Timetabling, in E. Burke and M. Carter (Eds.), *Practice and Theory of Automated Timetabling II (PATAT 1997, Toronto, Canada, August, selected papers) (Lecture Notes in Computer Science, Vol. 1408)*, Springer, (1998), 3–19.

[CaLaCh94] M. W. Carter, G. Laporte, and J. W. Chinneck, A General Examination Scheduling System, *Interfaces* 11 (1994), 109–120.

[CaLaLe96] M. W. Carter, G. Laporte, and S. Lee, Examination Timetabling: Algorithmic Strategies and Applications, *J. of the Operations Research Society* 47 (1996), 373–383.

[Ch71] P. Z. Chinn, A Graph with p Points and Enough Distinct $p-2$-order Subgraphs is Reconstructible, in M. Capobianco, J. B. Frechen, and M. Krolik (Eds.), *Recent Trends in Graph Theory*, Springer, (1971), 71–73.

[Co64] A. J. Cole, The Preparation of Examination Timetables Using a Small Store Computer, *The Computer J.* 7 (1964), 117–121.

[CoKi93] T. B. Cooper and J. H. Kingston, The Solution of Real Instances of the Timetabling Problem, *The Computer J.* 36 (1993), 645–653.

[Cs65] J. Csima, *Investigations on a Time-Table Problem*, Phd thesis, School of Graduate Studies, University of Toronto, 1965.

[de81] D. de Werra, Scheduling in Sports, in P. Handesn (Ed.), *Studies on Graphs and Discrete Programming*, North-Holland, (1981), 381–395.

[de82] D. de Werra, Minimizing Irregularities in Sports Schedules Using Graph Theory, *Discrete Applied Mathematics* 4 (1982), 217–226.

[de85-a] D. de Werra, An Introduction to Timetabling, *Euro. J. Oper. Res.* 19 (1985), 151–162.

[de85-b] D. de Werra, Graphs, Hyper-graphs and Timetabling, *Methods of Operations Research (Germany, F.R.)* 49 (1985), 201–213.

[de85-c] D. de Werra, On the Multiplication of Divisions: the Use of Graphs for Sports Scheduling, *Networks* 4 (1985), 125–136.

[de88] D. de Werra, Some Models of Graphs for Scheduling Sports Competitions, *Discrete Applied Mathematics* 21 (1988), 47–65.

[de97] D. de Werra, The Combinatorics of Timetabling, *Euro. J. of Oper. Res.* 96 (1997), 504–513.

[DeVa02] P. De Causmaecker and G. Vanden Berghe, Relaxation of Coverage Constraints in Hospital Personnel Rostering, in E. Burke and P. De Causmaecker (Eds.), *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent, Belgium, August, selected papers) (Lecture Notes in Computer Science, Vol. 2740)*, Springer, 2003.

[EaNeTr03] K. Easton, G. Nemhauser, and M. Trick, Solving the Traveling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach, in E. Burke and P. De Causmaecker (Eds.), *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent, Belgium, August, selected papers) (Lecture Notes in Computer Science, Vol. 2740)*, Springer, 2003.

[EvItSh76] S. Even, A. Itai, and A. Shamir, On the complexity of timetable multicommodity flow problems, *SIAM J. Comput.* 5 (1976), 691–703.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W.H. Freeman and Co., 1979.

[GlKo03] F. Glover and K. Kochenberger, *Handbook of Metaheuristics*, Kluwer, 2003.

[Go62] C. C. Gotlieb, The Construction of Class-Teacher Timetables, *Proceedings of the IFIP Congress*, (1962), 73–77.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[Hä83] R. Häggkvist, *Restricted Edge Colorings of Bipartite Graphs*, Manuscript 1983.

[Hede87] A. Hertz and D. de Werra, Using Tabu Search Techniques for Graph Coloring, *Computing* 39 (1987), 345–351.

[Ki00] J. H. Kingston, Modelling Timetabling Problems with STTL, in E. Burke and W. Erben (Eds.), *Practice and Theory of Automated Timetabling II (Lecture Notes in Computer Science, Vol. 2079)*, Springer, (2001), 342–350.

[Me81] N. K. Mehta, The Application of a Graph Colouring Method to an Examination Scheduling Problem, *Interfaces* 11 (1981), 57–64.

[Me82] N. K. Mehta, A Computer Based Examination Management System, *J. of Educational Technology Systems* 11 (1982), 185–198.

[MiIwMa02] R. Miyashiro, H. Iwasaki, and T. Matsui, *Characterizing Feasible Pattern Sets with a Minimum Number of Breaks*, in E. Burke and P. De Causmaecker (Eds.), *Practice and Theory of Automated Timetabling IV (PATAT 2002, Gent, Belgium, August, selected papers) (Lecture Notes in Computer Science, Vol. 2740)*, Springer, 2003.

[Pa82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

[Sc99] A. Schaerf, A Survey of Automated Timetabling, *Artificial Intelligence Review* 13 (1999), 87–127.

[ScSt80] G. Schmidt and T. Ströhlein, Timetable Construction—an Annotated Bibliography, *The Computer J.* 23 (1980), 307–316.

[ThDo96-a] J. Thompson and K. Dowsland, Variants of Simulated Annealing for the Examination Timetabling Problem, *Annals of Operations Research* 63 (1996), 105–128.

[ThDo96-b] J. Thompson and K. Dowsland, General Cooling Schedules for a Simulated Annealing Based Timetabling System, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*, Springer, (1996), 345–363.

[WePo67] D. J. A. Welsh and M. B. Powell, An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems, *The Computer J.* 10 (1967), 85–86.

[Wo68] D. C. Wood, A System for Computing University Examination Timetables, *The Computer J.* 11 (1968), 41–47.

[Wr96] A. Wren, Scheduling, Timetabling and Rostering – A Special Relationship?, in E. Burke and P. Ross (Eds.), *The Practice and Theory of Automated Timetabling I (PATAT 1995, Edinburgh, Aug/Sept, selected papers) (Lecture Notes in Computer Science, Vol. 1153)*, Springer, (1996), 46–75.

GLOSSARY FOR CHAPTER 5

achromatic number – of a graph G: largest number of colors in a proper vertex-coloring such that the union of any two color classes induces at least one edge.

almost regular factor (or semiregular factor) - of graph G: a factor of G of type $[k, k+1]$, for some integer $k \geq 0$.

antifactor set (or 1-barrier) – in graph G: a set $S \subseteq V(G)$ such that $c_o(G-S) > |S|$.

antihole: the complementary graph of a hole.

approximation (or approximate) algorithm: an algorithm that typically makes use of heuristics in reducing its computation but produces solutions that are not necessarily optimal.

$r(n)$**-approximation algorithm** – for χ: for every n and every input graph G with n vertices, the algorithm outputs an integer k such that $\chi(G) \leq k \leq r(n)\chi(G)$ (where $r(n) : \mathbb{N} \to \mathbb{R}^+$ is a given function).

arboricity – of a graph G: the minimum number of edge-disjoint acyclic subgraphs whose union is G.

k**-assignment** – on the vertices (edges) of a graph: a list assignment L where $|L_v| = k$ ($|L_e| = k$) for every vertex v (every edge e).

asteroidal triple – in a graph G: a set of three vertices in G such that there is a path P between any pair of vertices in the triple with no vertex on P adjacent to the third vertex in the triple.

asymptotically almost surely (or *a.a.s.*): an event E concerning a graph $G \in G_{n,p}$ is said to hold asymptotically almost surely (or a.a.s.), if $\lim_{n\to\infty}$ Prob $E = 1$.

1-barrier (or antifactor set) – in graph G: a set $S \subseteq V(G)$ such that $c_o(G-S) > |S|$.

Berge graph: a graph with no hole or antihole.

bicritical graph: a graph G in which $G - x - y$ has a 1-factor for every choice of two different vertices x and $y \in V(G)$.

bi-hypergraph: a mixed hypergraph with $\mathcal{C} = \mathcal{D}$.

binding number bind (G) – of a graph G: defined to be

$$\min\left\{|N(X)|/|X| \;\middle|\; \emptyset \neq X \subseteq V(G), \text{ and } N(X) \neq V(G)\right\}$$

bipartite graph: a graph G whose vertex set V can be partitioned into two sets V_1 and V_2 such that every edge of G connects a vertex in V_1 with a vertex in V_2.

choice number – of graph G: the smallest nonnegative integer k such that G is k-choosable. Denoted by $ch(G)$.

(f,g)**-choosable graph** – for two functions $f, g : V \to \mathbb{N}$: if $|L_v| = f(v)$ for every vertex, then one can choose subsets $C_v \subseteq L_v$ such that $|C_v| = g(v)$ and $C_u \cap C_v = \emptyset$ for all $uv \in E$.

k**-choosable graph**: L-colorable for every k-assignment L.

f**-choosable graph** – for a function $f : V \to \mathbb{N}$: L-colorable for every list assignment L with $|L_v| = f(v)$ for all $v \in V$.

chordal graph: a graph in which every circuit of length ≥ 4 has a *chord* (i.e., an edge joining non-consecutive vertices on the circuit).

k-chromatic graph: has precisely k as the smallest number of colors in a proper vertex coloring = k-colorable but not $(k-1)$-colorable.

chromatic index – of graph G: smallest number of colors in a proper edge-coloring of G; same as the chromatic number of the line graph. Denoted by $\chi'(G)$.

chromatic number – of graph G: the minimum number of colors in a proper vertex coloring of G. Denoted by $\chi(G)$.

chromatic polynomial – of graph G: for every natural number k, its value is the number of proper k-colorings of G; denoted $P(G, \lambda)$.

chromatic sum – of graph G: smallest sum of colors in a proper vertex-coloring with *natural numbers*; denoted $\Sigma(G)$.

Class 1 / 2: graph G is of Class 1 if its chromatic index is $\Delta(G)$, and of Class 2 if $\chi'(G) = \Delta(G) + 1$.

claw – of graph G: an induced subgraph of graph G isomorphic to the bipartite graph $K_{1,3}$.

claw-free graph G: a graph G containing no $K_{1,3}$ as an induced subgraph.

clique cover – of a graph G: a collection of cliques of G that contains every vertex of G.

clique hypergraph – of a graph $G = (V, E)$: the hypergraph on V whose edges are the vertex subsets inducing inclusionwise-maximal complete subgraphs in G, other than isolated vertices. Its chromatic number is denoted $\chi_C(G)$.

clique incidence matrix – of a graph G: a 0-1 matrix, denoted $\mathbf{A}(G)$, whose rows are incidence vectors of the cliques of G; that is, entry (i, j) in this matrix is 1 if and only if the j^{th} vertex is in the i^{th} clique.

clique number – of a graph G, denoted $\omega(G)$: the size of the largest clique in G.

clique partition number – of graph G: the smallest number $cp(G)$ such that there exists a set of $cp(G)$ cliques in G such that the cliques form a partition of $E(G)$.

clique$_1$ – in a graph G: a subset of vertices in G that are mutually adjacent to one another (caution: non-uniform definition).

clique$_2$ – in a graph G: a maximal mutually adjacent set of vertices (caution: non-uniform definition).

color class – in a vertex (edge) coloring: set consisting of all vertices (edges) having the same color.

color cost: same as chromatic sum.

k-colorable graph: has a proper vertex coloring with at most k colors.

colorable mixed hypergraph: a mixed hypergraph that has at least one strict coloring.

L-colorable – graph G, with respect to list assignment L: if G admits a proper vertex coloring φ such that $\varphi(v) \in L_v$ for all v.

coloring number – of graph G: smallest integer k such that every subgraph of G contains a vertex of degree less than k; denoted $col(G)$.

H-coloring – of graph G: homomorphism from G to H.

k-coloring: coloring with at most k colors.

comparability graph: a graph whose edges can be directed so that directed adjacency becomes a transitive relation, that is, whenever there exist directed edges (a, b) and (b, c) there must also exist the directed edge (a, c).

complement – of a graph G: the graph $G^c = (V, E^c)$ which is related to graph $G = (V, E)$ as follows: it has the same vertex set V as G and edges defined by (x, y) is in E^c if and only (x, y) is not in E.

complete r-uniform hypergraph: its edges are all the r-element subsets of the vertex set; denoted $\mathcal{K}_n^r$ (n is the number of vertices).

conflict graph: a graph in which the nodes represent events (e.g., courses, exams) and an edge between two nodes indicates that the two events cannot be scheduled in the same time slot.

consecutive 1's property – in a 0-1 matrix $\mathbf{M}$: this property exists if the rows of $\mathbf{M}$ can be rearranged so that the 1's in each column occur consecutively.

construction heuristic: a heuristic that produces a feasible solution without any attempt to improve it.

cost chromatic number – of graph G, with given cost set C: smallest number of colors in a minimum-cost coloring.

cost set: associates a positive real cost with each color.

k-critical graph: k-chromatic graph whose chromatic number decreases to $k-1$ whenever an edge is deleted.

critical imperfect graph: a graph which is not perfect but all proper vertex-induced subgraphs are perfect.

cyclically k-edge-connected graph: a graph in which at least k edges must be deleted in order to leave two components, each containing a cycle.

H-decomposition problem: the problem defined as follows: given a fixed graph H, can the edge-set of an input graph G be partitioned into copies of H?

divisible by t: A graph G which admits a partition of its edge-set into t isomorphic subgraphs is said to be *divisible by t*.

edge k-colorable graph: has a proper edge-coloring with at most k colors.

edge choice number – of graph G: the choice number of the line graph of G; denoted $ch'(G)$.

edge,C-, D-edge – of a mixed hypergraph: see *strict coloring*.

edge-chromatic number – of a graph G: is the minimum number of different colors required for a proper edge-coloring of G; denoted $\chi'(G)$.

edge-coloring: assignment of colors to the edges (each edge gets one color).

___, **proper**: edge-coloring where any two edges sharing a vertex have distinct colors.

exact algorithm: an algorithm that solves a certain optimization problem to optimality.

factor - of a graph G: a spanning subgraph of G.

___, $(1,f)$**-odd-**: a spanning subgraph F of G in which $\deg_F(v) \in \{1,3,\ldots,f(v)\}$, where f is a function from $V(G)$ to the odd positive integers.

___, (g,f)**-**: a spanning subgraph F of G such that $g(v) \le \deg_F(G) \le f(v)$ for all $v \in V(G)$.

___, $[a,b]$**-**: a factor of G for which $a \le \deg_H(v) \le b$, for all $v \in V(G)$, where a and b are integers such that $1 \le a \le b$.

___, **1-**: a set of vertex-disjoint edges in G which together span $V(G)$.

___, F**-**: a spanning subgraph of G in which each component is a single edge or an odd cycle.

___, f**-** – of multigraph G (possibly with loops): a spanning subgraph H of G such that $\deg_H(v) = f(v)$, for all $v \in V(G)$, where f, a non-negative, integer-valued function on $V(G)$.

___, G**-** – of a graph H: a set $\{G_1,\ldots,G_d\}$ of subgraphs of H such that each G_i is isomorphic to G and such that the sets $V(G_i)$ collectively partition $V(G)$.

___, k**-**: a k-regular spanning subgraph.

G**-factor recognition problem** $FACT(G)$: defined by
INSTANCE: A graph H.
QUESTION: Does H admit a G-factor?

factorization – of graph G: a set of factors $\{F_1,F_2,\ldots F_k\}$ of G such that the edge-disjoint union of factors $F_1,F_2,\ldots F_k$ is $E(G)$.

all (g,f)**-factors**: a graph G is said to have all (g,f)-factors if and only if G has an h-factor for every h such that $g(v) \le h(v) \le f(v)$ for all $v \in V(G)$.

feasible solution: a solution that satisfies all hard constraints.

fractional chromatic number – of a graph G: smallest ratio p/q such that there exist p independent sets that cover each vertex precisely q times; denoted $\chi^*(G)$.

fractional vertex-coloring: function from the family of independent vertex sets to $\mathbb{R}^{\ge 0}$, such that the sum over the sets containing vertex v is at least 1, for each v.

fractional (g,f)-factor – of graph G: a vector $x = (x_e)$ with $|E(G)|$ real components such that $0 \le x_e \le c_e$ and $g(v) \le \deg_x(v) \le f(v)$. Here $\deg_x(v) = \sum x_{uv}$, where the sum is over all edges incident with vertex v and c_e is the "capacity" or perhaps the "multiplicity" of edge e.

$[a,b]$**-graph**: a graph G in which $a \le \deg(v) \le b$, for every vertex $v \in V(G)$.

graphical – degree sequence: a sequence of non-negative integers $d_1,\ldots,d_n$ such that there exists a graph G of order n and degrees (in some order) $d_1,\ldots,d_n$.

Grundy number – of a graph G: largest number of colors in a proper vertex-coloring with natural numbers where each vertex v has a neighbor in each color smaller than the color of v.

Hamming distance – between a pair $u = (u_1,\ldots,u_n)$, $v = (v_1,\ldots,v_n)$ of binary vectors: the number of indices i for which $u_i \ne v_i$.

Hamming graph $H(n,d)$: a graph whose vertices are all binary vectors with n coordinates. A pair u,v of vertices in $H(n,d)$ are adjacent if the Hamming distance between them is at least d.

hard constraint: a constraint that must be satisfied.

hole – in a graph G: a circuit of odd length ≥ 5 with no chords, i.e., no edges between non-consecutive vertices on the circuit.

homomorphism – from (di)graph G to H: maps the vertices of G to vertices of H in such a way that the image of every edge is an edge of H. Notation if it exists: $G \to H$; if it does not exist: $G \not\to H$.

hypergraph: a pair $\mathcal{H} = (X, \mathcal{F})$ where X is a set (vertex set) and $\mathcal{F}$ is a set system on X (edge set).

hypohamiltonian – graph G: a graph G which has no Hamilton cycle, but for which $G - v$ does, for all $v \in V(G)$.

hypotraceable – graph G: a graph G which has no Hamilton path, but for which $G - v$ does, for all $v \in V(G)$.

independence number: the number of vertices in a maximum-size independent set of a graph.

independent set – of a graph G: a subset of vertices in G that are mutually non-adjacent.

independent set: a mutually non-adjacent set of vertices.

independent-set cover – of a graph G: a collection of independent sets of G that contains every vertex of G.

independent-set incidence matrix – of a graph G: a 0-1 matrix, denoted $\mathbf{B}(G)$, whose rows are incidence vector of the independent sets of G; that is, entry (i, j) in this matrix is 1 if and only if the j^{th} vertex is in the i^{th} independent set.

intersection graph – of a family F of subsets of a given set: a graph $G(F)$ with a 1-to-1 correspondence between subsets of F and vertices of G such that two vertices of G are adjacent if and only if they correspond to two subsets of F with a non-empty intersection.

interval graph: a graph for which there exists a family F of intervals on a line such that G is an intersection graph, that is, there is a 1-to-1 correspondence between intervals of F and vertices of G such that two vertices of G are adjacent if and only if they correspond to overlapping intervals of F.

proper: an interval graph with the property that there is an interval model F for G in which no interval of F is properly contained within another interval of F.

interval model – for an interval graph G: a family of intervals on the line for which G is an intersection graph.

2-join – in a graph $G = (V, E)$: a partition of V into two sets X_1 and X_2 such that there exist disjoint non-empty $A_i, B_i \subseteq X_i, i = 1, 2$, satisfying the following three conditions.

(i) The pairs (A_1, A_2) and (B_1, B_2) form complete bipartite subgraphs, and there are no other edges between X_1 and X_2.

(ii) For $i = 1, 2$, every component of the subgraph induced by X_i contains at least one member of A_i and of B_i.

(iii) For $i = 1, 2$, if A_i and B_i each consist of a single vertex and the subgraph induced by X_i consists of a path, then that path has length ≥ 3.

M-join – in a graph $G = (V, E)$: a partition of the vertex set V into six nonempty sets, A, B, C, D, E, F satisfying the following three conditions:

(i) Every vertex in A has a neighbor in B and a nonneighbor in B, and vice versa.

(ii) The pairs (A, C), (A, F), (B, D) and (B, F) form complete bipartite subgraphs.

(iii) There are no edges between the following pairs of vertex sets: (A, D), (A, E), (B, C) and (B, E).

line graph – of a graph G: a graph, denoted $L(G)$, with a vertex for each edge of G and two vertices of $L(G)$ joined by an edge if and only if they correspond to two edges in G with a common endpoint.

***k*-linear arboricity** – of a graph G: the minimum number of k-linear forests which partition $E(G)$.

***k*-linear forest**: a forest in which all components are paths of length at most k.

list assignment L – on the vertex set of graph G: associates a set L_v of "allowed" colors with each vertex v of G.

list chromatic index or **list edge chromatic number**: same as edge choice number.

list chromatic number: same as choice number.

list colorable graph: L-colorable with respect to a list assignment L that is understood.

local search: a heuristic that starts from a feasible solution and in each iteration, until termination, chooses the next solution from a *neighborhood* of solutions that are, in some prescribed sense, close to the current solution.

___, **improvement**: a local search in which we choose only a solution that is better than the current one and stop if we cannot find one.

lower chromatic number – of a colorable mixed hypergraph: the smallest number of colors in a strict coloring; denoted $\chi(\mathcal{H})$.

matching number of G: the size of a largest matching in G; denoted $\nu(G)$.

maximum clique: the number of vertices in a maximum-size clique of a graph.

mixed hypergraph: has two sets of edges, $\mathcal{C}$ and $\mathcal{D}$. (Cf. *strict coloring*.)

___, C-: mixed hypergraph without D-edges.

___, D-: mixed hypergraph without C-edges. Same as hypergraph.

___, **uncolorable**: does not have any strict coloring.

___, **uniquely colorable** : has just one strict coloring, apart from renaming the colors.

neighborhood complex – of graph G: simplicial complex whose vertices are the vertices of G, and the simplexes are the vertex subsets having a common neighbor in G.

nowhere-zero *k*-flow – on an oriented graph: weight function from the edge set to $\{0, 1, \ldots, k-1\}$, such that the in-flow (sum of weights on the edges oriented toward a vertex) equals the out-flow (sum on the out-going edges), at each vertex.

odd-cycle property: the property which states that every pair of odd cycles either have a vertex in common or are joined by an edge.

on-line coloring: receiving the vertices $v_1, \ldots, v_n$ of a graph G one by one, a color has to be assigned to each successive vertex v_i after only its neighbors in $\{v_1, \ldots, v_{i-1}\}$ are known.

oriented d-coloring – of graph G: a proper edge-d-coloring together with an assignment of a direction to each edge; used to model the schedule of home and away games in sports timetabling.

partial vertex/edge **coloring**: assignment of colors to a subset of the vertices/edges.

partition number – of a graph G, denoted $\rho(G)$: the size of the smallest clique cover of G.

path factor – of graph G: spanning subgraph of G each component of which is a path.

Perfect Graph Theorem, The: the theorem that a graph is α-perfect if and only if its complement is α-perfect.

perfect graph: a graph that is α-perfect and ω-perfect.

perfect matrix: a 0-1 matrix **A** with the property that for every integer-valued vector **c**, there exists an integer-value vector **x** that achieves the maximum value of the linear program: Max **cx**

α-perfect graph: when in every (non-empty) vertex-induced subgraph H of G (including $H = G$), $\alpha(H) = \rho(H)$.

ω-perfect graph: when in every (non-empty) vertex-induced subgraph H of G (including $H = G$), $\omega(H) = \chi(H)$.

permanent – of matrix A: the matrix function defined by

$$\text{per } A = \sum a_{1\pi(1)} a_{2\pi(2)} \cdots a_{n\pi(n)}$$

where the sum extends over all permutations π of the set $\{1, \ldots, n\}$; denoted per A.

permutation graph: a graph G with n vertices for which there exists a permutation π of the numbers 1 to n such that (x_i, x_j), for $i < j$, is an edge of G if and only if $\pi(j) < \pi(i)$.

Perron root $\lambda(G)$: the largest eigenvalue of A_G.

planar dual – of a plane graph G: the dual vertices are the faces of G, and the endpoints of a dual edge are the faces whose boundary contains the original edge; denoted $G^* = (V^*, E^*)$.

plane graph: planar graph imbedded in the plane.

precoloring extension or **PrExt** problem: asks whether a given partial coloring can be extended to a proper coloring of the entire graph, using at most a given number k of colors.

t-PrExt: Precoloring Extension where each color occurs on at most t vertices in the given partial coloring.

proper vertex-coloring – of a hypergraph $\mathcal{H}$: vertex partition where no partition class contains any edge of $\mathcal{H}$.

random graph $G_{n,p}$: Let $1, \ldots, n$ be a labelling of the vertices and let $\{e_{ij}\}, 1 \leq i < j \leq n$, be an array of independent random variables, where each e_{ij} assumes the value 1 with probability p and 0 with probability $1 - p$. This array determines a random graph on $\{1, \ldots, n\}$ where each (ij) is an edge if and only if $e_{ij} = 1$. This probability space (or random graph) is denoted by $G_{n,p}$.

ranking number – of graph G: smallest number of colors in a vertex-coloring such that each path with the same color on its endpoints contains a vertex of larger color; denoted $\chi_r(G)$.

t-rational graph G: a graph G which is divisible by t or else $t \not| \, |E(G)|$.

t-**rational Problem:** The problem defined by: Given a graph G and a positive integer t, is G t-rational?

simplicial vertex – of a graph G: a vertex whose neighboring vertices form a clique.

snark: 3-regular graph of chromatic index 4, which is also cyclically 4-edge-connected and has girth at least 5.

soft constraint: a constraint that it is desirable, but not necessary, to satisfy.

square – of graph G: is obtained from G by joining also the vertex pairs at distance 2; denoted G^2.

stability number – of a graph G, denoted $\alpha(G)$: the size of the largest independent set in G.

standard simplex: $\Delta = \{x \in R^n : \ x_i \geq 0, \ i = 1, \ldots, n, \ \sum_{j=1}^{n} x_j = 1\}$.

strength – of graph G: cost chromatic number of G where the cost set is $\mathbb{N}$.

strict k**-coloring** – of a mixed hypergraph: vertex-coloring with exactly k colors, such that every C-edge has two vertices with a <u>c</u>ommon color and every D-edge has two vertices with <u>d</u>ifferent colors.

Strong Perfect Graph Theorem, The: the theorem that a graph is perfect if and only if it contains no holes or anti-holes.

Szekeres–Wilf number: same as coloring number.

tabu search: a local search in which solutions that are worse than the current one can be chosen provided that they are not in any of the so-called tabu lists.

timetabling problem: the assignment of times and resources to meetings so as to satisfy a set of constraints as best as possible.

total coloring: assignment of colors to the vertices and to the edges (each vertex and each edge gets one color).

___, **proper**: total coloring where no two adjacent or incident vertices/edges have the same color.

total graph – of $G = (V, E)$: its vertex set is $V \cup E$, and $x, y \in V \cup E$ are adjacent if they are incident or adjacent in G; denoted $T(G)$.

toughness – of graph G: defined to be $+\infty$ when G is complete and otherwise to be

$$\min\{|S|/c(G-S) | S \subseteq V(G)\},$$

where the minimum is taken over all subsets $S \subseteq V(G)$ and $c(G-S)$ denotes the number of components of $G - S$. Denoted tough (G).

triangulation: graph imbedded in a surface, with all faces being cycles of length 3.

r-**uniform hypergraph**: every edge has precisely r vertices.

uniquely (vertex-)colorable graph: has just one proper coloring with the minimum number of colors, apart from renaming the colors.

uniquely edge-colorable graph: has just one proper edge-coloring with the minimum number of colors, apart from renaming the colors.

upper chromatic number – of a colorable mixed hypergraph: the largest number of colors in a strict coloring; denoted $\overline{\chi}(\mathcal{H})$.

vertex cover – of graph G: a subset $C \subseteq V(G)$ such that every edge of G has at least one endpoint in C.

k-vertex-critical graph: k-chromatic graph whose chromatic number decreases to $k-1$ whenever a vertex is deleted.

vertex-coloring: assignment of colors to the vertices (each vertex gets one color).

___, **proper**: coloring where no two adjacent vertices have the same color.

vertex-covering number – of graph G: size of any smallest vertex cover in G; denoted $\tau(G)$.

adjacency matrix – of a graph G: a 0-1 matrix with a row and a column for each vertex and entry (i, j) is 1 if and only if the i^{th} vertex is adjacent to the jth vertex.

vertex-induced subgraph: a subgrapn $H = (V', E')$ of a graph $G = (V, E)$ with the property that $V' \subseteq V$ and for any two vertices $x, y \in V'$, $(x, y) \in E'$ if and only if $(x, y) \notin E$.

vertex-weighted graph: a graph G in which every vertex x is assigned a non-negative weight $w(x)$.

Chapter 6

ALGEBRAIC GRAPH THEORY

6.1 AUTOMORPHISMS

Mark E. Watkins, Syracuse University

6.2 CAYLEY GRAPHS

Brian Alspach, University of Regina, Canada

6.3 ENUMERATION

Paul K. Stockmeyer, The College of William and Mary

6.4 GRAPHS AND VECTOR SPACES

Krishnaiyan "KT" Thulasiraman, University of Oklahoma

6.5 SPECTRAL GRAPH THEORY

Michael Doob, University of Manitoba, Canada

6.6 MATROIDAL METHODS IN GRAPH THEORY

James Oxley, Louisiana State University

GLOSSARY

6.1 AUTOMORPHISMS

Mark E. Watkins, Syracuse University

6.1.1 The Automorphism Group
6.1.2 Graphs with Given Group
6.1.3 Groups of Graph Products
6.1.4 Transitivity
6.1.5 s-Regularity and s-Transitivity
6.1.6 Graphical Regular Representations
6.1.7 Primitivity
6.1.8 More Automorphisms of Infinite Graphs
References

Introduction

An automorphism of a graph is a permutation of its vertex set that preserves incidences of vertices and edges. Under composition, the set of automorphisms of a graph forms a group that gives much information about both the local and the global structure of the graph. It may determine the graph's connectivity stucture (§4.2) and the kinds of surfaces in which it may be imbedded (§7.1, §7.5). It is indispensable for counting the number of essentially "distinct" graphs with a variety of different properties (§6.3).

In this section, graphs are assumed to be simple. There are references to infinite graphs, but in most cases the graphs are "locally finite", i.e., all valences are finite, though they may be arbitrarily large.

NOTATION: We need to use the formal notation $\{u, v\}$ (rather than uv) to represent an edge with endpoints u and v, for reasons that become clear in §6.1.3.

6.1.1 The Automorphism Group

DEFINITIONS

D1: Given a graph X, a permutation α of $V(X)$ is an ***automorphism*** of X if

$$\{u, v\} \in E(X) \Leftrightarrow \{\alpha(u), \alpha(v)\} \in E(X), \text{ for all } u, v \in V(X)$$

D2: The set of all automorphisms of X, together with the operation of composition of functions, forms a subgroup of the symmetric group on $V(X)$ called the ***automorphism group*** of X, and it is denoted by $\mathrm{Aut}(X)$.

NOTATION: The identity of any permutation group is denoted by ι.

D3: A graph is ***asymmetric*** if the identity ι is its only automorphism.

REMARKS

R1: While all vertices in the same *orbit* of $\mathrm{Aut}(X)$ must have the same valence, there exist asymmetric graphs all of whose vertices have the same valence.

R2: If Y is a subgraph of X, then except in special cases there is no relationship between $\mathrm{Aut}(X)$ and $\mathrm{Aut}(Y)$.

R3: When saying that the automorphism group of a graph X "is isomorphic to" a group G, it is ambiguous whether we mean that the isomorphism is between *abstract* groups or between *permutation* groups (see §6.1.2). In the examples immediately below, the automorphism groups $\mathrm{Aut}(X)$ are abstractly isomorphic to the given groups G.

EXAMPLES

E1: Let $V(K_4) = \{a,b,c,d\}$ and let $X = K_4 - \{a,c\}$, the result of removing edge $\{a,c\}$ from K_4. Then $\mathrm{Aut}(X) = \{\iota, \alpha, \beta, \alpha\beta\}$, where α interchanges a and c but fixes both b and d, while β fixes a and c but interchanges b and d. Thus $\mathrm{Aut}(X) = \mathbb{Z}_2 \times \mathbb{Z}_2$. Strictly speaking, one should say that $\mathrm{Aut}(X)$ is *isomorphic* to $\mathbb{Z}_2 \times \mathbb{Z}_2$, but for brevity we will often abuse language in this way throughout this section.

E2: The automorphism group of a circuit of length n is the dihedral group D_n with $2n$ elements.

E3: [Fr37] The automorphism group of the Petersen graph is isomorphic to $\mathrm{Sym}(5)$, the symmetric group on five objects.

E4: The automorphism group of a *double ray* (2-way infinite path) is the infinite dihedral group D_∞, but a *ray* (1-way infinite path) is asymmetric.

FACTS

F1: Let the components of X be $X_1, \ldots, X_k$. Then $\mathrm{Aut}(X)$ is the direct product $\prod_{i=1}^{k} \mathrm{Aut}(X_i)$.

F2: For any graph X, the automorphism group X and that of its edge-complement $\overline{X}$ are identical.

F3: Given any finite tree, either there is a unique vertex or there is a unique edge that is fixed by every automorphism.

6.1.2 Graphs with Given Group

DEFINITIONS

D4: A group G of permutations of a set S ***acts transitively*** or ***is transitive*** on S if for every $x, y \in S$, there exists $\alpha \in G$ such that $\alpha(x) = y$.

D5: A graph X is said to be ***vertex-transitive*** if $\mathrm{Aut}(X)$ acts transitively on $V(X)$. Intuitively speaking, a vertex-transitive graph looks the same, no matter from what vertex it is viewed.

D6: A group G of permutations of a set S ***acts doubly transitively*** on S if for any two ordered pairs of distinct elements $(x_1, x_2), (y_1, y_2) \in S \times S$ there exists $\alpha \in G$ such that $\alpha(x_1) = y_1$ and $\alpha(x_2) = y_2$.

D7: For $i = 1, 2$, let G_i be a group of permutations of the set S_i. We say that G_1 and G_2 are ***isomorphic as permutation groups*** if there exist a group-isomorphism $\Phi : G_1 \to G_2$ and a bijection $f : S_1 \to S_2$ such that

$$f(\alpha(x)) = [\Phi(\alpha)](f(x)) \text{ for all } \alpha \in G_1,\ x \in S_1,$$

i.e., the diagram in Figure 6.1.1 commutes.

$$\begin{tikzcd}
S_1 \arrow[r, "f"] \arrow[d, "\alpha"'] & S_2 \arrow[d, "\Phi(\alpha)"] \\
S_1 \arrow[r, "f"] & S_2
\end{tikzcd}$$

Figure 6.1.1 **Isomorphism of permutation groups.**

In this case, the ***order*** of G is $|G|$, and the ***degree*** of G is $|S|$.

D8: An ***edge-isomorphism*** from a graph X_1 to a graph X_2 is a bijection $\eta : E(X_1) \to E(X_2)$ such that edges e_1 and e_2 are incident with a common vertex of X_1 if and only if $\eta(e_1)$ and $\eta(e_2)$ are incident with a common vertex of X_2.

D9: An ***edge-automorphism*** is an edge-isomorphism from a graph to itself.

D10: The set of edge-automorphisms forms a subgroup of the symmetric group on $E(X)$; it is called the ***edge-group*** of X.

NOTATION

NOTATION: For a finite group G, let $\mu(G)$ denote the least $|V(X)|$ such that $\mathrm{Aut}(X)$ is isomorphic to G.

EXAMPLES

E5: $\mathrm{Aut}(K_n)$ is the symmetric group $\mathrm{Sym}(n)$. Here the isomorphism is between permutation groups.

E6: For no graph X on n vertices is $\mathrm{Aut}(X)$ ever isomorphic to the alternating group $\mathrm{Alt}(n)$. This is because $\mathrm{Alt}(n)$ acts doubly transitively on an n-set. The only graphs on n vertices whose automorphism group acts doubly transitively are K_n and its complement, the edgeless graph $\overline{K_n}$, but the automorphism group of these two graphs is $\mathrm{Sym}(n)$.

E7: The edge-groups of the 3-circuit C_3 and of $K_{1,3}$ are isomorphic as permutation groups to each other and to $\mathrm{Aut}(C_3)$ but are abstractly isomorphic to $\mathrm{Aut}(K_{1,3})$.

FACTS

Every automorphism α of a graph X induces a unique edge-automorphism η_α; namely, if $\{u, v\} \in E(X)$, then $\eta_\alpha(\{u, v\}) = \{\alpha(u), \alpha(v)\}$. The converse is not true.

F4: [HarPa68] The edge-group of a graph X and $\mathrm{Aut}(X)$ are (abstractly) isomorphic if and only if X has at most one isolated vertex and K_2 is not a component of X.

F5: [Wh32] Let X_1 an X_2 be connected graphs, neither of which is isomorphic to $K_{1,3}$. If there exists an edge-automorphism from X_1 to X_2, then X_1 and X_2 are isomorphic graphs.

F6: [Fr38] **Frucht's Theorem:** Given any group G, there exist infinitely many connected graphs X such that $\mathrm{Aut}(X)$ is (abstractly) isomorphic to G. Moreover, X may be chosen to be 3-valent [Fr49].

F7: [Sa57] In Frucht's Theorem, in addition to having $\mathrm{Aut}(X)$ isomorphic to a given group G, one may further impose that X

- has connectivity κ for any integer $\kappa \geq 1$, *or*
- has chromatic number c for any integer $c \geq 2$ (see §5.1), *or*
- is r-valent for any integer $r \geq 3$, *or*
- is spanned by a graph $\tilde{Y}$ homeomorphic to a given connected graph Y.

In most cases, the graphs X that Frucht constructed had vertex-sets much larger than the order of G, and so there has been interest in seeking the smallest graph X such that $\mathrm{Aut}(X)$ is abstractly isomorphic to G. (If $\mathrm{Aut}(X)$ were to be isomorphic to G as a permutation group, then $|V(X)|$ would be prescribed by the degree of G.) Unless stated otherwise, one usually understands the word "isomorphic" between groups to mean "abstractly isomorphic".

F8: The asymmetric graph with the fewest edges is obtained from a path of length 5 by adjoining a new edge to a vertex at distance 2 from an end-vertex of the path, yielding a tree on seven vertices. Thus $\mu(\{\iota\}) = 7$.

F9: [Bab74] If G is a nontrivial finite group different from the cyclic groups of orders 3, 4, and 5, then $\mu(G) \leq 2|G|$.

F10: $\mu(\mathbb{Z}_3) = 9$; $\mu(\mathbb{Z}_4) = 10$; $\mu(\mathbb{Z}_5) = 15$. (See [Sa67].)

REMARK

R4: As Example 6 shows, Frucht's Theorem does not apply to isomorphism of permutation groups.

FURTHER READING

The automorphism groups of the generalized Petersen graphs are presented in detail in [FrGraWa71].

6.1.3 Groups of Graph Products

In this subsection, we use the symbol & to indicate an arbitrary graph product $X\&Y$ of graphs X and Y, where we define a ***graph product*** of graphs X and Y to be a graph with vertex set $V(X) \times V(Y)$, whose edge set is determined in a prescribed way by (and only by) the adjacency relations in X and in Y. It has been shown (see [ImIz75]) that there exist exactly 20 graph products that satisfy this definition. One is generally interested in products that are *associative*, in the sense that, for all graphs W, X, Y, the graphs $(W\&X)\&Y$ and $W\&(X\&Y)$ are isomorphic.

DEFINITIONS

The four most commonly used associative graph products are now defined.

D11: Let Z be a graph product of arbitrary graphs X and Y. Let x_1, x_2 be (not necessarily distinct) vertices of X, and let y_1, y_2 be (not necessarily distinct) vertices of Y. Suppose that $\{(x_1, y_1), (x_2, y_2)\} \in E(Z) \Leftrightarrow$

- $[\{x_1, x_2\} \in E(X)$ and $y_1 = y_2]$ or $[\{x_1 = x_2\}$ and $\{y_1, y_2\} \in E(Y)]$. Then Z is the ***cartesian product*** of X by Y, and we write $Z = X \square Y$;
- $[\{x_1, x_2\} \in E(X)$ and $y_1 = y_2]$ or $[x_1 = x_2$ and $\{y_1, y_2\} \in E(Y)]$ or $[\{x_1, x_2\} \in E(X)$ and $\{y_1, y_2\} \in E(Y)]$. Then Z is the ***strong product*** of X and Y, and we write $Z = X \boxtimes Y$;
- $\{x_1, x_2\} \in E(X)$ and $\{y_1, y_2\} \in V(Y)$. Then Z is the ***weak product*** or ***categorical product*** of X and Y, and we write $Z = X \times Y$;
- $\{x_1, x_2\} \in E(X)$ or $[x_1 = x_2$ and $\{y_1, y_2\} \in E(Y)]$. Then Z is the ***lexicographic product*** of X and Y, and we write $Z = X[Y]$.

These four products are illustrated in Figure 6.1.2, when both X and Y denote the path of length 2.

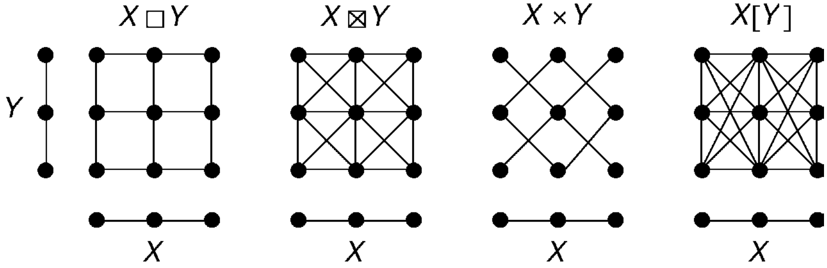


Figure 6.1.2 **The four products of the 2-path by the 2-path.**

D12: A graph X is a ***divisor*** of a graph Z (with respect to a product &) if there exists a graph Y such that $Z = X \& Y$ or $Z = Y \& X$.

D13: A graph Z is ***prime*** (with respect to a given product &) if Z has no ***proper divisor***, i.e., no divisor other than itself and the graph consisting of a single vertex.

D14: Graphs X and Y are ***relatively prime*** (with respect to a given product &) if they have no common proper divisor.

D15: Let G and H be groups of permutations of sets S and T, respectively. We define the ***wreath product*** $G \mathrm{wr} H$ to be the group of permutations π of $S \times T$ such that there exists $\alpha \in G$ and for each $s \in S$ there exists $\beta_s \in H$ such that $\pi(s,t) = (\alpha(s), \beta_{\alpha(s)}(t))$ for all $(s,t) \in S \times T$. The group operation is componentwise composition.

GENERAL FACTS

F11: [Im69] The lexicographic product is the only one of these four products that is not commutative. In fact, if $X[Y] \cong Y[X]$, then either both X and Y are complete, or both are edgeless, or both are powers (with respect to the lexicographic product) of the same graph W.

F12: The lexicographic product is the only one of these four products that is self-complementary, in the sense that for any graphs X and Y, we have $\overline{X}[\overline{Y}] \cong \overline{X[Y]}$.

FACTS ABOUT CONNECTEDNESS

F13: The cartesian (respectively, strong) product of two graphs X and Y is connected if and only if both X and Y are connected.

F14: Let X and Y be graphs with at least one edge. Then $X \times Y$ is connected if and only if both X and Y are connected and not both X and Y are bipartite.

F15: $X[Y]$ is connected if and only if X is connected.

FACTS ABOUT DECOMPOSITION

F16: [Sa60] Every connected graph has a unique prime decomposition with respect to the cartesian product and with respect to the strong product.

F17: Finite nonbipartite graphs have a prime decomposition into nonbipartite divisors with respect to the categorical product.

F18: [Im72] Any prime decomposition with respect to the lexicographic product can be transformed into any other by transpositions of edgeless or complete divisors. Thus, if a graph has a prime decomposition without such divisors, then it is unique.

FACTS ABOUT AUTOMORPHISMS

F19: [Sa60] If X is connected, then $\mathrm{Aut}(X)$ is generated by the automorphisms of its prime divisors with respect to the cartesian product and the transpositions interchanging isomorphic prime divisors. Fact 20 is an important corollary.

F20: Let X be the cartesian product $X = X_1\square, \cdots, \square X_k$ of relatively prime connected graphs. Then $\mathrm{Aut}(X)$ is the direct product $\prod_{i=1}^{k} \mathrm{Aut}(X_i)$.

The following notation is needed in order to characterize the group of the lexicographic product.

NOTATION: For a subgraph Y of X, let

$$\partial(Y) = \{x \in V(X)\backslash V(Y) : \{x, y\} \in E(X) \text{ for some } y \in V(Y)\}$$

If Y consists of a single vertex $Y = \{y\}$, we write simply $\partial(y)$.

NOTATION: We define binary relations $R(X)$ and $S(X)$ on $V(X)$ by

- $(u, v) \in R(X) \Leftrightarrow \partial(u) = \partial(v)$;
- $(u, v) \in S(X) \Leftrightarrow \partial(u) \cup \{u\} = \partial(v) \cup \{v\}$;
- $(u, v) \in \Delta(X) \Leftrightarrow u = v$.

F21: [Sa61] Let X and Y be graphs that are not edgeless. Then $\mathrm{Aut}(X[Y]) = \mathrm{Aut}(X)\mathrm{wr}\mathrm{Aut}(Y)$ if and only if the following two conditions hold:

(i) $R(X) \neq \Delta(X) \Rightarrow Y$ is connected, and

(ii) $S(X) \neq \Delta(X) \Rightarrow \overline{Y}$ is connected.

F22: Each of the four products $X\&Y$ is vertex-transitive if and only if both X and Y are vertex-transitive.

FURTHER READING

For a comprehensive and up-to-date treatment of all graph products, see [ImKl00].

6.1.4 Transitivity

DEFINITIONS

D16: If G is a group of permutations of a set S and $x \in S$, then the ***stabilizer*** of x (in G) is the subgroup $G_x = \{\alpha \in G : \alpha(x) = x\}$.

NOTATION: The stabilizer of a vertex $u \in V(X)$ in $\mathrm{Aut}(X)$ will be denoted by $\mathrm{Aut}_u(X)$.

D17: A graph X is ***edge-transitive*** if given e_1, $e_2 \in E(X)$, there exists an automorphism $\alpha \in \mathrm{Aut}(X)$ such that $\alpha(e_1) = \alpha(e_2)$.

D18: A graph X is ***arc-transitive*** if given ordered pairs (u_1, v_1), (u_2, v_2) of adjacent vertices, there exists $\alpha \in \mathrm{Aut}(X)$ such that $\alpha(u_1) = v_1$ and $\alpha(u_2) = v_2$.

D19: A graph that is vertex- and edge-transitive but not arc-transitive is said to be ***half-transitive***.

D20: A graph with constant valence that is edge-transitive but not vertex-transitive is said to be ***semisymmetric***.

D21: Given a graph X, let $d(k)$ denote the number of vertices at distance k from some given vertex, and let a be some real number such that $a > 1$. The following definitions are useful when discussing infinite locally finite graphs.

- The ***growth*** of X is defined to be $\mathrm{gr}(X) = \liminf_{k\to\infty}[d(k)/a^k]$.
- If $\mathrm{gr}(X) > 0$, then X has ***exponential growth***.
- If $\mathrm{gr}(X) = 0$, then the growth is ***subexponential***.
- If $\liminf_{k\to\infty}[\sum_{j=0}^{k} d(j)/k^n]$ is a positive number, then X has ***polynomial growth of degree*** n.
- If X has subexponential growth but grows faster than any polynomial, then X is said to have ***intermediate growth***.

FACTS

F23: If X is vertex-transitive, then for any $u, v \in V(X)$ we have:

- $\mathrm{Aut}_u(X)$ and $\mathrm{Aut}_v(X)$ are conjugate subgroups of $\mathrm{Aut}(X)$;
- $|\mathrm{Aut}_u(X)| = |\{\alpha \in \mathrm{Aut}(X) : \alpha(u) = v\}|$;
- If X is finite, then $|\mathrm{Aut}(X)| = |\mathrm{Aut}_u(X)| \cdot |V(X)|$.

F24: If a vertex-transitive graph is not connected, then all of its components are isomorphic and vertex-transitive.

F25: If a graph X is edge-transitive but not vertex-transitive, then it is bipartite. In this case, $\mathrm{Aut}(X)$ induces exactly two orbits in $V(X)$, namely, the two sides of the bipartition.

F26: [Fo67] The smallest semisymmetric graph is 4-valent and has 20 vertices.

F27: If a graph is arc-transitive, then it is both vertex-transitive and edge-transitive.

F28: [Tu66] **Tutte's Theorem:** Every finite half-transitive graph has even valence.

Finite half-transitive graphs do indeed exist, although they are not plentiful.

F29: [Bo70] For every positive integer n, there exists a half-transitive graph on $9 \cdot 6^n$ vertices and valence $2(n+1)$.

F30: [Ho81] The smallest half-transitive graph has 27 vertices and is 4-valent.

F31: [ThWa89] Every infinite half-transitive graph of subexponential growth has even valence.

F32: [Tr85] Let X be a connected, vertex-transitive, locally finite graph. Then the following are equivalent:

- X has polynomial growth;
- There is a system $\mathcal{S}$ of imprimitivity of $\mathrm{Aut}(X)$ on $V(X)$ with finite (possibly singleton) blocks such that $\mathrm{Aut}(X)/\mathcal{S}$ is a finitely generated nilpotent-by-finite group and stabilizers in $\mathrm{Aut}(X)/\mathcal{S}$ of vertices in $X/\mathcal{S}$ are finite.

Coupling Trofimov's result with the result [Bas72] that the rate of growth of a nilpotent group is always polynomial, we have the following:

F33: If the growth of a connected, locally finite, vertex-transitive graph X is not greater than that of every polynomial, then the graph X has polynomial growth, i.e., if $0 < \liminf_{k \to \infty} [\sum_{i=0}^{k} d(i)/k^n] < \infty$, then n must be an integer.

F34: [Gri85] There exist r-valent graphs for small values of r that have intermediate growth. They are *Cayley graphs* (see §6.2) of groups with the same kind of growth. These groups are finitely generated by a set of four elements and all elements have finite order, but these groups are not finitely presentable.

F35: [Se91] Finitely generated groups with intermediate growth cannot act transitively on connected locally finite graphs of polynomial growth.

EXAMPLES

E8: When $m \neq n$, the graph $K_{m,n}$ is edge-transitive but not vertex-transitive.

E9: The graph P_n of the n-sided prism is the cartesian product of an n-circuit by K_2. If $n \neq 4$, then P_n is vertex-transitive but has two edge-orbits, each consisting of the edges of one of the two n-circuits. Furthermore $|\mathrm{Aut}(P_n)| = 4n$. However, P_4 is the graph of the 3-dimensional cube; it is arc-transitive and $|\mathrm{Aut}(P_4)| = 48$.

E10: The lexicographic product of an n-circuit by the edgeless graph on k vertices is arc-transitive. Note that it has even valence $2k$ (cf. Tutte's Theorem).

E11: Let $D = D_1$ be a double ray, and let $D_{n+1} = D_n \square D$ for $n \geq 1$. Then D_n has polynomial growth of degree n.

E12: The cartesian, strong, or lexicographic product of an (infinite) r-valent tree for $r \geq 3$ with any finite connected graph has exponential growth.

E13: An infinite half-transitive graph may be constructed by taking a two-way infinite sequence of copies of Folkman's graph (see Fact 26) with bipartition $\{V_1, V_2\}$ and identifying the vertices in V_2 of the nth copy with the vertices in V_1 in the $(n+1)$st copy for $n \in \mathbb{Z}$ (cf. [ThWa89]).

REMARK

R5: For a detailed list of conditions for the existence and non-existence of semisymmetric graphs, see [Iv87].

6.1.5 s-Regularity and s-Transitivity

In the next three subsections we consider some refinements of transitivity of automorphism groups of graphs and begin by reviewing some more notions from the theory of permutation groups.

DEFINITIONS

D22: If G is a group of permutations of a set S, we say that G ***acts semiregularly*** if $G_x = \{\iota\}$ for all $x \in S$.

D23: We say that a permutation group G ***acts regularly*** or ***is regular*** if G acts both transitively and semiregularly.

D24: For $s \geq 0$, an ***s-arc*** in a graph X is a directed walk of length s in which consecutive edges are distinct.

D25: A graph X is said to be ***s-transitive***if it contains at least one s-arc and $\mathrm{Aut}(X)$ acts transitively on the set of all s-arcs. The terms *1-transitive* and *arc-transitive* are synonymous.

D26: A graph X is said to be ***s-regular*** if it contains at least one s-arc and $\mathrm{Aut}(X)$ acts regularly on the set of all s-arcs.

D27: An ***m-cage*** is a smallest 3-valent graph with girth m.

FACTS

F36: If G is regular on S, then

- for all $x, y \in S$, there is a unique $\alpha \in G$ such that $\alpha(x) = y$.
- If S is finite, then $|G| = |S|$.

F37: [Tu66] Let X be a connected s-transitive graph with no pendant vertex that is not a circuit. Then

- X is r-transitive for $1 \leq r \leq s$;
- $s \leq \frac{1}{2}\gamma(X) + 1$, where $\gamma(X)$ denotes the girth of X;
- if X is not s-regular, then X is $(s+1)$-transitive.

F38: [We74] Let X be a finite $(1 + p^r n)$-valent graph, where p is prime, $r \geq 1$, and $1 \leq n \leq p$. If $\mathrm{Aut}(X)$ contains a subgroup that acts regularly on the s-arcs of X, then $s \leq 7$ and $s \neq 6$. (This result generalizes to infinite graphs of polynomial growth. See [Se91].)

EXAMPLES

E14: A circuit is s-transitive for all $s \geq 0$.

E15: The graphs of the cube and the dodecahedron are 2-regular.

E16: The graph $K_{1,n}$ for $n \geq 1$ is 2-transitive but not 1-transitive.

E17: The bipartite graph $K_{n,n}$ is 3-transitive for $n \geq 2$ and is 3-regular if $n = 2, 3$.

E18: The complete graph K_4 is the unique 3-cage; $K_{3,3}$ is the unique 4-cage.

E19: The Petersen graph is 3-regular. It is the unique 5-cage.

E20: The unique 6-cage is the 4-regular ***Heawood graph*** H, defined as follows. Let $V(H)$ be the cyclic group $\mathbb{Z}_{14}$. For $j = 0, \ldots, 6$, let the vertex $2j$ be adjacent to the three vertices $2j-1$, , $2j+1$, and $2j+5$.

6.1.6 Graphical Regular Representations

DEFINITIONS

D28: Given a group G, a graph X such that $\text{Aut}(X)$ is isomorphic to G and acts regularly on $V(X)$ is called a ***graphical regular representation***, or ***GRR***, of G.

D29: A ***generalized dicyclic group*** is a (finite or infinite) group G with the following properties:

- G contains an abelian subgroup A of index 2;
- There exists $b \in G \backslash A$ such that $bab^{-1} = a^{-1}$ for all $a \in A$;
- There exists an element $a_0 \in A$ of order $2m$ where $m \geq 2$;
- $b^2 = a_0^m$.

The 8-element quaternion group Q is the smallest generalized dicyclic group.

FACTS

F39: If X is a GRR of G, then $|V(X)| = |G|$. It will be seen in §6.2 that X must be a Cayley graph of G.

F40: [Ch64, Sa64, Im70] The only abelian groups that admit a GRR are the elementary abelian groups of order 2^n for $n = 1$ and $n \geq 5$.

F41: [No68, Wa71] No generalized dicyclic group admits a GRR.

F42: The following ten non-abelian and non-generalized dicyclic groups do not admit a GRR:

- the dihedral groups D_n, for $n = 3, 4, 5$ [Wa71];
- Alt(4) [Wa74]);
- $Q \times \mathbb{Z}_n$ for $n = 2, 3, 4$ [Wa72];
- the nonabelian group of order 27 and exponent 3 [NoWa72b];
- the nonabelian group of order 18 and exponent 6, another group of order 16, and another group of order 24 [NoWa72a, Wa72].

F43: [He76] Every finite solvable group which is non-abelian, non-generalized dicyclic, and not one of the above ten exceptional groups admits a GRR.

F44: [Wa74] The groups $\text{Sym}(n)$ for $n \geq 4$ and $\text{Alt}(n)$ for $n \geq 5$ admit a GRR.

F45: [Go81] Every finite non-solvable group admits a GRR.

F46: [BabGo82] Let G be a nilpotent non-abelian group of odd order g. Let $\mathcal{C}(G)$ be the set of all graphs whose automorphism group contains a regular subgroup isomorphic to G. Then almost all the graphs in $\mathcal{C}(G)$ are GRR's of G, i.e., there are only $o(2^{g-1})$ exceptions as $g \to \infty$.

We close this subsection with a result about infinite GRR's.

NOTATION: If n is a positive integer, let $[n] = \{j \in \mathbb{Z} : 1 \leq j \leq n\}$.

F47: [Wa76] Let $n \geq 2$, and let $\{G_j : j \in [n]\}$ be a family of pairwise-disjoint groups such that $r(G_j) \leq \aleph_0$, where $r(G_j)$ denotes the cardinality of a smallest generating set for G_j. Then the free product $G = \coprod_{j \in [n]} G_j$ admits a GRR. If $\sum_{j \in [n]} r(G_j)$ is finite, then G admits a locally finite GRR.

6.1.7 Primitivity

DEFINITIONS

D30: Let G be a group of permutations of a set S. A subset $B \subseteq S$ is called a ***block*** (of imprimitivity with respect to G) if for all $\alpha \in G$, either $\alpha(B) = B$ or $\alpha(B) \cap B = \emptyset$.

- Clearly $\emptyset, S$, and the singleton subsets of V are blocks; they are called the ***trivial blocks***.
- If G acts transitively and admits no block other than the trivial blocks, then G is ***primitive***; if G is transitive but admits nontrivial blocks, then G is ***imprimitive*** on S.

D31: If G is imprimitive on S and B is a nontrivial block, then the set of images of B under G forms a partition of S, called a ***system of imprimitivity***.

D32: A graph is said to be ***primitive*** if its automorphism group acts as a primitive permutation group on its vertex set.

EXAMPLES

E21: An n-circuit ($n \geq 3$) is primitive if and only if n is an odd prime. If d is a proper divisor of n, then a nontrivial block is obtained by starting at any vertex v and selecting all vertices lying at distance a multiple of d from v.

E22: The complete graph K_n is primitive for all n, but $K_{n,n}$ is never primitive; the two sides of the bipartition form a system of imprimitivity of $V(K_{n,n})$.

E23: The Petersen graph is primitive.

E24: Except for the tetrahedron, the graphs of all the Platonic solids are imprimitive. A system of imprimitivity is formed by blocks of size 2, each consisiting of a vertex and its antipodal vertex.

FACTS

F48: The vertex sets of the components of a vertex-transitive graph X are blocks of $\text{Aut}(X)$. Hence primitive graphs with nonempty edge-sets are connected.

F49: In a primitive graph, the connectivity κ must equal the valence. Otherwise the graph has nontrivial atoms (see §4.2.3), and the family of vertex sets of the atoms is a system of imprimitivity.

F50: [GraWa88] Let X be a finite planar graph. Then X is primitive if and only if it is K_2, K_4, a circuit of prime length, or an edgeless graph.

We state four primitivity results for infinite graphs.

F51: [JuWa89] Let n be a nonnegative integer. There exists an infinite, locally finite, primitive graph with connectivity $\kappa = n$ if and only if $n \neq 2$.

F52: [JuWa77b] Let X be a vertex-transitive graph with connectivity $\kappa(X) = 1$. A necessary and sufficient condition for X to be primitive is that its maximal 2-connected subgraphs be primitive, have at least three vertices, and be pairwise isomorphic.

F53: [WaGra04] Let X be an infinite, locally finite planar graph. Then X is primitive if and only if $\kappa(X) = 1$, X has no separating edges, and every vertex is incident with the same number of maximal biconnected subgraphs. Moreover, either all these biconnected subgraphs are isomorphic to K_4 or they are all isomorphic p-circuit graphs C_p for some given odd prime p.

F54: [GoImSeWaWo89] Let X be a locally finite, connected, vertex-transitive graph of polynomial growth. Then X is not primitive.

6.1.8 More Automorphisms of Infinite Graphs

The overall "shape" of an infinite graph can be described effectively with the aid of the concept of the "ends" of the graph. This notion, previously used in describing infinite groups, was first applied to graphs by R. Halin [Hal64]. In this subsection, the symbol X will always denote an infinite graph.

NOTATION: Given a graph X, let $\mathcal{R}(X)$ denote the set of all rays in X.

DEFINITIONS

D33: Let $R_1, R_2 \in \mathcal{R}(X)$. We say that R_1 and R_2 are ***end-equivalent*** and write $R_1 \sim R_2$ if there exists $R_3 \in \mathcal{R}(X)$ such that both $V(R_3 \cap R_1)$ and $V(R_3 \cap R_2)$ are infinite.

It is not hard to show that $\sim$ is an equivalence relation on the set $\mathcal{R}(X)$.

D34: The equivalence classes with respect to $\sim$ are the ***ends*** of X.

NOTATION: $\mathcal{E}(X)$ will denote the set of ends of X, and we define the cardinal $\epsilon(X) = |\mathcal{E}(X)|$.

D35: If Y is a subgraph of X and $R \in \mathcal{R}(X)$, we say that the ray R ***is contained in*** the subgraph Y if Y contains a subray of R. If $R_1, R_2 \in \mathcal{R}(X)$, we say that Y ***separates*** R_1 and R_2 if these two rays are contained in distinct components of $X - Y$.

D36: A function $\alpha : V(X) \to V(X)$ is an ***endomorphism*** of the graph X if for any u, $v \in V(X)$,

$$\{u, v\} \in E(X) \Rightarrow \{\alpha(u), \alpha(v)\} \in E(X)$$

If an endomorphism is a bijection, then it is an automorphism. Endomorphisms of finite graphs are bijections.

D37: An endomorphism η of X is a ***translation*** if it fixes no finite nonempty subset of $V(X)$, i.e., $\eta(Y) = Y \Rightarrow Y$ is empty or infinite.

D38: A subgroup $G \leq \mathrm{Aut}(X)$ acts ***almost transitively*** on $V(X)$ if its action induces only finitely many orbits. We say that X is ***almost transitive*** if $\mathrm{Aut}(X)$ acts almost transitively.

D39: A ***torsion subgroup*** of an infinite group G is a subgroup all of whose elements have finite order.

FACTS

F55: **König's "Unendlichkeitslemma"** [Kö36]: Let $S_1, S_2, \ldots$ be a (countably infinite) sequence of finite, nonempty, pairwise-disjoint subsets of vertices of a graph X. Suppose that for each positive integer n, each vertex in S_{n+1} is adjacent to some vertex in S_n. Then X contains a ray with vertices $x_1, x_2, \ldots$, where $x_n \in S_n$.

F56: If a graph Y contains no ray at all, then either Y is finite or is infinite but not locally finite. In this case $\epsilon(Y) = 0$.

F57: [Hal64] Let $R_1, R_2 \in \mathcal{R}(X)$. The following statements are equivalent:

- $R_1 \sim R_2$.
- No finite subgraph of X separates R_1 and R_2.
- X contains infinitely many pairwise-disjoint (finite) R_1R_2-paths.

F58: When X is locally finite, then $\epsilon(X)$ equals the supremum of the number of infinite components of $X - S$ as S ranges over the finite subsets of $V(X)$.

Clearly $\mathrm{Aut}(X)$ induces a group of permutations of $\mathcal{E}(X)$.

F59: [Hal73] Every endomorphism of a connected graph fixes either an end or a finite nonempty subgraph.

F60: [Hal73] If an endomorphism of a connected locally finite graph fixes a finite nonempty subgraph, then it is an automorphism.

F61: Given an automorphism α of a connected locally finite graph, either all orbits of α are infinite (in which case α is a translation) or all orbits of α are finite (although they may be arbitrarily large).

F62: [Hal73] Every translation of a connected locally finite graph fixes some double ray and at most two ends.

F63: [Hal73] If X is connected and locally finite, and if $\mathrm{Aut}(X)$ contains a translation, then $\epsilon(X) = 1$, 2, or ∞.

By combining Fact 63 with a theorem of H. A. Jung [Ju81], we obtain the following important classification of almost transitive infinite graphs.

F64: If X is connected, locally finite and almost transitive, then $\epsilon(X) = 1$, 2 or $2^{\aleph_0}$.

F65: Suppose that X is almost transitive. If X has linear growth, then $\epsilon(X) = 2$ (cf. "strips" below). If X has polynomial growth of degree $d \geq 2$, then $\epsilon(X) = 1$. If X has exponential growth, then $\epsilon(X) = 1$ or $2^{\aleph_0}$.

F66: [SeTr97] If the graph X is almost transitive and has quadratic growth, then $\mathrm{Aut}(X)$ contains a subgroup isomorphic to $\mathbb{Z} \oplus \mathbb{Z}$ that acts on $V(X)$ with finitely many orbits.

F67: [SeTr97] There are only countably many nonisomorphic almost transitive graphs with linear or quadratic growth.

F68: [BabWa80] Suppose that X is connected and locally finite. If $\mathrm{Aut}(X)$ contains a torsion subgroup T that acts almost transitively on $V(X)$, then $\epsilon(X) = 1$. If T acts transitively on $V(X)$ and X is r-valent, then X has connectivity $\kappa(X) \geq \frac{3}{4}(r+1)$ and edge-connectivity $\lambda(X) = r$.

F69: [Se91] Let X be connected, locally finite, and vertex-transitive. Then $\mathrm{Aut}(X)$ is uncountable if and only if it contains a finitely generated subgroup of exponential growth that acts transitively on $V(X)$.

F70: [Hal68] Every end of a graph contains a family of pairwise disjoint rays of maximum cardinality, i.e., if an end of a graph contains arbitrarily large finite families of pairwise disjoint rays, then that end contains an infinite family of pairwise disjoint rays. In this same sense, every graph contains a family of pairwise disjoint double rays of maximum cardinality.

EXAMPLES

E25: The complete bipartite graph $K_{n,\aleph_0}$ where n is a positive integer, has zero ends, because it contains no rays. Of course, it is not locally finite. Since it has finite diameter, its growth is not defined.

E26: Let D be a double ray, and let Y be any connected graph. If Y is infinite, then the cartesian product $D \square Y$ has exactly one end. However, if Y is a finite, then $D \square Y$ has exactly two ends. In fact, $D \square Y$ is a *strip* (see below).

E27: Let $V(X) = \mathbb{Z} \times \mathbb{Z}$, and let $E(X)$ consist of all edges of the form $\{(m,n),(m,n+1)\}$ or $\{(m,0),(m+1,0)\}$ for $m,n \in \mathbb{Z}$. Then X has quadratic growth and $\epsilon(X) = \aleph_0$. Thus X is not almost transitive, by Fact 64.

E28: The cartesian or strong product with any finite graph of the (infinite) r-valent tree for $r \geq 3$ has $2^{\aleph_0}$ ends.

E29: There are three *regular tessellations* of the Euclidean plane. They may have

(i) six congruent equilateral triangles meeting at each vertex, or

(ii) four congruent squares meeting at each vertex, or

(iii) three congruent regular hexagons meeting at each vertex.

Their underlying graphs all have quadratic growth and exactly one end.

E30: The regular tessellations of the hyperbolic plane (e.g., four conguent pentagons meeting at every vertex) also have exactly one end, but their growth is exponential.

Strips

A special class of 2-ended graphs is of some interest.

DEFINITIONS

D40: A connected graph X is called a ***strip*** if there exists a connected subgraph Y of X and an automorphism $\alpha \in \mathrm{Aut}(X)$ such that $\partial(Y)$ and $Y - \alpha(Y)$ are finite and $\alpha(Y \cup \partial(Y)) \subseteq Y$.

D41: The ***infinite connectivity*** of X, denoted $\kappa_\infty(X)$, is the minimum cardinality of a set $S \subset V(X)$ such that $X - S$ has at least two infinite components.

REMARK

R6: In §4.2.3, the notions of *fragment* and *atom* are presented with respect to the connectivity κ of a graph. These terms may also be defined with respect to the more restrictive parameter of infinite connectivity with essentially identical results. In particular, distinct κ_∞-atoms are disjoint. (See [JuWa77a].)

FACTS

F71: For a connected infinite graph X, the following statements are equivalent:

- X is a strip;
- X is locally finite and $\mathrm{Aut}(X)$ contains an automorphism with finitely many orbits [JuWa84];
- X is locally finite, $\epsilon(X) = 2$, and $\mathrm{Aut}(X)$ contains a translation [ImSe88].

F72: [ImSe88] Let X be connected, locally finite, and vertex-transitive. Then X has linear growth if and only if X is a strip.

F73: [JuWa84] Suppose that X is connected and that $\mathrm{Aut}(X)$ contains an abelian subgroup H that acts transitively on $V(X)$. Then either

- $\kappa_\infty(X) = \infty$, i.e., $\epsilon(X) = 1$, or
- X is a strip and $H \cong \mathbb{Z} \oplus F$ for some finite abelian group F.

F74: [Wa91] If a strip is edge-transitive, then all vertices have even valence.

F75: [Wa91] Let S be a planar edge-transitive strip with connectivity $k \geq 3$. Then $V(S) = (\mathbb{Z} \times \mathbb{Z})/\rho$ where $\rho = \{\{(x, -x), (-x+k, x+k)\} : x \in \mathbb{Z}\}$ and the vertex $(x, y)\rho$ is adjacent to $(x, y \pm 1)\rho$ and $(x \pm 1, y)\rho$ for all $(x, y) \in \mathbb{Z} \times \mathbb{Z}$.

The next result says that strips can be found as subgraphs of multi-ended graphs.

F76: [Ju94] Let X be locally finite and suppose that a subgroup $G \leq \mathrm{Aut}(X)$ fixes some 2-subset of $\mathcal{E}(X)$. If G contains a translation, then there is a G-invariant induced subgraph S of X that is a strip (with respect to G), and ∂Y is finite for every component Y of $X - S$.

Some Results Involving Distance

DEFINITIONS

D42: An automorphism $\alpha \in \mathrm{Aut}(X)$ is ***bounded*** if there exists $M > 0$ such that for all $v \in V(X)$, the distance $d(v, \alpha(v)) < M$.

D43: A path or ray or double ray is said to be ***geodetic*** if it contains a shortest path joining any two of its vertices . A geodetic double ray is called a ***geodesic***.

D44: Let D be a ray or double ray in X. The ***straightness*** $\sigma(D)$ of D is defined to be

$$\sigma(D) = \liminf_{d_D(u,v)\to\infty} \frac{d(u,v)}{d_D(u,v)}$$

where $u, v \in V(D)$ and $d_D(u,v)$ is the length of the subpath of D joining u and v.

D45: A ray or double ray R is said to be ***metric*** if $\sigma(R) > 0$.

D46: If $\alpha \in \mathrm{Aut}(X)$, then a ray R is ***α-essential*** if $\alpha^n(R) \subset R$ for some positive integer n, and α is ***of metric type*** if there exists a metric α-essential ray.

FACTS

F77: [JuWa84] The set of bounded automorphisms of a graph Y forms a normal subgroup of $\mathrm{Aut}(Y)$.

The following result extends König's "Unendlichkeitslemma", stated here in weaker form for locally finite graphs:

F78: [PoWa95] For each vertex $u \in V(X)$ and for each end of X, there exists a metric ray belonging to that end that originates at u.

F79: $0 \leq \sigma(D) \leq 1$ for any double ray D. If D is a geodesic, then $\sigma(D) = 1$, but not conversely.

F80: [PoWa95] If a translation $\tau \in \mathrm{Aut}(X)$ fixes some metric double ray, then every double ray fixed by a nonzero power of τ is also metric.

F81: [PoWa95] Suppose that a translation $\tau \in \mathrm{Aut}(X)$ fixes some metric double ray D_0. If $\sigma(D_0) < 1$, then some power of τ fixes a metric double ray D_1 such that $\sigma(D_1) > \sigma(D_0)$. If $\sigma(D_0) = 1$, then D_0 is a geodesic.

F82: [JuNi94] If $\tau \in \mathrm{Aut}(X)$ is a translation of metric type, then

$$\sup\{\sigma(R) : R \text{ is } \tau\text{−essential}\} = 1$$

EXAMPLE

E31: Let $V(X) = \mathbb{Z}_m \times \mathbb{Z}$ where each vertex (x,y) is adjacent to vertices $(x \pm 1, y)$ and $(x, y \pm 1)$, the first coordinate being read modulo m. Let $\tau(x,y) = (x+1, y+1)$ for all $(x,y) \in V(X)$. Let D be the double ray with edges of the forms $\{(x,x),(x,x+1)\}$ and $\{(x,x+1),(x+1,x+1)\}$. Then $\sigma(D) = \frac{1}{2}$, and so D is metric. Note that all rays with vertex set $\{(x,y) : y \in \mathbb{Z}\}$ are τ-essential, as they are fixed by τ^m.

References

[Bab74] L. Babai, On the minimum order of graphs with given group, *Canad. Math. Bull.* **17** (1974), 467–470.

[BabGo82] L. Babai and C. D. Godsil, On the automorphism groups of almost all Cayley graphs, *Europ. J. Combin.* **3** (1982), 9–15.

[BabWa80] L. Babai and M. E. Watkins, Connectivity of infinite graphs having a transitive torsion group action, *Arch. Math.* **34** (1980), 90–96.

[Bas72] H. Bass, The degree of polynomial growth of finitely generated nilpotent groups, *Proc. London Math. Soc. (3)* **25** (1972), 603–614.

[Bo70] I. Z. Bouwer, Vertex and edge transitive but not 1-transitive graphs, *Canad. Math. Bull.* **13** (1970), 231–237.

[Ch64], C.-Y. Chao, On a theorem of Sabidussi, *Proc. Amer. Math. Soc.* **15** (1964), 291–292.

[Fo67] J. Folkman, Regular line-symmetric graphs, *J. Combin. Theory* **3** (1967), 215–232.

[Fr37] R. Frucht, Die Gruppe des Petersen'schen Graphen und der Kantensysteme der regulären Polyheder, *Comment. Math. Helvetici* **9** (1936/37), 217–223.

[Fr38] R. Frucht, Herstellung von Graphen mit vorgegebener abstrakter Gruppe, *Compositio Math.* **6** (1938), 239–250.

[Fr49] R. Frucht, Graphs of degree 3 with given abstract group, *Canad. J. Math.* **1** (1949), 365–378.

[FrGraWa71] R. Frucht, J. E. Graver, and M. E. Watkins, The groups of the generalized Petersen graphs, *Proc. Cambridge Phil. Soc.* **70** (1971), 211–218.

[Go81] C. D. Godsil, GRR's for non-solvable groups, in: *Algebraic Methods in Graph Theory* (Proc. Conf. Szeged 1978), L. Lovász and V. T. Sós, eds.), Colloq. Soc. János Bolyai **25**, North-Holland, Amsterdam, 1981, pp. 221–239.

[GoImSeWaWo89] C. D. Godsil, W. Imrich, N. Seifter, M. E. Watkins, and W. Woess, On bounded automorphisms of infinite graphs, *Graphs and Combin.* **5** (1989), 333–338.

[GraWa88] J. E. Graver and M. E. Watkins, A characterization of finite planar primitive graphs, *Scientia* **1** (1988), 59–60.

[Gri85] R. I. Grigorchuk, Degrees of growth of finitely generated groups and the theory of invariant means, *Math. USSR-Izv* **25** (1985), 259–300.

[Hal64] R. Halin, Über unendliche Wege in Graphen, *Math. Ann* **157** (1964), 125–137.

[Hal68] R. Halin, Die Maximalzahl fremder zweiseitig unendlicher Wege in Graphen, *Math. Nachr.* **44** (1968), 119–127.

[Hal73] R. Halin, Automorphisms and endomorphisms of infinite locally finite graphs, *Abh. Math. Sem. Univ. Hamburg* **39** (1973), 251–283.

[HarPa68] F. Harary and E. M. Palmer, On the point-group and line-group of a graph, Acta Math. Acad. Sci. Hungar. **19** (1968), 263–269.

[He76] D. Hetzel, Über reguläre Darstellung von auflösbaren Gruppen, *Dipomarbeit, Technische Universität Berlin*, 1976.

[Ho81] D. F. Holt, A graph which is edge transitive but not arc transitive, *J. Graph Theory* **5** (1981), 201–204.

[Im69] W. Imrich, Über das lexikographische Produkt von Graphen, *Arch. Math. (Basel)* **20** (1969), 228–234.

[Im70] W. Imrich, Graphs with transitive Abelian automorphism group, in: *Combinatorial Theory and Its Applications* (Colloq. Math. Soc. János Bolyai **4** Proc. Colloq. Balatonfűred, Hungary 1969), P. Erdős, A. Renyi, and V. T. Sós eds., North-Holland, Amsterdam, 1970, pp. 651–656.

[Im72] W. Imrich, Assoziative Produkte von Graphen, *Österreich. Akad. Wiss. Math.-Natur. K. S.-B. II*,**180** (1972), 203–293.

[ImIz75] W. Imrich and H. Izbicki, Associative products of graphs, *Monatsh. Math.* **80** (1975), 277–281.

[ImKl00] W. Imrich and S. Klavžar, *Product Graphs, Structure and Recognition*, John Wiley & Sons, Inc., New York, 2000.

[ImSe88] W. Imrich and N. Seifter, A note on the growth of transitive graphs, *Discrete Math.* **73** (1988/89), 111–117.

[Iv87] A. V. Ivanov, On edge but not vertex transitive regular graphs, *Annals Discrete Math.* **34** (1987), 273–286.

[Ju81] H. A. Jung, A note on fragments of infinite graphs, *Combinatorica* **1** (1981), 285–288.

[Ju94] H. A. Jung, On finite fixed sets in infinite graphs, *Discrete Math.* **131** (1994), 115–125.

[JuNi94] H. A. Jung and P. Niemeyer, Decomposing ends of locally finite graphs, *Math. Nachr.* **174** (1995), 185–202.

[JuWa77a] H. A. Jung and M. E. Watkins, On the connectivities of finite and infinite graphs, *Monatsh. Math.* **83** (1977), 121–131.

[JuWa77b] H. A. Jung and M. E. Watkins, On the structure of infinite vertex-transitive graphs, *Discrete Math.* **18**, 45–53.

[JuWa84] H. A. Jung and M. E. Watkins, Fragments and automorphisms of infinite graphs, *Europ. J. Combinatorics* **5** (1984), 149–162.

[JuWa89] H. A. Jung and M. E. Watkins, The connectivities of locally finite primitive graphs, *Combinatorica* **9** (1989), 261–267.

[Kö36] D. König, *Theorie der endlichen und undendichen Graphen*, Akad. Verlagsgesellschaft, Leipzig, 1936.

[No68] L. A. Nowitz, On the non-existence of graphs with transitive generalized dicyclic groups, *J. Combin. Theory* **4** (1968), 49–51.

[NoWa72a] L. A. Nowitz and M. E. Watkins, Graphical regular representations of non-abelian groups, I, *Canad. J. Math.* **14** (1972), 993–1008.

[NoWa72b] L. A. Nowitz and M. E. Watkins, Graphical regular representations of non-abelian groups, II, *Canad. J. Math.* **14** (1972), 1009–1018.

[PoWa95] N. Polat and M. E. Watkins, On translations of double rays in graphs, *Per. Math. Hungar.* **30** (1995), 145–154.

[Sa57] G. Sabidussi, Graphs with given group and given graph-theoretical properties, *Canad. J. Math.* **9** (1957), 515–525.

[Sa60] G. Sabidussi, Graph multiplication, *Math. Zeitschr.* **72** (1960), 446–457.

[Sa61] G. Sabidussi, The lexicographic product of graphs, *Duke Math. J.* **28** (1961), 573–578.

[Sa564] G. Sabidussi, Vertex-transitive graphs, *Monatsh. Math.* **68** (1964), 426–428.

[Sa67] G. Sabidussi, Review #2563, *Math. Rev.* **33**, No. 3, March 1967.

[Se91] N. Seifter, Properties of graphs with polynomial growth, *J. Combin. Theory Ser. B* **52** (1991), 222–235.

[SeTr97] N. Seifter and V. I. Trofimov, Automorphism groups of graphs with quadratic growth, *J. Combin. Theory Ser. B* **71** (1997), 205–210.

[ThWa89] C. Thomassen and M. E. Watkins, Infinite vertex-transitive, edge-transitive, non 1-transitive graphs, *Proc. Amer. Math. Soc.* **105** (1989), 258–261.

[Tr85] V. I. Trofimov, Graphs with polynomial growth, *Math. USSR Sbornik* **51** (1985), No. 2, 404–417.

[Tu66] W. T. Tutte, *Connectivity in Graphs*, University of Toronto Press, Toronto, 1966.

[Wa71] M. E. Watkins, On the action of non-Abelian groups on graphs, *J. Combin. Theory* **11** (1971), 95–104.

[Wa72] M. E. Watkins, On graphical regular representations of $C_n \times Q$, in: *Graph Theory and Its Applications*, (Y. Alavi, D. R. Lick, and A. T. White, eds.) Springer-Verlag, Berlin, 1972, pp. 305–311.

[Wa74] M. E. Watkins, Graphical regular representations of alternating, symmetric, and miscellaneous small groups, *Aequat. Math.* **11** (1974), 40–50.

[Wa76] M. E. Watkins, Graphical regular representations of free products of groups, *J. Combin. Theory* **21** (1976), 47–56.

[Wa91] M. E. Watkins, Edge-transitive strips, *J. Combin. Theory Ser. B* **95** (1991), 350–372.

[WaGra04] M. E. Watkins and J. E. Graver, A characterization of infinite planar primitive graphs, *J. Combin. Theory Ser. B*, to appear.

[We74] R. M. Weiss, Über s-reguläre Graphen, *J. Combin. Theory Ser. B* **16** (1974), 229–233.

[Wh32] H. Whitney, Congruent graphs and the connectivity of graphs, *Amer. J. Math.* **54** (1932), 150–168.

6.2 CAYLEY GRAPHS

Brian Alspach, University of Regina, Canada

6.2.1 Construction and Recognition
6.2.2 Prevalence
6.2.3 Isomorphism
6.2.4 Subgraphs
6.2.5 Factorization
6.2.6 Further Reading
References

Introduction

There are frequent occasions for which graphs with a lot of symmetry are required. One such family of graphs is constructed using groups. These graphs are called Cayley graphs and are the subject of this chapter.

6.2.1 Construction and Recognition

We restrict ourselves to finite graphs, which means we use finite groups, but the basic construction is the same for infinite groups. While Cayley graphs on finite groups and Cayley graphs on infinite groups share a variety of features, there are aspects of Cayley graphs on finite groups that do not carry over to Cayley graphs on infinite groups, and vice versa.

DEFINITIONS

D1: Let $\mathcal{G}$ be a finite group with identity 1. Let $\mathcal{S}$ be a subset of $\mathcal{G}$ satisfying $1 \notin \mathcal{S}$ and $\mathcal{S} = \mathcal{S}^{-1}$, that is, $s \in \mathcal{S}$ if and only if $s^{-1} \in \mathcal{S}$. The ***Cayley graph*** on $\mathcal{G}$ with ***connection set*** $\mathcal{S}$, denoted $\mathrm{Cay}(\mathcal{G};\mathcal{S})$, satisfies:

- the vertices of $\mathrm{Cay}(\mathcal{G};\mathcal{S})$ are the elements of $\mathcal{G}$;
- there is an edge joining $g, h \in \mathrm{Cay}(\mathcal{G};\mathcal{S})$ if and only if $h = gs$ for some $s \in \mathcal{S}$.

We do not require that the connection set $\mathcal{S}$ generate the group $\mathcal{G}$. It is standard to use additive notation when $\mathcal{G}$ is an abelian group and multiplicative notation for nonabelian groups. Thus, for abelian groups, we have $\mathcal{S} = -\mathcal{S}$ and $h = g + s$.

NOTATION: The set of all Cayley graphs on $\mathcal{G}$ is denoted $\mathrm{Cay}(\mathcal{G})$.

D2: Cayley graphs on the cyclic group $\mathbb{Z}_n$ are called ***circulant graphs***. We use the special notation $\mathrm{Circ}(n;\mathcal{S})$ for a circulant graph on $\mathbb{Z}_n$ with connection set $\mathcal{S}$.

NOTATION: g_L denotes the permutation on the group $\mathcal{G}$ given by the rule $g_L(h) = gh$.

NOTATION: $\mathcal{G}_L$ denotes the group $\{g_L : g \in \mathcal{G}\}$.

D3: An ***automorphism of a simple graph*** G is a bijection f on the vertex set $V(G)$ such that $\langle u, v\rangle$ is an edge if and only if $\langle f(u), f(v)\rangle$ is an edge.

NOTATION: The group of all automorphisms of a graph G is denoted $\mathrm{Aut}(G)$.

D4: A graph G is said to be ***vertex-transitive*** if $\mathrm{Aut}(G)$ acts transitively on $V(G)$.

D5: Let $\mathcal{G}$ be a transitive permutation group acting on a finite set Ω. If $\mathcal{G}$ satisfies any one of the following three equivalent conditions, then it is said to be ***regular***:

- the only element of $\mathcal{G}$ fixing an element of Ω is the identity permutation;
- $|\mathcal{G}|=|\Omega|$;
- for any $\omega_1, \omega_2 \in \Omega$, there is a unique element $g \in \mathcal{G}$ satisfying $\omega_1 g = \omega_2$.

EXAMPLES

E1: The hypercube Q_n may be realized as a Cayley graph on the elementary abelian 2-group $\mathbb{Z}_2^n$ using the standard generators $e_1, e_2, \ldots, e_n$ for the connection set, where e_i has a 1 in the i-th coordinate and zeroes elsewhere. The hypercube Q_n may be realized in other ways as a Cayley graph, but the realization just given is the common one.

E2: The complete graph K_n is representable as a Cayley graph on any group $\mathcal{G}$ of order n, where the connection set is the set of non-identity elements of the group. We get the complement of K_n by using the empty set as the connection set.

E3: The complete multipartite graph $K_{m;n}$, with m parts each of cardinality n, is realizable as a circulant graph of order mn, with the connection set being all the elements not congruent to zero modulo n.

E4: The graph formed on the finite field $GF(q)$, $q \equiv 1 (\mathrm{mod}\ 4)$, where the connection set is the set of quadratic residues in $GF(q)$, is called a ***Paley graph***.

E5: The circulant graph of even order n with connection set $S = \{\pm 1, n/2\}$ is known as the ***Möbius ladder*** of order n.

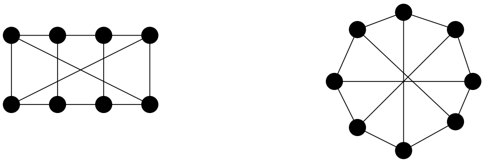


Figure 6.2.1 **Two drawings of the Möbius ladder of order 8.**

FACTS

F1: Every Cayley graph is vertex-transitive.

F2: The Cayley graph $\mathrm{Cay}(\mathcal{G}; \mathcal{S})$ is connected if and only if $\mathcal{S}$ generates $\mathcal{G}$.

F3: [Sa58] **Sabidussi's Theorem**: A graph G is a Cayley graph if and only if $\mathrm{Aut}(G)$ contains a regular subgroup.

REMARKS

R1: Sabidussi's Theorem above is the basis for all work on recognizing whether or not an arbitrary graph is a Cayley graph. It is an absolutely fundamental result.

R2: The Cayley graphs on the group $\mathcal{Z}_\ell^n$ with the standard generators of the group as connection set are used as theoretical models of interconnection networks of homogeneous processors in computer science.

6.2.2 Prevalence

The family of Cayley graphs provides us with a straightforward construction for vertex-transitive graphs. A natural question to pose is whether or not the family of Cayley graphs encompasses all finite vertex-transitive graphs. The Petersen graph is the smallest vertex-transitive graph that is not a Cayley graph, and it suggests the topic of this section.

NOTATION: $\mathbb{NC}$ denotes the set of integers n for which there exists a non-Cayley vertex-transitive graph of order n.

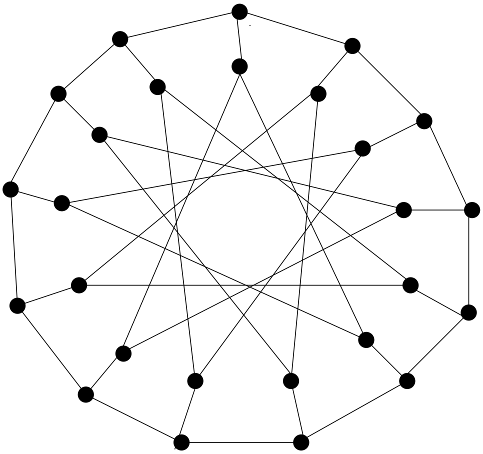


Figure 6.2.2 A non-Cayley vertex-transitive graph of order 26.

EXAMPLE

E6: If $n \in \mathbb{NC}$, then any multiple of n belongs to $\mathbb{NC}$. This follows by taking the appropriate number of vertex-disjoint copies of a non-Cayley, vertex-transitive graph of order n. Thus, in order to determine $\mathbb{NC}$, it suffices to find the minimal elements belonging to $\mathbb{NC}$.

FACTS

The first two facts reduce the problem of trying to characterize membership in $\mathbb{NC}$ to the consideration of square-free integers.

F4: A prime power $p^e \in \mathbb{NC}$ if and only $e \geq 4$.

F5: Any positive integer, other than 12, divisible by a square is in $\mathbb{NC}$.

F6: Let p and q be distinct primes with $p < q$. Then $pq \in \mathbb{NC}$ if and only if one of the following holds:

- p^2 divides $q - 1$;
- $q = 2p - 1 > 3$ or $q = \frac{p^2+1}{2}$;
- $q = 2^t + 1$ and either p divides $2^t - 1$ or $p = 2^{t-1} - 1$;
- $q = 2^t - 1$ and $p = 2^{t-1} + 1$; and
- $p = 7$, $q = 11$;

F7: Let p and q be odd primes satisfying $p < q$. Then $2pq \in \mathbb{NC}$ if and only if one of the following holds:

- p^2 divides $q - 1$;
- $p \equiv 1 (\text{mod } 4)$ or $q \equiv 1 (\text{mod } 4)$;
- $p = 7$, $q = 11$;
- $p \equiv q \equiv 3 (\text{mod } 4)$, p divides $q - 1$, and p^2 does not divide $q - 1$;
- $p \equiv q \equiv 3 (\text{mod } 4)$, and $p = \frac{q+1}{4}$; and
- $p = 7$, $q = 19$.

F8: Let p, q, r be distinct odd primes satisfying $p < q < r$. Then $pqr \in \mathbb{NC}$ if and only if at least one of pq, pr or qr is a member of $\mathbb{NC}$, or none of pq, pr and qr is a member of $\mathbb{NC}$ but one of the following holds:

- $pqr = (2^{2^t} + 1)(2^{2^{t+1}} + 1)$, for some t;
- $pqr = (2^{d\pm1} + 1)(2^d - 1)$, for some prime d;
- $pq = 2r \pm 1$ or $pq = (r + 1)/2$;
- $pq = (r^2 + 1)/2$ or $pr = (q^2 + 1)/2$;
- $pq = (r^2 - 1)/24x$ or $pr = (q^2 - 1)/24x$, where $x \in \{1, 2, 5\}$;
- $ab = 2^t + 1$ and c divides $2^t - 1$, where $\{a, b, c\} = \{p, q, r\}$;
- the largest power of p dividing $q - 1$ is p^p and the largest power of q dividing $r - 1$ is q^q;
- $q = (3p + 1)/2$ and $r = 3p + 2$, or $q = 6p - 1$ and $r = 6p + 1$;
- $q = (r - 1)/2$ and p divides $r + 1$, where $p > q$ when $p = (r + 1)/2$;
- $p = (k^{d/2} + 1)/(k + 1)$, $q = (k^{d/2} - 1)/(k - 1)$, $r = (k^{d-1} - 1)/(k - 1)$, where $k, d - 1, d/2$ are primes and $p > q$ may be the case;
- $p = (k^{(d-1)/2} + 1)/(k + 1)$, $(k^{(k-1)/2} - 1)/(k - 1)$, $r = (k^d - 1)/(k - 1)$, where $k, d, (d - 1)/2$ are primes and $p > q$ may be the case;
- $p = k^2 - k + 1$, $q = (k^5 - 1)/(k - 1)$, $r = (k^7 - 1)/(k - 1)$, where k is prime;
- $p = 3$, $q = (2^d + 1)/3$, $r = 2^d - 1$, where d is a prime;
- $p = (2^d + 1)/3$, $q = 2^d - 1$, $r = 2^{2d\pm2} + 1$, where $d = 2^t \pm 1$ is prime;
- $p = 5$, $q = 11$ and $r = 19$; and
- $p = 7$, $q = 73$ and $r = 257$.

RESEARCH PROBLEM

P1: Is there a number $k > 0$ such that every product of k distinct primes is in $\mathbb{NC}$? There is no known characterization of the members of $\mathbb{NC}$ that are products of four distinct primes.

6.2.3 Isomorphism

Some of the most interesting and deepest work on Cayley graphs has revolved around the question of trying to determine when two Cayley graphs are isomorphic.

DEFINITIONS AND NOTATIONS

D6: A Cayley graph $\mathrm{Cay}(\mathcal{G};\mathcal{S})$ is a ***CI-graph*** if whenever $\mathrm{Cay}(\mathcal{G};\mathcal{S}) \cong \mathrm{Cay}(\mathcal{G};\mathcal{S}')$, there exists an automorphism $\alpha \in \mathcal{A}\mathrm{ut}(\mathcal{G})$ such that $\mathcal{S}' = \alpha(\mathcal{S})$.

D7: A group $\mathcal{G}$ is a ***CI-group*** if every Cayley graph on $\mathcal{G}$ is a CI-graph.

D8: A ***unit in a ring*** is an element with a multiplicative inverse.

NOTATION: Let $\mathbb{Z}_n^*$ denote the multiplicative group of units in the ring $\mathbb{Z}_n$.

D9: Let $n = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$ be the factorization of n into a product of distinct prime powers. For any integer m such that $0 \le m < n$, the ***residue type***

$$m(n) = (m_1, m_2, \ldots, m_t)$$

is the unique t-tuple such that $m_i \equiv m (\mathrm{mod}\ p_i^{e_i})$ and $0 \le m_i < p_i^{e_i}$, $i = 1, 2, \ldots t$.

NOTATION: Let $\mathcal{T}(n) = \{m(n) : 0 \le m < n\}$. For a subset $R \subseteq \{1, 2, \ldots, t\}$, let $\mathcal{T}_R(n)$ denote the elements of $\mathcal{T}(n)$ for which $m_i \ne 0$ if and only if $i \in T$. When the value n under discussion is clear, we use the notation $\mathcal{T}$ and $\mathcal{T}_R$.

D10: Let $n = p_1^{e_1} p_2^{e_2} \cdots p_t^{e_t}$ be the factorization of n into a product of distinct prime powers. Then the ***order-type*** of the t-tuple $a = (a_1, a_2, \ldots, a_t) \in \mathbb{Z}_n^*$is the t-tuple

$$(d_1, d_2, \ldots, d_t), \quad \text{where } d_i \text{ is the order of } a_i \text{ in } \mathbb{Z}_{p_i^e}^*$$

Note that $a \in \mathbb{Z}_n^*$ implies $a_i \ne 0$, $i = 1, 2, \ldots, t$.

NOTATION: Let $(d_1, d_2, \ldots, d_t)$ be the order-type of some unit $a \in \mathbb{Z}_n^*$, and let $R \subseteq \{1, 2, \ldots, t\}$. If each d_i, $i \in R$, has the form $d_i = 2^e b$, where $e \ge 1$ and b is odd, then let $\mathrm{lcm}^*(R) = \mathrm{lcm}(R)/2$. In all other cases, let $\mathrm{lcm}^*(R) = \mathrm{lcm}(R)$, where $\mathrm{lcm}(R)$ denotes the least common multiple of the d_i terms in the coordinates corresponding to the elements of R.

EXAMPLES

E7: The two circulant graphs $\mathrm{Circ}(7; 1, 2, 5, 6)$ and $\mathrm{Circ}(7; 1, 3, 4, 6)$ are isomorphic via the mapping that takes g to $3g$ for all elements of $\mathbb{Z}_7$. This illustrates Definition 6 above, because this mapping is an automorphism of $\mathbb{Z}_7$.

E8: For $n = 25$, let

$$[\mathcal{S} = \{1, 4, 5, 6, 9, 11, 14, 16, 19, 20, 21, 24\}]$$

and

$$\left[\mathcal{S}' = \{1, 4, 6, 9, 10, 11, 14, 15, 16, 19, 21, 24\}\right]$$

The two circulant graphs $\mathrm{Circ}(25; \mathcal{S})$ and $\mathrm{Circ}(25; \mathcal{S}')$ are isomorphic since both are wreath products of a 5-cycle with a 5-cycle. On the other hand, it is easy to see there is no $a \in \mathbb{Z}_{25}^*$ for which $\mathcal{S}' = a\mathcal{S}$ is satisfied. Thus, $\mathbb{Z}_{25}$ is not a CI-group.

FACTS

F9: [Ba77] Let G be a Cayley graph on the finite group $\mathcal{G}$. Then G is a CI-graph if and only if all regular subgroups of $\mathrm{Aut}(G)$ isomorphic to G are conjugate in $\mathrm{Aut}(G)$.

F10: [Li02] If $\mathcal{G}$ is a CI-group, then $\mathcal{G}$ is solvable.

F11: [Mu97] The cyclic group $\mathbb{Z}_n$ is a CI-group if and only $n = 2^e m$, where m is odd and square-free and $e \in \{0, 1, 2\}$, or $n \in \{8, 9, 18\}$.

F12: Let p a prime. The elementary abelian p-groups $\mathbb{Z}_p^e$ are CI-groups for $1 \leq e \leq 4$. On the other hand, for $e \geq 2p - 1 + \binom{2p-1}{p}$, $\mathbb{Z}_p^e$ is not a CI-group. The latter inequality is sharp for $p = 2$.

One motivation for classifying CI-groups is that the Cayley graphs on a CI-group may be enumerated in a straightforward way using Pólya's enumeration theorem [Br64]. The next two theorems illustrate this for circulant graphs.

F13: [Tu67] If p is an odd prime, then the number of isomorphism classes of vertex-transitive graphs of order p is

$$\left[\frac{2}{p-1}\sum_{d}\Phi(d)2^{(p-1)/2d},\right]$$

where the summation runs over all divisors d of $(p-1)/2$ and Φ denotes the Euler totient function.

F14: [AlMi02] If $n = p_1 p_2 \cdots p_t$ is a product of distinct primes, or if $p_1 = 4$ and $p_2, p_3, \ldots, p_t$ are distinct odd primes, then the number of isomorphism classes of circulant graphs of order n is

$$\left[\frac{1}{\Phi(n)}\sum_{(d_1, d_2, \ldots, d_t)}\Phi(d_1)\Phi(d_2)\cdots\Phi(d_t)\prod_{R}2^{|\mathcal{T}_R|/2\mathrm{lcm}^*(R)},\right]$$

where the sum is taken over all possible order types of $a \in \mathbb{Z}_n^*$, and the product is taken over all non-empty subsets R of $\{1, 2, \ldots, t\}$ unless $p_1 = 4$, in which case $R = \emptyset$ is included and $|\mathcal{T}_\emptyset|$ is taken to be 2.

EXAMPLE

E9: We illustrate Fact 14 for $n = 20$. The possible order-types are (1,2), (1,2), (1,4), (2,1), (2,2) and (2,4). There are, for example, two automorphisms of order type (1,4). Since $p_1 = 4$, the term corresponding to $R = \emptyset$ appears in the product, and the contribution is 2 since $|\mathcal{T}_\emptyset| = 2$ and we consider $\mathrm{lcm}^*(\emptyset) = 1$. For $R = \{1\}$, we have $\mathrm{lcm}^*(R) = 1$ and $|\mathcal{T}_R| = 2$ making a contribution of 2 to the product. For $R = \{2\}$, we have $\mathrm{lcm}^*(R) = 2$ and $|\mathcal{T}_R| = 8$, thereby contributing 2^2. Finally, when $R = \{1, 2\}$, we have $\mathrm{lcm}^*(R) = 4$ and $|\mathcal{T}_R| = 8$ so the contribution is 2. Altogether the term in the product is 2^5. We do the same thing for all possible order-types and find that there are 336 non-isomorphic circulant graphs of order 20.

RESEARCH PROBLEM

P2: For an odd prime p, determine the values of e for which $\mathbb{Z}_p^e$ is a CI-group.

6.2.4 Subgraphs

There are interesting results and questions regarding subgraphs of Cayley graphs. Some of the results we mention hold for all vertex-transitive graphs, and we state them accordingly. It is not always clear just what impact vertex-transitivity has on the existence of certain subgraphs.

DEFINITIONS

D11: A connection set $\mathcal{S}$ is said to be ***quasi-minimal*** if the elements of $\mathcal{S}$ can be ordered $s_1, s_2, \ldots, s_t$ so that

- if $|s_i| > 2$, then s_i^{-1} is either s_{i-1} or s_{i+1}, and
- if $\mathcal{S}_i$ denotes the prefix $\{s_1, s_2, \ldots, s_i\}$, then for each i such that $|s_i| = 2$, the group $\langle \mathcal{S}_i \rangle$ is a proper supergroup of $\langle \mathcal{S}_{i-1} \rangle$, and for each i such that $|s_i| > 2$ and $s_i^{-1} = s_{i-1}$, the group $\langle \mathcal{S}_i \rangle$ is a proper supergroup of $\langle \mathcal{S}_{i-2} \rangle$.

D12: A graph G is ***Hamilton-connected*** if for any two vertices u, v of G, there is a Hamilton path whose terminal vertices are u and v.

TERMINOLOGY NOTE: In some sections of the *Handbook* the term *hamiltonian-connected* is used instead.

D13: A bipartite graph G with parts A and B is ***Hamilton-laceable*** if for any $u \in A$ and $v \in B$, there is a Hamilton path whose terminal vertices are u and v.

FACTS

F15: Let G be a connected vertex-transitive graph. If G has even order, then it has a 1-factor. If G has odd order, then $G - v$ has a 1-factor for every vertex $v \in G$.

F16: If a d-regular graph G is connected and vertex-transitive, then G is d-edge-connected.

F17: [Ma71, Wa70] If a d-regular G is connected and vertex-transitive, then the vertex-connectivity of G exceeds $\frac{2d}{3}$. Furthermore, for every $\epsilon > 0$, there exists a connected vertex-transitive graph H whose vertex connectivity is smaller than $\epsilon + \frac{2d}{3}$.

F18: [Al92] If $\mathcal{S}$ is a quasi-minimal generating set of the group $\mathcal{G}$, then the Cayley graph $\mathrm{Cay}(\mathcal{G}; \mathcal{S})$ has connectivity $|\mathcal{S}|$.

F19: For every positive integer m, there exists a Paley graph containing all graphs of order m as induced subgraphs.

F20: [ChQi81] Let G be a connected Cayley graph on a finite abelian group. If G is bipartite and has degree at least 3, then G is Hamilton-laceable. If G is not bipartite and has degree at least 3, then G is Hamilton-connected.

F21: [Wi84] Every connected Cayley graph on a group of order p^e, p a prime and $e \geq 1$, has a Hamilton cycle.

6.2.5 Factorization

DEFINITIONS

D14: A 1-***factorization*** of a graph is a partition of the edge set into 1-factors.

D15: The connection set $\mathcal{S}$ is a ***minimal generating Cayley set*** for the group $\mathcal{G}$ if $\mathcal{S}$ generates $\mathcal{G}$, but $\mathcal{S} - \{s, s^{-1}\}$ generates a proper subgroup for every $s \in \mathcal{S}$.

D16: A ***Hamilton decomposition*** of a graph G is a partition of the edge set into Hamilton cycles when the degree is even, or a partition into Hamilton cycles and a 1-factor when the degree is odd.

D17: An ***isomorphic factorization*** of a graph G is a partition of the edge set of G so that the subgraphs induced by the edges in each part are pairwise isomorphic.

FACTS

F22: [St85] Every connected Cayley graph on the group $\mathcal{G}$ has a 1-factorization if one of the following holds:

- $|\mathcal{G}| = 2^k$ for an integer k;
- $\mathcal{G}$ is an even order abelian group; or
- $\mathcal{G}$ is dihedral or dicyclic.

F23: [St85] Cayley graphs whose connection sets are minimal generating Cayley sets have 1-factorizations whenever the group is one of the following:

- an even order nilpotent group;
- the group contains a proper abelian normal subgroup of index 2^k; or
- the group has order $2^m p^k$ for a prime p satisfying $p > 2^m$.

F24: If the automorphism group of a cubic Cayley graph G has a solvable subgroup that acts transitively on the vertex set of G, then G has a 1-factorization.

F25: [Li96, Li03ta] If $G = \mathrm{Cay}(\mathcal{G}, \mathcal{S})$ is a connected Cayley graph on an abelian group $\mathcal{G}$ and $\mathcal{S}$ is a minimal generating Cayley set, then G has a Hamilton decomposition.

F26: [Fi90] If T is any tree with n edges, then the n-dimensional cube Q_n has an isomorphic factorization by T. Furthermore, there is an isomorphic factorization so that each copy of T is an induced subgraph.

RESEARCH PROBLEM

P3: Let $\mathcal{C}$ be one of the classes of circulant graphs, or Cayley graphs, or vertex-transitive graphs. Is it the case that for every graph $G \in \mathcal{C}$, whenever d divides $|E(G)|$, then there is an isomorphic factorization of G into d subgraphs?

6.2.6 Further Reading

REMARKS

R3: There is a long history and an extensive literature about imbedding Cayley graphs on orientable and non-orientable surfaces. See Chapter 7 in this volume. The books [GrTu87], [Ri74], and [Wh01] and a recent excellent survey [RiJaTuWa03ta] provide a good starting point for this topic.

R4: There are a variety of meaningful applications of Cayley graphs.

- Circulant graphs appear in the study of circular chromatic number. For a recent survey see [Zh01].
- Cayley graphs occur frequently in the literature on networks. A recent book on this topic is [Xu01] and a fundamental paper is [AkKr89].
- Cayley graphs play a central role in the work on expanders. Two excellent references are [Al95] and [Lu95].

R5: A survey on Cayley graph isomorphism is provided in [Li02].

R6: A good general discussion about vertex-transitive graphs and Cayley graphs is [Ba95]. A good starting point for reading about $\mathbb{NC}$ is [IrPr01].

References

[AkKr89] S. Akers and B. Krishnamurthy, A group-theoretic model for symmetric interconnection networks, *IEEE Trans. Comput.* **38** (1989), 555–566.

[Al92] B. Alspach, Cayley graphs with optimal fault tolerance, *IEEE Trans. Comput.* 41 (1992), 1337–1339.

[Al95] N. Alon, Tools from higher algebra, in *Handbook of Combinatorics Vol II*, eds. R. L. Graham, M. Grötschel and L. Lovász, MIT Press and North-Holland, 1995, 1749–1783.

[AlMi02] B. Alspach and M. Mishna, Enumeration of Cayley graphs and digraphs: I, *Discrete Math.* 256 (2002), 527–539.

[Ba77] L. Babai, Isomorphism problem for a class of point-symmetric structures, *Acta. Math. Acad. Sci. Hungar.* 29 (1977), 329–336.

[Ba95] L. Babai, Automorphism groups, isomorphism, reconstruction, in *Handbook of Combinatorics Vol II*, eds. R. L. Graham, M. Grötschel and L. Lovász, MIT Press and North-Holland, 1995, 1447–1540.

[Br64] N. G. de Bruijn, Pólya's theory of counting, *Applied Combinatorial Mathematics*, Ch. 5, Wiley, New York, 1964.

[ChQu81] C. C. Chen and N. Quimpo, On strongly hamiltonian abelian group graphs, Combinatorial Mathematics VIII, *Lecture Notes in Mathematics* **884**, Springer-Verlag, 1981, 23–34.

[Fi90] J. Fink, On the decomposition of n-cubes into isomorphic trees, *J. Graph Theory* **14** (1990), 405–411.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Wiley, New York, 1987.

[IrPr01] M. Iranmanesh and C. E. Praeger, On non-Cayley vertex-transitive graphs of order a product of three primes, *J. Combin. Theory Ser. B* **81** (2001), 1–19.

[Li99] C. H. Li, Finite CI-groups are soluble, *Bull. London Math. Soc.* **31** (1999), 419–423.

[Li02] C. H. Li, On isomorphisms of finite Cayley graphs — a survey, *Discrete Math.* **256** (2002), 301–334.

[Liu96] J. Liu, Hamiltonian decompositions of Cayley graphs on abelian groups of odd order, *J. Combin. Theory Ser. B*, **66** (1996), 75–86.

[Liu03ta] J. Liu, Hamiltonian decompositions of Cayley graphs on abelian groups of even order, *J. Combin. Theory Ser. B*, to appear.

[Lu95] A. Lubotzky, Cayley graphs: eigenvalues, expanders and random walks, *London Math. Soc. Lecture Note Ser.* **218** (1995), 155–189.

[Ma71] W. Mader, Eine Eigenschaft der Atome endlicher Graphen, *Arch. Math. (Basel)* **22** (1971), 333–336.

[Mu97] M. Muzychuk, On Ádám's conjecture for circulant graphs, *Discrete Math.* **167** (1997), 497–510.

[RiEtal03ta] R. B. Richter, J. Širáň, R. Jajcay, T. Tucker and M. Watkins, Cayley maps, preprint.

[Ri74] G. Ringel, *Map Color Theorem*, Springer-Verlag, New York, 1974.

[Sa58] G. Sabidussi, On a class of fixed-point-free graphs, *Proc. Amer. Math. Soc.* **9**(1958), 800–804.

[St85] R. Stong, On 1-factorizability of Cayley graphs, *J. Combin. Theory Ser. B* **39** (1985), 298–307.

[Tu67] J. Turner, Point-symmetric graphs with a prime number of points, *J. Combin. Theory* 3 (1967), 136–145.

[Wa70] M. Watkins, Connectivity of transitive graphs, *J. Combin. Theory* **8** (1970), 23–29.

[Wh01] A. White, *Graphs of Groups on Surfaces*, Mathematics Studies **188**, North-Holland, Amsterdam, 2001.

[Wi84] D. Witte, Cayley digraphs of prime-power order are hamiltonian, *J. Combin. Theory Ser. B* **40** (1984), 107–112.

[Xu01] J. Xu, *Topological Structure and Analysis of Interconnection Networks*, Kluwer, Dordrecht, 2001.

[Zh01] X. Zhu, Circular chromatic number: a survey, *Discrete Math.* **229** (2001), 371–410.

6.3 ENUMERATION

Paul K. Stockmeyer, The College of William and Mary

6.3.1 Counting Simple Graphs and Multigraphs
6.3.2 Counting Digraphs and Tournaments
6.3.3 Counting Generic Trees
6.3.4 Counting Trees in Chemistry
6.3.5 Counting Trees in Computer Science
References

Introduction

It is often important to know how many graphs there are with some desired property. Computer scientists can use such numbers in analyzing the time or space requirements of their algorithms, and chemists can make use of these numbers in organizing and cataloging lists of chemical molecules with various shapes. Indeed, any time that graphs are used to model some form of physical structure, the techniques of graphical enumeration are extremely valuable.

Many of the techniques for counting graphs are based on the master theorem in the historic 1937 work of George Pólya (see [PóRe87] for an English translation). Frank Harary [Ha55] and others exploited this master theorem in counting simple graphs, multigraphs, digraphs, and similar graphical structures.

Tree counting began with Arthur Cayley [Ca57, Ca89], who was the first to use the word "tree" for these structures. Methods for counting trees representing organic chemicals were developed by Blair and Henze [BlHe31a, BlHe31b]. Generic tree counting methods were advanced by Pólya [PóRe87], Richard Otter [Ot48], Harary and Gert Prins [HaPr59] and many others.

An exhaustive survey of results in graphical enumeration, far beyond what can be included here, can be found in [HaPa73]. Alternatively, if you know the first few terms of a graph-counting sequence, you can quite likely find more terms and at least one reference in [SlPl95].

6.3.1 Counting Simple Graphs and Multigraphs

DEFINITIONS

D1: A ***labeled graph*** is a graph with labels, typically $v_1, v_2, \ldots, v_n$, assigned to the vertices. Two labeled graphs with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

D2: The ***symmetric group*** S_n is the group of all permutations acting on the set $X_n = \{1, 2, \ldots, n\}$.

D3: The ***pair-permutation*** $\gamma^{(2)}$ induced by the permutation γ acting on the set X_n is the permutation acting on unordered pairs of elements of X_n defined by the rule $\gamma^{(2)} : \{x, y\} \mapsto \{\gamma(x), \gamma(y)\}$.

D4: The ***symmetric pair group*** $S_n^{(2)}$ induced by the symmetric group S_n is the permutation group $\{\gamma^{(2)} \mid \gamma \in S_n\}$.

FACTS

F1: The number of labeled simple graphs with n vertices and m edges is the binomial coefficient $\binom{\binom{n}{2}}{m}$. See Table 6.3.1.

F2: For $m > \frac{\binom{n}{2}}{2}$, the number of labeled simple graphs with n vertices and m edges is the same as the number of labeled simple graphs with n vertices and $\binom{n}{2} - m$ edges.

F3: The total number of labeled simple graphs with n vertices is $2^{\binom{n}{2}}$. See Table 6.3.1.

Table 6.3.1 **Labeled simple graphs with n vertices and m edges.**

$m \backslash n$	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1
1		1	3	6	10	15	21	28
2			3	15	45	105	210	378
3			1	20	120	455	1,330	3,276
4				15	210	1,365	5,985	20,475
5				6	252	3,003	20,349	98,280
6				1	210	5,005	54,264	376,740
7					120	6,435	116,280	1,184,040
8					45	6,435	203,490	3,108,105
9					10	5,005	293,930	6,906,900
10					1	3,003	352,716	13,123,110
11						1,365	352,716	21,474,180
12						455	293,930	30,421,755
13						105	203,490	37,442,160
14						15	116,280	40,116,600
Total	1	2	8	64	1,024	32,768	2,097,152	268,435,456

F4: [Gi56] The number K_n of connected labeled simple graphs with n vertices can be determined from the recurrence system

$$K_1 = 1, \qquad K_n = 2^{\binom{n}{2}} - \frac{1}{n}\sum_{i=1}^{n-1} i\binom{n}{i} 2^{\binom{n-i}{2}} K_i \quad \text{for } n > 1$$

See Table 6.3.2.

F5: Pólya's enumeration theorem, when used for counting graphical structures, involves partitions (j) of the set $X_n = \{1, 2, \ldots, n\}$. The integer j_k denotes the number of

Table 6.3.2 **Connected labeled simple graphs with n vertices.**

n	1	2	3	4	5	6	7	8
K_n	1	1	4	38	728	26,704	1,866,256	251,548,592

blocks of size k in the partition (j), for $k = 1, \ldots, n$. For example, if (j) is the partition $\{\{1,2\},\{3,4\},\{5,6,7\}\}$, then $j_2 = 2$, $j_3 = 1$, and $j_1 = j_4 = j_5 = j_6 = j_7 = 0$.

F6: The *cycle index* $Z(S_n^{(2)})$ of the symmetric pair group, used in counting simple graphs with n vertices, is

$$Z(S_n^{(2)}) = \frac{1}{n!}\sum_{(j)} \frac{n!}{\prod_k k^{j_k} j_k!} \prod_k a_k^{k\binom{j_k}{2}} (a_k a_{2k}^{k-1})^{j_{2k}} a_{2k+1}^{k j_{2k+1}} \prod_{r<s} a_{\mathrm{lcm}(r,s)}^{\gcd(r,s) j_r j_s}$$

Here the sum is taken over all partitions (j) of the set X_n, and $\mathrm{lcm}(r,s)$ and $\gcd(r,s)$ are the least common multiple and greatest common divisor of r and s, respectively. Explicit formulas for $Z(S_n^{(2)})$ for small values of n are:

$$Z(S_1^{(2)}) = 1$$

$$Z(S_2^{(2)}) = a_1$$

$$Z(S_3^{(2)}) = \frac{1}{3!}(a_1^3 + 3a_1a_2 + 2a_3)$$

$$Z(S_4^{(2)}) = \frac{1}{4!}(a_1^6 + 9a_1^2a_2^2 + 8a_3^2 + 6a_2a_4)$$

$$Z(S_5^{(2)}) = \frac{1}{5!}(a_1^{10} + 10a_1^4a_2^3 + 20a_1a_3^3 + 15a_1^2a_2^4 + 30a_2a_4^2 + 20a_1a_3a_6 + 24a_5^2)$$

$$Z(S_6^{(2)}) = \frac{1}{6!}(a_1^{15} + 15a_1^7a_2^4 + 40a_1^3a_3^4 + 60a_1^3a_2^6 + 180a_1a_2a_4^3 + 120a_1a_2a_3^2a_6 + 144a_5^3 + 40a_3^5 + 120a_3a_6^2)$$

F7: The generating function for counting simple graphs with n vertices by number of edges has the form

$$g_n(x) = \sum_{m=0}^{\binom{n}{2}} G_{n,m}x^m$$

where $G_{n,m}$ denotes the number of simple graphs with n vertices and m edges.

F8: [Ha55, PóRe87] The generating function $g_n(x)$ for counting n-vertex simple graphs by number of edges is obtained from the cycle index $Z(S_n^{(2)})$ by replacing each variable a_i with $1 + x^i$. See Table 6.3.3.

F9: The total number G_n of simple graphs with n vertices is obtained from the cycle index $Z(S_n^{(2)})$ by replacing each variable a_i with the number 2. See Table 6.3.3.

Table 6.3.3 **Simple graphs with n vertices and m edges.**

$m \backslash n$	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1
1		1	1	1	1	1	1	1
2			1	2	2	2	2	2
3			1	3	4	5	5	5
4				2	6	9	10	11
5				1	6	15	21	24
6				1	6	21	41	56
7					4	24	65	115
8					2	24	97	221
9					1	21	131	402
10					1	15	148	663
11						9	148	980
12						5	131	1,312
13						2	97	1,557
14						1	65	1,646
Total	1	2	4	11	34	156	1,044	12,346

F10: Asymptotically, the number G_n satisfies $G_n \sim \frac{2^{\binom{n}{2}}}{n!}$.

F11: The generating function for counting loopless multigraphs with n vertices by number of edges has the form

$$m_n(x) = \sum_{m=0}^{\infty} M_{n,m} x^m$$

where $M_{n,m}$ denotes the number of loopless multigraphs with n vertices and m edges.

F12: [Ha55, PóRe87] The generating function $m_n(x)$ for counting n-vertex loopless multigraphs by number of edges is obtained from the cycle index $Z(S_n^{(2)})$ by replacing each variable a_i with the infinite series $1 + x^i + x^{2i} + x^{3i} + \cdots$. See Table 6.3.4.

Table 6.3.4 **Loopless multigraphs with n vertices and m edges.**

$m \backslash n$	1	2	3	4	5	6
0	1	1	1	1	1	1
1		1	1	1	1	1
2		1	2	3	3	3
3		1	3	6	7	8
4		1	4	11	17	21
5		1	5	18	35	52
6		1	7	32	76	132
7		1	8	48	149	313
8		1	10	75	291	741
9		1	12	111	539	1,684
10		1	14	160	974	3,711

EXAMPLES

E1: Figure 6.3.1 shows the three different simple graphs with 4 vertices and 3 edges. There are 4 essentially different ways to label each of the first two and 12 ways to label the third. Thus there are 20 different labeled simple graphs with 4 vertices and 3 edges.

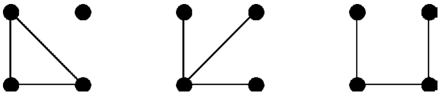


Figure 6.3.1 **Simple graphs with 4 vertices and 3 edges.**

E2: Figure 6.3.2 shows the three loopless multigraphs that together with the graphs in Figure 6.3.1 form the six different multigraphs with 4 vertices and 3 edges.

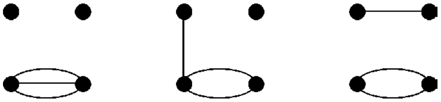


Figure 6.3.2 **Additional loopless multigraphs with 4 vertices and 3 edges.**

6.3.2 Counting Digraphs and Tournaments

DEFINITIONS

D5: A ***labeled digraph*** is a digraph with labels, typically $v_1, v_2, \ldots, v_n$, assigned to the vertices. Two labeled digraphs with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

D6: A ***tournament*** (or ***round-robin tournament***) is a digraph in which, for each pair u, v of distinct vertices, either there exists an arc from u to v or an arc from v to u but not both.

D7: A tournament is ***strong*** (or ***strongly connected***) if for each pair u, v of vertices, there exist directed paths from u to v and from v to u.

D8: The ***ordered-pair-permutation*** $\gamma^{[2]}$ induced by the permutation γ acting on the set $X_n = \{1, 2, \ldots, n\}$ is the permutation acting on ordered pairs of elements of X_n defined by the rule $\gamma^{[2]} : (x, y) \mapsto (\gamma(x), \gamma(y))$.

D9: The ***symmetric ordered pair group*** $S_n^{[2]}$ induced by the symmetric group S_n is the permutation group $\{\gamma^{[2]} \mid \gamma \in S_n\}$.

FACTS

F13: The number of labeled loopless digraphs with n vertices and m arcs is the binomial coefficient $\binom{n(n-1)}{m}$. See Table 6.3.5.

F14: For $m > \frac{n(n-1)}{2}$, the number of labeled loopless digraphs with n vertices and m arcs is the same as the number of labeled loopless digraphs with n vertices and $n(n-1) - m$ arcs.

Table 6.3.5 **Labeled loopless digraphs with n vertices and m arcs.**

$m \backslash n$	1	2	3	4	5
0	1	1	1	1	1
1		2	6	12	20
2		1	15	66	190
3			20	220	1,140
4			15	495	4,845
5			6	792	15,504
6			1	924	38,760
7				792	77,520
8				495	125,970
9				220	167,960
10				66	184,756
Total	1	4	64	4,096	1,048,576

F15: The total number of labeled loopless digraphs with n vertices is $2^{n(n-1)}$. See Table 6.3.5.

F16: The number of labeled tournaments with n vertices is $2^{\binom{n}{2}}$, the same as the number of simple graphs with n vertices.

F17: The *cycle index* $Z(S_n^{[2]})$ of the symmetric ordered pair group, used in counting loopless digraphs with n vertices, is

$$Z(S_n^{[2]}) = \frac{1}{n!} \sum_{(j)} \frac{n!}{\prod_k k^{j_k} j_k!} \prod_k a_k^{(k-1)j_k + 2k\binom{j_k}{2}} \prod_{r<s} a_{\operatorname{lcm}(r,s)}^{2\gcd(r,s) j_r j_s}$$

Here the sum is taken over all partitions (j) of the set X_n, and $\operatorname{lcm}(r,s)$ and $\gcd(r,s)$ are the least common multiple and greatest common divisor of r and s, respectively. Explicit formulas for $Z(S_n^{[2]})$ for small values of n are:

$$Z(S_1^{[2]}) = 1$$

$$Z(S_2^{[2]}) = \frac{1}{2!}(a_1^2 + a_2)$$

$$Z(S_3^{[2]}) = \frac{1}{3!}(a_1^6 + 3a_2^3 + 2a_3^2)$$

$$Z(S_4^{[2]}) = \frac{1}{4!}(a_1^{12} + 6a_1^2a_2^5 + 8a_3^4 + 3a_2^6 + 6a_4^3)$$

$$Z(S_5^{[2]}) = \frac{1}{5!}(a_1^{20} + 10a_1^6a_2^7 + 20a_1^2a_3^6 + 15a_2^{10} + 30a_4^5 + 20a_2a_3^2a_6^2 + 24a_5^4)$$

$$Z(S_6^{[2]}) = \frac{1}{6!}(a_1^{30} + 15a_1^{12}a_2^9 + 40a_1^6a_3^8 + 45a_1^2a_2^{14} + 90a_1^2a_4^7 + 120a_2^3a_3^4a_6^2$$
$$+ 144a_5^6 + 15a_2^{15} + 90a_2a_4^7 + 40a_3^{10} + 120a_6^5)$$

F18: The generating function $d_n(x)$ for counting loopless digraphs with n vertices by number of arcs has the form

$$d_n(x) = \sum_{m=0}^{n(n-1)} D_{n,m} x^m$$

where $D_{n,m}$ denotes the number of loopless digraphs with n vertices and m arcs.

F19: [Ha55, PóRe87] The generating function $d_n(x)$ for counting n-vertex loopless digraphs by number of arcs is obtained from the cycle index $Z(S_n^{[2]})$ by replacing each variable a_i with $1 + x^i$. See Table 6.3.6.

F20: The total number D_n of loopless digraphs with n vertices is obtained from the cycle index $Z(S_n^{[2]})$ by replacing each variable a_i with the number 2. See Table 6.3.6.

Table 6.3.6 Digraphs with n vertices and m arcs.

$m \backslash n$	1	2	3	4	5
0	1	1	1	1	1
1		1	1	1	1
2		1	4	5	5
3			4	13	16
4			4	27	61
5			1	38	154
6			1	48	379
7				38	707
8				27	1,155
9				13	1,490
10				5	1,670
Total	1	3	16	218	9,608

F21: Asymptotically, the number D_n satisfies $D_n \sim \frac{2^{n(n-1)}}{n!}$.

F22: [Da54] The number T_n of tournaments on n vertices is given by the formula

$$T_n = \frac{1}{n!} \sum_{(j)}{}' \frac{n!}{\prod_k k^{j_k} j_k!} 2^{D(j)}$$

where the sum is over all partitions (j) of X_n into odd size blocks, and where

$$D(j) = \frac{1}{2} \left(\sum_{r=1}^{n} \sum_{s=1}^{n} \gcd(r,s) j_r j_s - \sum_{k=1}^{n} j_k \right)$$

See Table 6.3.7.

F23: [Mo68] Let $T(x) = x + x^2 + 2x^3 + 4x^4 + 12x^5 + 56x^6 + \cdots$ be the generating function for tournaments, from the formula of Fact 22. Then the generating function $S(x) = x + x^3 + x^4 + 6x^5 + 35x^6 + \cdots$ for strong tournaments can be computed from the relation

$$S(x) = \frac{T(x)}{1 + T(x)}$$

See Table 6.3.7. Note that there are no strong tournaments with exactly two vertices.

Table 6.3.7 **Tournaments and strong tournaments with n vertices.**

n	Tournaments	Strong Tournaments
1	1	1
2	1	0
3	2	1
4	4	1
5	12	6
6	56	35
7	456	353
8	6,880	6,008
9	191,536	178,133
10	9,733,056	9,355,949
11	903,753,248	884,464,590
12	154,108,311,168	152,310,149,735

EXAMPLES

E3: Figure 6.3.3 shows the four loopless digraphs with 3 vertices and 3 arcs. There are 6 essentially different ways to label each of the first three and 2 ways to label the fourth. Thus there are 20 different labeled loopless digraphs with 3 vertices and 3 arcs.

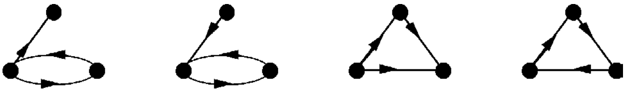


Figure 6.3.3 **The four loopless digraphs with 3 vertices and 4 arcs.**

E4: Figure 6.3.4 shows the four tournaments with 4 vertices. Only the last tournament is strong.

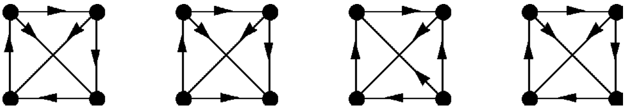


Figure 6.3.4 **The four tournaments with 4 vertices.**

6.3.3 Counting Generic Trees

DEFINITIONS

D10: A ***labeled tree*** is a tree in which labels, typically $v_1, v_2, \ldots, v_n$, have been assigned to the vertices. Two labeled trees with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

D11: A ***rooted tree*** is a tree in which one vertex, the root, is distinguished. Two rooted trees are considered the same only if there is an isomorphism from one to the other that maps the root of the first to the root of the second.

D12: A ***reduced tree*** (or ***homeomorphically reduced tree***) is a tree with no vertices of degree 2.

FACTS

F24: **Cayley's formula** [Ca89]: The number of labeled trees with n vertices is n^{n-2}. See Table 6.3.8.

F25: The number of rooted labeled trees n^{n-1}. See Table 6.3.8.

Table 6.3.8 **Labeled trees and rooted labeled trees with n vertices.**

n	Labeled Trees	Rooted Labeled Trees
1	1	1
2	1	2
3	3	9
4	16	64
5	125	625
6	1,296	7,776
7	16,807	117,649
8	262,144	2,097,152
9	4,782,969	43,046,721
10	100,000,000	1,000,000,000
11	2,357,947,691	25,937,424,601
12	61,917,364,224	743,008,370,688
13	1,792,160,394,037	23,298,085,122,481
14	56,693,912,375,296	793,714,773,254,144
15	1,946,195,068,359,375	29,192,926,025,390,625
16	72,057,594,037,927,936	1,152,921,504,606,846,980

F26: The generating function for the number R_n of rooted trees with n vertices is

$$r(x) = \sum_{n=1}^{\infty} R_n x^n = x + x^2 + 2x^3 + 4x^4 + 9x^5 + 20x^6 + \cdots$$

F27: [Ca57] The coefficients R_n of the generating function $r(x)$ for rooted trees can be determined from the recurrence relation

$$r(x) = x \prod_{i=1}^{\infty} (1 - x^i)^{-R_i}$$

An alternative defining expression for this function due to Pólya (see [PóRe87]) is

$$r(x) = x \exp \left(\sum_{i=1}^{\infty} \frac{r(x^i)}{i} \right)$$

See Table 6.3.9.

F28: The generating function for the number T_n of trees with n vertices is

$$t(x) = \sum_{n=1}^{\infty} T_n x^n = x + x^2 + x^3 + 2x^4 + 3x^5 + 6x^6 + \cdots$$

F29: **Otter's formula** [Ot48]: The coefficients T_n of the generating function $t(x)$ for trees can be determined from the generating function $r(x)$ for rooted trees in Fact 27 by using the formula

$$t(x) = r(x) - \tfrac{1}{2}\left(r(x)^2 - r(x^2)\right)$$

See Table 6.3.9.

F30: Counting reduced trees requires an auxiliary function

$$f(x) = \sum_{i=1}^{\infty} F_i x^i = x + x^3 + x^4 + 2x^5 + 3x^6 + 6x^7 + 10x^8 + \cdots$$

F31: The coefficients F_i of the auxiliary function can be determined from the recurrence relation

$$f(x) = \frac{x}{1+x} \prod_{i=1}^{\infty} (1 - x^i)^{-F_i}$$

An alternative defining expression for this function is

$$f(x) = \frac{x}{1+x} \exp\left(\sum_{i=1}^{\infty} \frac{f(x^i)}{i} \right)$$

F32: The generating function for the number H_n of reduced trees with n vertices is

$$h(x) = \sum_{n=1}^{\infty} H_n x^n = x + x^2 + x^4 + x^5 + 2x^6 + 2x^7 + 4x^8 + \cdots$$

F33: [HaPr59] The coefficients H_n of the generating function $h(x)$ for counting reduced trees can be determined from the auxiliary function $f(x)$ in Fact 31 by using the formula

$$h(x) = (1+x)f(x) - \left(\frac{1+x}{2}\right) f(x)^2 + \left(\frac{1-x}{2}\right) f(x^2)$$

See Table 6.3.9 below. Note that there are no reduced trees with exactly 3 vertices.

EXAMPLES

E5: Figure 6.3.5 shows the three trees with 5 vertices. There are 60 essentially different ways to label each of the first two and 5 essentially different ways to label the third. Thus there are 125 different labeled trees with 5 vertices.

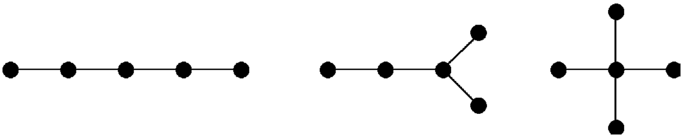


Figure 6.3.5 **The three trees with 5 vertices.**

Table 6.3.9 Rooted trees, trees, and reduced trees with n vertices.

n	Rooted Trees	Trees	Reduced Trees
1	1	1	1
2	1	1	1
3	2	1	0
4	4	2	1
5	9	3	1
6	20	6	2
7	48	11	2
8	115	23	4
9	286	47	5
10	719	106	10
11	1,842	235	14
12	4,766	551	26
13	12,486	1,301	42
14	32,973	3,159	78
15	87,811	7,741	132
16	235,381	19,320	249
17	634,847	48,629	445
18	1,721,159	123,867	842
19	4,688,676	317,955	1,561
20	12,826,228	823,065	2,988
21	35,221,832	2,144,505	5,671
22	97,055,181	5,623,756	10,981
23	268,282,855	14,828,074	21,209
24	743,724,984	39,299,897	41,472
25	2,067,174,645	104,636,890	81,181
26	5,759,636,510	279,793,450	160,176
27	16,083,734,329	751,065,460	316,749
28	45,007,066,269	2,023,443,032	629,933
29	126,186,554,308	5,469,566,585	1,256,070
30	354,426,847,597	14,830,871,802	2,515,169
31	997,171,512,998	40,330,829,030	5,049,816
32	2,809,934,352,700	109,972,410,221	10,172,638
33	7,929,819,784,355	300,628,862,480	20,543,579
34	22,409,533,673,568	823,779,631,721	41,602,425
35	63,411,730,258,053	2,262,366,343,746	84,440,886
36	179,655,930,440,464	6,226,306,037,178	171 794,492
37	509,588,049,810,620	17,169,677,490,714	350,238,175
38	1,447,023,384,581,029	47,436,313,524,262	715,497,037
39	4,113,254,119,923,150	131,290,543,779,126	1,464,407,113
40	11,703,780,079,612,453	363,990,257,783,343	3,002,638,286

E6: There are 3 essentially different ways to root the first tree in Figure 6.3.5, 4 essentially different ways to root the second, and 2 essentially different ways to root the third. Thus there are 9 rooted trees with 5 vertices.

E7: The third tree in Figure 6.3.5 is the only reduced tree with 5 vertices.

6.3.4 Counting Trees in Chemistry

DEFINITIONS

D13: A ***1-4 tree*** is a tree in which each vertex has degree 1 or 4.

D14: A ***1-rooted 1-4 tree*** is a 1-4 tree rooted at a vertex of degree 1.

REMARKS

R1: The 2-4 trees model many types of organic chemical molecules such as saturated hydrocarbons (or alkanes). These molecules have the chemical formula $\mathrm{C_nH_{2n+2}}$ and consist of n carbon atoms of valence 4 and $2n+2$ hydrogen atoms of valence 1.

R2: The 1-rooted 1-4 trees model the monosubstituted hydrocarbons such as the alcohols with the chemical formula $\mathrm{C_nH_{2n+2}OH}$ and consisting of n carbon atoms, $2n+1$ hydrogen atoms, and an OH group.

FACTS

F34: The generating function for the number A_n of 1-rooted 1-4 trees with n vertices of degree 4 and $2n+2$ vertices of degree 1 (including the root) is

$$a(x) = \sum_{n=0}^{\infty} A_n x^n = 1 + x + x^2 + 2x^3 + 4x^4 + 8x^5 + 17x^6 + \cdots$$

F35: [BlHe31a] The coefficients A_n of the generating function $a(x)$ for 1-rooted 1-4 trees can be determined from the recurrence relation

$$a(x) = 1 + \tfrac{x}{6}\left(a(x)^3 + 3a(x)a(x^2) + 2a(x^3)\right)$$

See Table 6.3.10.

F36: Counting (unrooted) 1-4 trees requires first counting the 4-rooted 1-4 trees, rooted at a vertex of degree 4. The generating function for the number G_n of 4-rooted 1-trees is

$$g(x) = \sum_{n=1}^{\infty} G^n x^n = x + x^2 + 2x^3 + 4x^4 + 9x^5 + 18x^6 + \cdots$$

F37: The coefficients G_n of the generating function $g(x)$ for counting 4-rooted 1-4 trees can be determined from the generating function $a_n(x)$ for 1-rooted 1-4 trees in Fact 35 by using the formula

$$g(x) = \tfrac{x}{24}\left(a(x)^4 + 6a(x)^2a(x^2) + 8a(x)a(x^3) + 3a(x^2)^2 + 6a(x^4)\right)$$

F38: The generating function for the number B_n of (unrooted) 1-4 trees with n vertices of degree 4 and $2n+2$ vertices of degree 1 is

$$b(x) = \sum_{n=0}^{\infty} B_n x^n = 1 + x + x^2 + x^3 + 2x^4 + 3x^5 + 5x^6 + \cdots$$

F39: [BlHe31b] The coefficients B_n of the generating function $b(x)$ for counted 1-4 trees can be determined from the functions $g(x)$ and $a(x)$ using the formula

$$b(x) = g(s) + a(x) - \tfrac{1}{2}\left(a(x)^2 - a(x^2)\right)$$

See Table 6.3.10 below.

EXAMPLES

E8: Figure 6.3.6 shows the three different 1-4 trees with 5 vertices of degree 4.

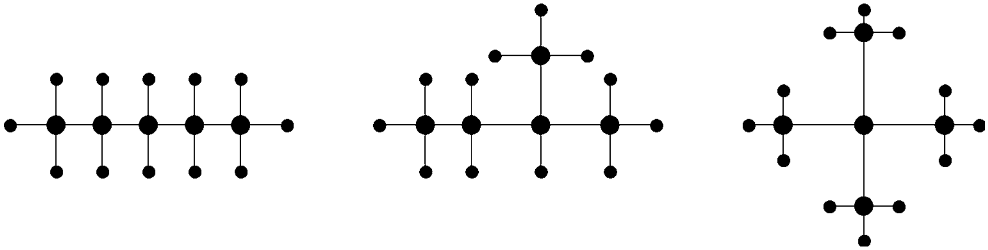


Figure 6.3.6 **The three 1-4 trees with 5 vertices of degree 4.**

E9: The first 1-4 tree in Figure 6.3.6 can be rooted at a vertex of degree 1 in 3 essentially different ways, the second in 4 essentially different ways, and the third in essentially only 1 way. Thus there are 8 different 1-rooted 1-4 trees with 5 vertices of degree 4.

6.3.5 Counting Trees in Computer Science

DEFINITIONS

D15: An ***ordered tree*** is recursively defined to consist of a root vertex and a sequence $t_1, t_2, \ldots,\ t_m$ of $m \geq 0$ principal subtrees that are themselves ordered trees. The root vertex of an ordered tree is joined by an edge to the root of each principal subtree.

D16: A ***binary tree*** consists of a root vertex and at most two principal subtrees that are themselves binary trees. Each principal subtree must be specified as either the left subtree or the right subtree.

D17: The ***children*** of the root vertex of an ordered tree or a binary tree are the roots of the principal subtrees.

D18: A ***left-right tree*** is a binary tree in which each vertex has either 0 or 2 children.

REMARKS

R3: In computer science, trees are usually drawn with the root at the top.

R4: Ordered trees are used to represent structures such as family trees, showing all descendants of a person represented by the root. The roots of the principal subtrees would represent the children of the root person, in order of birth.

Table 6.3.10 **1-Rooted 1-4 trees and 1-4 trees with n vertices of degree 4**

	1-Rooted 1-4 Trees	1-4 Trees
n	(Alcohols)	(Alkanes)
1	1	1
2	1	1
3	2	1
4	4	2
5	8	3
6	17	5
7	39	9
8	89	18
9	211	35
10	507	75
11	1,238	159
12	3,057	355
13	7,639	802
14	19,241	1,858
15	48,865	4,347
16	124,906	10,359
17	321,198	24,894
18	830,219	60,523
19	2,156,010	148,284
20	5,622,109	366,319
21	14,715,813	910,726
22	38,649,152	2,278,658
23	101,821,927	5,731,580
24	269,010,485	14,490,245
25	712,566,567	36,797,588
26	1,891,993,344	93,839,412
27	5,034,704,828	240,215,803
28	13,425,117,806	617,105,614
29	35,866,550,869	1,590,507,121
30	95,991,365,288	4,111,846,763
31	257,332,864,506	10,660,307,791
32	690,928,354,105	27,711,253,769
33	1,857,821,351,559	72,214,088,660
34	5,002,305,607,153	188,626,236,139
35	13,486,440,075,669	493,782,952,902
36	36,404,382,430,278	1,295,297,588,128
37	98,380,779,170,283	3,404,490,780,161
38	266,158,552,000,477	8,964,747,474,595
39	720,807,976,831,447	23,647,478,933,969
40	1,954,002,050,661,819	62,481,801,147,341

R5: Binary trees are some of the tree structures most easily represented in a computer. Other types of trees are often converted into binary trees for computer representation.

R6: Left-right trees are frequently used to represent arithmetic expressions, in which the leaves correspond to numbers and the other vertices represent binary operations such as $+$, $-$, $\times$, or $\div$.

FACTS

F40: The ***Catalan numbers*** C_n are defined by the recursion

$$C_0 = 1 \qquad C_n = C_0C_{n-1} + C_1C_{n-2} + \cdots + C_{n-1}C_0 \quad \text{for } n \geq 1$$

whose solution is

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)!\,n!} \quad \text{for } n \geq 0$$

See Table 6.3.11.

Table 6.3.11 **The Catalan Numbers**

n	Catalan Number	n	Catalan Number
1	1	17	129,644,790
2	2	18	477,638,700
3	5	19	1,767,263,190
4	14	20	6,564,120,420
5	42	21	24,466,267,020
6	132	22	91,482,563,640
7	429	23	343,059,613,650
8	1,430	24	1,289,904,147,324
9	4,862	25	4,861,946,401,452
10	16,796	26	18,367,353,072,152
11	58,786	27	69,533,550,916,004
12	208,012	28	263,747,951,750,360
13	742,900	29	1,002,242,216,651,368
14	2,674,440	30	3,814,986,502,092,304
15	9,694,845	31	14,544,636,039,226,909
16	35,357,670	32	55,534,064,877,048,198

F41: The number of ordered trees with n vertices is the Catalan number C_{n-1}. See Table 6.3.11.

F42: The number of binary trees with n vertices is C_n. See Table 6.3.11.

F43: The number of left-right trees with $2n+1$ vertices is also C_n. See Table 6.3.11.

EXAMPLES

E10: Figure 6.3.7 shows the 5 ordered trees with 4 vertices, Figure 6.3.8 shows the 5 binary trees with 3 vertices, and Figure 6.3.9 shows the 5 left-right trees with 7 vertices.

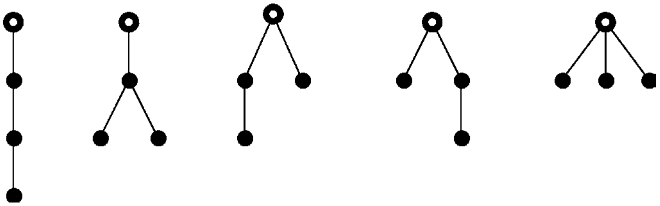


Figure 6.3.7 The 5 ordered trees with 4 vertices.

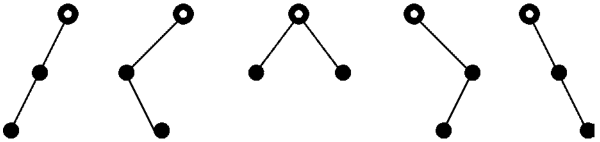


Figure 6.3.8 The 5 binary trees with 3 vertices.

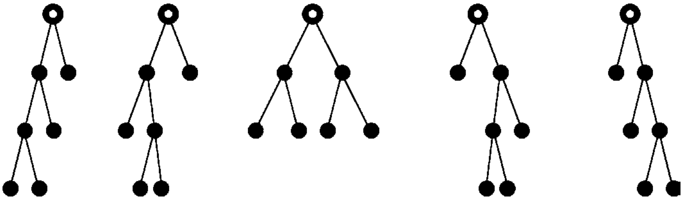


Figure 6.3.9 The 5 left-right trees with 7 vertices.

References

[BlHe31a] C. M. Blair and H. R. Henze, The number of structurally isomeric alcohols of the methanol series, *J. Amer. Chem. Soc.* 53 (1931), 3042–3046.

[BlHe31b] C. M. Blair and H. R. Henze, The number of isomeric hydrocarbons of the methane series, *J. Amer. Chem Soc.* 53 (1931), 3077–3085.

[Ca57] A. Cayley, On the theory of the analytical forms called trees, *Philos. Mag.* 13 (1857), 19–30.

[Ca89] A. Cayley, A theorem on trees, *Quart. J. Math.* 23 (1889), 376–378.

[Da54] R. L. Davis, Structures of dominance relations, *Bull. Math. Biophys.* 16 (1954), 131–140.

[Ha55] F. Harary, The number of linear, directed, rooted, and connected graphs, *Trans. Amer. Math. Soc.* 78 (1955), 445–463.

[Ha69] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[HaPa73] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic Press, 1973.

[HaPr59] F. Harary and G. Prins, The number of homeomorphically irreducible trees, and other species, *Acta Math.* 101 (1959), 141–162.

[Gi56] E. N. Gilbert, Enumeration of labeled graphs, *Canad. J. Math.* 8 (1956), 405–411.

[Mo68] J. W. Moon, *Topics on Tournaments*, Holt, Rinehart and Winston, 1968.

[Ot48] R. Otter, The number of trees, *Ann. of Math.* 49 (1948), 583–599.

[PóRe87] G. Pólya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs, and Chemical Compounds*, Springer-Verlag, 1987.

[Ri58] J. Riordan, *An Introduction to Combinatorial Analysis*, Wiley, 1958.

[SlPl95] N. J. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995.

6.4 GRAPHS AND VECTOR SPACES

Krishnaiyan "KT" Thulasiraman, University of Oklahoma

6.4.1 Basic Concepts and Definitions
6.4.2 The Circuit Subspace in an Undirected Graph
6.4.3 The Cutset Subspace in an Undirected Graph
6.4.4 Relationship between Circuit and Cutset Subspaces
6.4.5 The Circuit and Cutset Spaces in a Directed Graph
6.4.6 Two Circ/Cut-Based Tripartitions of a Graph
6.4.7 Realization of Circuit and Cutset Spaces
References

Introduction

Electrical circuit theory is one of the earliest applications of graph theory to a problem in physical science. The dynamic behavior of an electrical circuit is governed by three laws: Kirchhoff's voltage law, Kirchhoff's current law and Ohm's law. Each element in a circuit is associated with two variables, namely, the current variable and the voltage variable. Kirchhoff's voltage law requires that the algebraic sum of the voltages around a circuit is zero, and Kirchhoff's current law requires that the algebraic sum of the currents across a cut is zero. Thus, circuits and cuts define a linear relationship among the voltage variables and a linear relationship among the current variables, respectively. It is for this reason that circuits, cuts, and the vector spaces associated with them have played a major role in the discovery of several fundamental properties of electrical circuits arising from the structure or the interconnection of the circuit elements. Several graph theorists and circuit theorists have immensely contributed to the development of what we may now call the structural theory of electrical circuits. The significance of the results to be presented in this section goes well beyond their application to circuit theory. They will bring out the fundamental duality that exists between circuits and cuts and the influence of this duality on the structural theory of graphs. Most of the results in this section are also relevant to the development of combinatorial optimization theory as well as matroid theory.

6.4.1 Basic Concepts and Definitions

Although the terms *node* and *oriented graph* are commonly used in electrical circuit theory, we use the terms *vertex* and *directed graph* along with all the other basic terminology of graph theory established in Chapter 1. For the sake of completeness, we begin with a review of certain basic concepts and definitions. For concepts not discussed here, the reader is referred to [GrYe99] and [ThSw92].

NOTATION: Unless otherwise specified, $G = (V, E)$ is a graph (or digraph) with n vertices, $V = \{v_1, v_2, \ldots, v_n\}$, and m edges, $E = \{e_1, e_2, \ldots e_m\}$.

NOTATION: If vertices v_i and v_j are the endpoints (or end vertices) of an edge then, when there is no ambiguity, we denote that edge by the ordered pair (v_i, v_j).

DEFINITION

D1: A graph is called a ***trivial graph*** if it has only one vertex and no edge. A graph with no edges is called an ***empty graph***. A graph with no vertices and hence no edges is called a ***null graph*** and will be denoted by Ø.

REMARK

R1: In this section we consider only graphs in which all edges have two distinct endpoints (i.e., no self-loops).

EXAMPLE

E1: Examples 1 through 9 in this section refer to the graph shown in Figure 6.4.1.

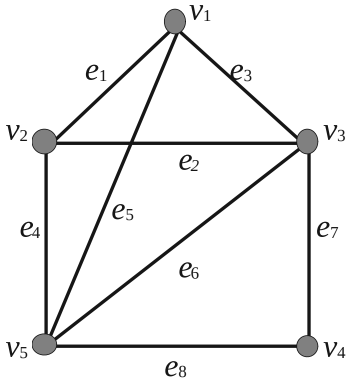


Figure 6.4.1

Subgraphs and Complements

DEFINITIONS

D2: A graph $G' = (V', E')$ is called a ***subgraph*** of graph $G = (V, E)$ if $V' \subseteq V$, $E' \subseteq E$ and V' contains all the endpoints of all the edges in E'.

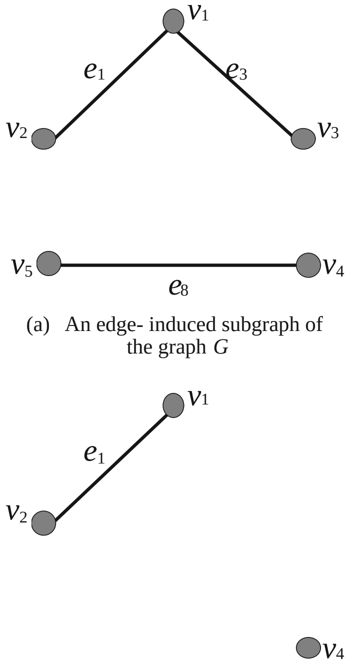
D3: Each subset $E' \subseteq E$ defines a unique subgraph $G' = (V', E')$ of graph $G = (V, E)$, where V' consists of only those vertices which are the endpoints of the edges in E'. The subgraph G' is called the ***induced subgraph of G on the edge set E'***. Note that an edge-induced subgraph will not have isolated vertices.

D4: Each subset $V' \subseteq V$ defines a unique subgraph $G' = (V', E')$ of graph $G = (V, E)$, where E' consists of those edges whose endpoints are in V'. The subgraph G' is called the ***induced subgraph of G on the vertex set V'***. Note that a vertex-induced subgraph may have isolated vertices.

D5: Given a subgraph $G' = (V', E')$ of graph $G = (V, E)$, the subgraph $G'' = (V, E - E')$ is called the ***(edge-)complement of G' in G***.

EXAMPLES

E2: For the set $E' = \{e_1, e_3, e_8\}$, the corresponding edge-induced subgraph of graph G in Figure 6.4.1 is shown in Figure 6.4.2 (a). For the set $V' = \{v_{1,} v_2, v_4\}$, the corresponding vertex-induced subgraph of G is shown in Figure 6.4.2 (b).

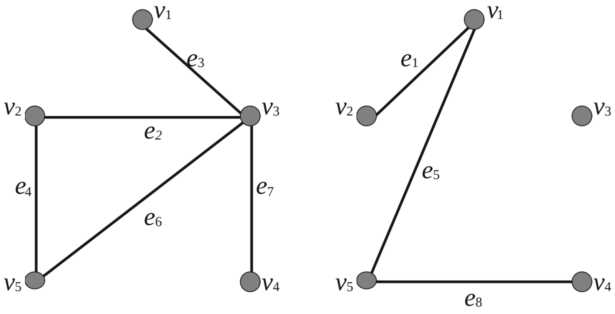


(a) An edge- induced subgraph of the graph *G*

(b) A vertex-induced subgraph of the graph *G*

Figure 6.4.2 **An edge-induced subgraph and a vertex-induced subgraph.**

E3: The complement of the subgraph G' of Figure 6.4.3(a) in the graph G of Figure 6.4.1 is shown in Figure 6.4.3(b).



(a) Subgraph *G*′ (b) Complement of *G*′ in the graph G

Figure 6.4.3 **A subgraph G' and its complement in G.**

Components, Spanning Trees, and Cospanning Trees

DEFINITIONS

D6: A ***closed trail*** is a closed walk with no repeated edges.

TERMINOLOGY: A closed trail is also called a *circ*, which we formally state in Definition 20 and use thereafter.

D7: A ***circuit*** is a closed trail with no repeated vertices except the initial and terminal ones.

TERMINOLOGY: Several authors use the term *cycle* instead of circuit. In electrical circuit literature, the term circuit is commonly understood as defined in Definition 7.

D8: A graph G is ***connected*** if there is a path between every pair of vertices of G.

D9: A maximal connected subgraph of a graph is called a ***component*** of the graph. An isolated vertex is by itself considered a single component.

D10: A ***tree*** of a graph G is a connected subgraph containing no circuits. If a tree of a connected graph G contains all the vertices of G then it is called a ***spanning tree*** of G. The complement of a spanning tree T in G is called a ***cospanning tree*** of G.

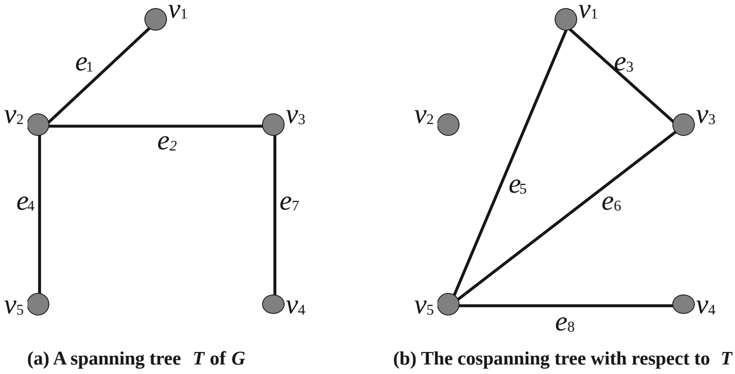
D11: A ***spanning forest*** of a non-connected graph G with p components is a collection of p spanning trees, one for each component.

D12: The edges of a spanning tree T are called the ***branches*** of T. The edges of a cospanning tree are called the ***chords*** of the spanning tree.

D13: Let G be an n-vertex graph with m edges and p components. The ***rank*** $\rho(G)$ and ***nullity*** $\mu(G)$ of G are given by $\rho(G) = n - p$ and $\mu(G) = m - n + p$.

EXAMPLE

E4: A spanning tree T and the corresponding cospanning tree of the connected graph G of Figure 6.4.1 are shown in Figure 6.4.4.



(a) A spanning tree T of G

(b) The cospanning tree with respect to T

Figure 6.4.4 **A spanning tree and corresponding cospanning tree of graph G.**

FACTS

F1: There is exactly one path between any two vertices of a spanning tree.

F2: A spanning tree of a connected n-vertex graph has $n-1$ branches and a cospanning tree has $m-n+1$ chords. A spanning forest of a graph having p components has $n-p$ branches and $m-n+p$ chords.

REMARK

R2: Unless stated otherwise, all graphs G considered in this section are connected.

Cuts and Cutsets

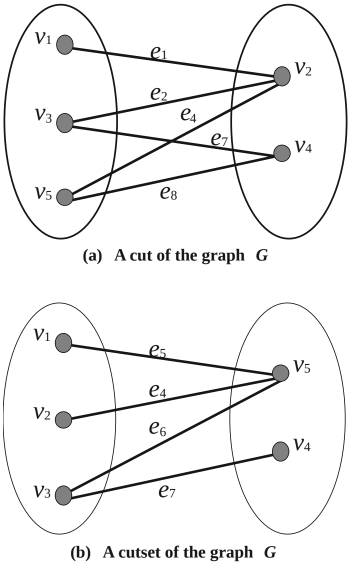
DEFINITIONS

D14: Consider a connected graph $G = (V, E)$. Let V_1 and V_2 be two disjoint subsets of V such that $V = V_1 \cup V_2$ (i.e., V_1 and V_2 form a *partition* of V). Then the set of all those edges of G having one end vertex in V_1 and the other in V_2 is called a ***cut*** of G. This cut is denoted as $\langle V_1, V_2\rangle$. The set of edges incident on a vertex forms a cut, and is called an ***incidence set***.

D15: Removal of the edges in a cut from a connected graph G will disconnect the graph. In other words, the resulting graph will have at least two components. A cut of a connected graph is called a ***cutset*** if the removal of the edges in the cut results in a non-connected graph with *exactly* two components. Equivalently, a cutset of a connected graph is a minimal set of edges whose removal disconnects the graph.

EXAMPLE

E5: For the graph G in Figure 6.4.1, the cut $\langle V_1, V_2\rangle$, where $V_1 = \{v_1, v_3, v_5\}$ and $V_2 = \{v_2, v_4\}$ consists of the edges e_1, e_2, e_4, e_7, and e_8, and it is shown in Figure 6.4.5(a). Removing these edges results in a non-connected graph with three components. So, $\langle V_1, V_2\rangle$ is not a cutset. A cutset consisting of the edges e_4, e_5, e_6, and e_7 is shown in Figure 6.4.5(b). Removing these edges results in a non-connected graph with two components.



(a) A cut of the graph G

(b) A cutset of the graph G

Figure 6.4.5 A cut and a cutset of the graph G.

The Vector Space of a Graph under Ring Sum of Its Edge Subsets

DEFINITIONS

D16: Suppose the edge-set of a graph G is $E = \{e_1, e_2, e_3, \ldots, e_m\}$. Then each subset E' of E can be represented by a ***binary m-vector*** in which the ith component is 1 if and only if the edge e_i is in E'. For example, the binary vector $(1, 0, 0, 1, 1, 1, 0, 0)$ represents the edge subset $\{e_1, e_4, e_5, e_6\}$ of the graph G of Figure 6.4.1.

D17: The ***ring sum*** (or ***symmetric difference***) of two sets E_1 andE_2, denoted as $E_1 \oplus E_2$, is the set of those edges which belong to E_1 or to E_2 but not to both E_1 and E_2.

D18: The ***ring sum*** of two m-vectors $X = (x_1, x_2, x_3, \ldots, x_i, \ldots, x_m)$ and $Y = (y_1, y_2, y_3, \ldots, y_i, \ldots, y_m)$ is the vector $Z = (z_1, z_2, z_3, \ldots, z_i, \ldots, z_m)$, where $z_i = x_i \otimes y_i$ and $\otimes$ is the logical *exclusive-or* operation (i.e., $1 \otimes 0 = 1$; $0 \otimes 1 = 1$; $0 \otimes 0 = 0$; and $1 \otimes 1 = 0$).

FACT

F3: The m-vector representing the ring sum of two subsets of edges is the ring sum of the m-vectors representing these edge subsets. The set of m-vectors representing all the 2^m edge subsets of a graph G (including the null set) forms an m-dimensional ***vector space*** over GF(2), the field of integers modulo 2, under the ring sum operation $\oplus$.

NOTATION: This vector space of edge subsets of a graph G (and hence of the corresponding edge-induced subgraphs of G) is denoted by $\Psi(G)$.

REMARKS

R3: Throughout this section all vectors are assumed to be row vectors.

R4: In this section an edge subset is used to refer to the corresponding edge-induced subgraph. The vector space $\mathit{\Psi}(G)$ will be used to denote the vector space of all binary m-vectors as well as the vector space of all edge-induced subgraphs of G. Observe that the null set (or null graph $\emptyset$) is the 0-vector of $\Psi(G)$.

R5: In electrical engineering literature, a cut is also referred to as a ***seg*** [Re61].

R6: Proofs of most results in this section may be found in standard texts [SeRe61], [Ch71b], [De74], [ThSw92], and [SwTh81].

6.4.2 The Circuit Subspace in an Undirected Graph

DEFINITIONS

D19: A graph is ***even*** if the degree of every vertex in the graph is even. Clearly, a circuit is an even graph.

D20: A ***circ*** of a graph is a closed trail. The null graph is considered as a circ.

NOTATION: The set of all circs of a graph G is denoted by $\hat{C}(G)$. In other words, $\hat{C}(G)$ is the set of all circuits and unions of edge-disjoint circuits of the graph G (including the null graph $\emptyset$).

FACTS

F4: A subgraph of a graph is a circ if and only if it is even.

F5: A circ is a circuit or union of edge-disjoint circuits. Thus, the edge set of an even graph can be partitioned into edge subsets such that each subset in the partition forms a circuit.

F6: The ring sum of any two even subgraphs of a graph is even. Thus, the set $\hat{C}(G)$ is closed under ring sum.

F7: $\hat{C}(G)$ is a subspace of the vector space $\Psi(G)$ and is called the ***circuit subspace*** of G.

EXAMPLE

E6: Two circs of the graph G of Figure 6.4.1 and their ring sum, which is clearly a circ, are shown in Figure 6.4.6.

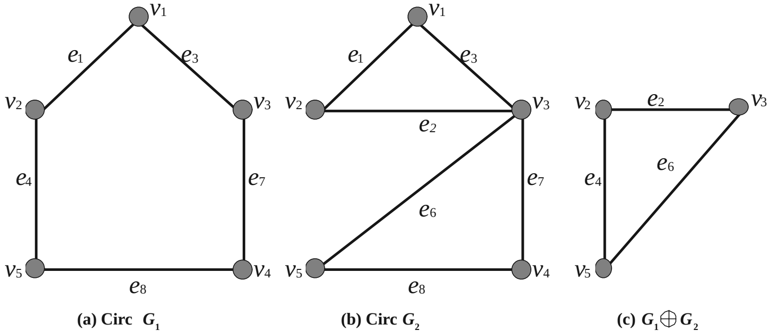


Figure 6.4.6 **Two circs of the graph G and their ring sum.**

REMARKS

R7: Fact 5 is attributed to Veblen [Ve31].

R8: A connected, even graph G is *eulerian*, i.e., there exists a circ that contains all the edges of G (see §4.2).

Fundamental Circuits and the Dimension of the Circuit Subspace

DEFINITION

D21: Adding a chord c to a spanning tree T of a connected graph G produces a unique circuit in G, called the ***fundamental circuit*** of G with respect to chord c.

NOTATION: If e_i is a chord of a spanning tree T, then C_i will denote the fundamental circuit with respect to e_i.

FACTS

F8: Given a connected graph G and a spanning tree T, there are $m-n+1$ fundamental circuits, one for each chord of T.

F9: The fundamental circuit with respect to chord c contains only one chord of the spanning tree T, namely, the chord c. The chord c is not present in any other fundamental circuit with respect to T.

F10: The $(m - n + 1)$ fundamental circuits with respect to a spanning tree of a connected graph G are linearly independent in the circuit subspace $\hat{C}(G)$.

F11: If a circ of a graph G contains the chords $e_a, e_b, \ldots, e_k$, then the circ can be expressed as the ring sum of the fundamental circuits $C_a, C_b, \ldots, C_k$.

F12: The fundamental circuits with respect to a spanning tree of a connected graph G constitute a basis for the circuit subspace $\hat{C}(G)$, and hence, the dimension of $\hat{C}(G)$ is equal to $m - n + 1$, the nullity $\mu(G)$.

F13: The dimension of the circuit subspace $\hat{C}(G)$ of a graph having p components is equal to $\mu(G) = m - n + p$.

EXAMPLE

E7: The set of fundamental circuits with respect to the spanning tree $T = \{e_1, e_2, e_4, e_7\}$ of the graph shown in Figure 6.4.1 is

Chord e_3	$C_3 = \{e_3, e_1, e_2\}$
Chord e_5	$C_5 = \{e_5, e_1, e_4\}$
Chord e_6	$C_6 = \{e_6, e_2, e_4\}$
Chord e_8	$C_8 = \{e_8, e_2, e_4, e_7\}$

It can be verified that the circ $\{e_1, e_4, e_5, e_6, e_7, e_8\}$, which contains the chords e_5, e_6 and e_8, is the ring sum of the fundamental circuits C_5, C_6, and C_8. This illustrates Fact 11.

6.4.3 The Cutset Subspace in an Undirected Graph

Recall from the definitions in §6.4.1 that a cutset is also a cut. Several facts that highlight the duality between cuts and circs will be presented next.

DEFINITION

D22: The collection of all cutsets and unions of edge-disjoint cutsets of a graph G is called the ***cutset subspace*** of G and is denoted by $\lambda(G)$. The null graph $\emptyset$ is considered a cut and hence belongs to $\lambda(G)$.

FACTS

F14: Every cut of a connected graph G is the union of some edge-disjoint cutsets of G. Thus, $\lambda(G)$ is the collection of cuts of G.

F15: The ***cutset subspace*** $\lambda(G)$ of a graph G is a subspace of the vector space $\Psi(G)$.

F16: The ring sum of any two cuts of a graph G is also a cut of G; i.e., $\lambda(G)$ is closed under ring sum.

EXAMPLE

E8: Consider the graph in Figure 6.4.1 and the cuts $S_1 = \langle V_1, V_2\rangle$ and $S_2 = \langle V_3, V_4\rangle$ in Figure 6.4.5, where $V_1 = \{v_1, v_3, v_5\}$, $V_2 = \{v_2, v_4\}$, $V_3 = \{v_1, v_2, v_3\}$ and $V_4 = \{v_4, v_5\}$. Then $S_1 = \{e_1, e_2, e_4, e_7, e_8\}$, $S_2 = \{e_4, e_5, e_6, e_7\}$, and $S_1 \oplus S_2 = \{e_1, e_2, e_5, e_6, e_8\}$.

Moreover, it can be seen that $S_1 \oplus S_2 = \langle A \cup D, B \cup C\rangle$, where

$$A = V_1 \cap V_3 = \{v_1, v_3\},$$
$$B = V_1 \cap V_4 = \{v_5\},$$
$$C = V_2 \cap V_3 = \{v_2\},$$
$$D = V_2 \cap V_4 = \{v_4\}.$$

In fact, this illustration is also the basis of the proof of Fact 16.

Fundamental Cutsets and the Dimension of the Cutset Subspace

DEFINITIONS

D23: Let T be a spanning tree of a connected graph G, and let b be a branch of T. If V_1 and V_2 are the vertex-sets of the two components of $T - b$, then we can verify that the cut $\langle V_1, V_2\rangle$ is a cutset of G. This cutset is called the ***fundamental cutset*** of G with respect to the branch b of T.

NOTATION: If e_i is a branch of a spanning tree T, then S_i denotes the fundamental cutset with respect to the branch e_i.

D24: An ***incidence set*** of a vertex v in a graph G is the cut consisting of the set of edges of G that are incident on v.

FACTS

F17: Given a connected graph G and a spanning tree T, there are $n - 1$ fundamental cutsets, one for each branch of T.

F18: The fundamental cutset with respect to branch b of a spanning tree T contains only one branch, namely, the branch b. The branch b is not present in any other fundamental cutset with respect to T.

F19: The $n - 1$ fundamental cutsets with respect to a spanning tree of a connected n-vertex graph G are linearly independent in the cutset subspace $\lambda(G)$.

F20: If a cut of a graph G contains the branches $e_a, e_b, \ldots e_k$, then the cut can be expressed as the ring sum of the fundamental cutsets $\mathrm{S}_{a,} S_b, \ldots, S_k$.

F21: The fundamental cutsets with respect to a spanning tree of a connected graph G constitute a basis for the cutset subspace $\lambda(G)$ of G, and hence the dimension of $\lambda(G)$ is equal to $n - 1$, the rank $\rho(G)$.

F22: The dimension of the cutset subspace $\lambda(G)$ of a graph having p components is equal to $\rho(G) = n - p$.

F23: The incidence sets of any $n - 1$ vertices of a connected n-vertex graph G form a basis of the cutset subspace $\lambda(G)$.

EXAMPLE

E9: For the graph shown in Figure 6.4.1, the fundamental cutsets with respect to the spanning tree $T = \{e_1, e_2, e_4, e_7\}$ are

Branch e_1	$S_1 = \{e_1, e_3, e_5\}$,
Branch e_2	$S_2 = \{e_2, e_3, e_6, e_8\}$,
Branch e_4	$S_4 = \{e_4, e_5, e_6, e_8)$,
Branch e_7	$S_7 = \{e_7, e_8\}$.

It can be verified that the cut $= \{e_1, e_2, e_4, e_7, e_8\}$ containing the branches e_1, e_2, e_4, and e_7 is the ring sum of the fundamental cutsets S_1, S_2, S_4, and S_7. This illustrates Fact 20.

6.4.4 Relationship between Circuit and Cutset Subspaces

By now it should be evident that circs and cuts are dual concepts in the sense that for each result that involves circuits or circs, there is a corresponding result involving cutsets or cuts. Facts 5 through 13 correspond to Facts 14 through 22. Spanning trees and cospanning trees provide the links between circs and cuts. This duality is further explored next.

Orthogonality of Circuit and Cutset Subspaces

DEFINITIONS

D25: The binary m-vector representing a circ is called a ***circuit vector***; the binary m-vector representing a cut is called a ***cut vector***; and the m-vector representing an incidence set is called an ***incidence vector***.

D26: Two subspaces W' and W'' of a vector space W are ***orthogonal*** to each other if the inner product (or dot product) of every vector in W' with every vector in W'' is zero. Note that the zero vector belongs to every subspace.

FACTS

F24: A circuit and a cutset of a connected graph have an even number of edges in common. Hence, a circ and a cut have an even number of edges in common.

F25: The inner product of a circuit vector and a cut vector over GF(2) is zero under the ring sum operation.

F26: A subgraph of a graph G belongs to the circuit subspace of the graph if and only if it has an even number of edges in common with every subgraph in the cutset subspace of G. Equivalently, a vector is a circuit vector if and only if it is orthogonal to every cut vector.

F27: A subgraph of a graph G belongs to the cutset subspace of the graph if and only if it has an even number of edges in common with every subgraph in the circuit subspace of G. Equivalently, a vector is a cut vector if and only if it is orthogonal to every circuit vector.

F28: The circuit and cutset subspaces of a graph are orthogonal to each other.

Circ/Cut-Based Decomposition of Graphs and Subgraphs

DEFINITION

D27: Two orthogonal subspaces W' and W'' of a vector space W are ***orthogonal complements*** if every vector in W can be expressed as the ring sum of a vector of W' and a vector of W''. Note that the zero vector is the only vector that is in the intersection of the orthogonal complements W' and W''.

FACTS

F29: If the orthogonal subspaces W' and W'' of a vector space W are not orthogonal complements, then the dimension of their union is less than the dimension of the vector space W.

F30: [Ch71a] The circuit and the cutset subspaces of a graph are orthogonal complements if and only if the graph has an odd number of spanning forests.

F31: If the circuit and cutset subspaces of a graph are orthogonal complements, then every subgraph (including the graph itself) can be expressed as the ring sum of a circ and a cut.

F32: [Ch71b, WiMa71] Every graph can be represented as the ring sum of a circ and a cut of the graph. If the dimension of the intersection of the circuit and cutset subspaces of a graph is equal to k, then there are 2^k such representations.

EXAMPLES

E10: Consider the graph G_a in Figure 6.4.7. It can be verified that no nonempty subgraph of this graph is both a circ and a cut. So the cutset and circuit subspaces of G_a are orthogonal complements. Then the set of fundamental cutsets and fundamental circuits with respect to a spanning tree of G_a constitutes a basis of the vector space $\Psi(G)$. One such set with respect to the spanning tree formed by the edges e_1, e_2, e_3, and e_4 is as follows:

$$\begin{array}{ll}
S_1 & = (1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0) \\
S_2 & = (0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0) \\
S_3 & = (0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1) \\
S_4 & = (0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1) \\
C_5 & = (1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0) \\
C_6 & = (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0) \\
C_7 & = (0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1)
\end{array}$$

It is easy to verify that every subgraph can be expressed as the ring sum of a circ and a cut, which illustrates Fact 31.

For instance, the vector (0 0 1 1 0 1 1), which represents the induced subgraph on the edge subset $\{e_3, e_4, e_6, e_7\}$, can be expressed as:

$$\begin{aligned}(0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1) &= S_1 \oplus S_2 \oplus C_6 \oplus C_7 \\ &= (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0) \oplus (1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1)\end{aligned}$$

where (1 1 0 0 0 0 0) represents a cut in G_a, and (1 1 1 1 0 1 1) represents a circ.

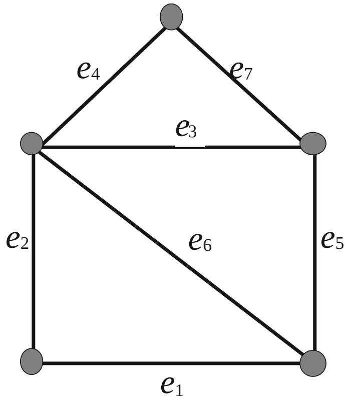


Figure 6.4.7 Graph G_a for illustration of Fact 31.

E11: Consider the graph G_b in Figure 6.4.8. In this graph the edges e_1, e_2, e_3, and e_5 constitute a circuit as well as a cut. Hence the circuit and cutset subspaces are not orthogonal complements. This means that there is a subgraph of G_b that cannot be expressed as the ring sum of a circ and a cut. However, according to Fact 32, such a decomposition is possible for G_b. This is verified as follows:

$$(1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1) \quad = \quad (1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0) \oplus (0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1)$$

where (1 1 0 1 0 0) represents the cut of edges e_1, e_2, and e_4, and (0 0 1 0 1 1) represents the circuit of edges e_3, e_5, and e_6 in G_b.

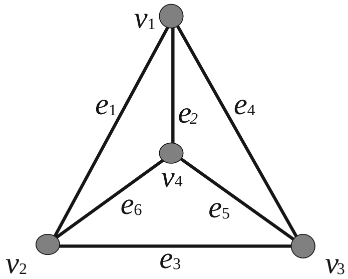


Figure 6.4.8 Graph G_b for illustration of Fact 32.

6.4.5 The Circuit and Cutset Spaces in a Directed Graph

In most engineering applications of graph theory, directed graphs are encountered. But, as we shall see next, the effect of orientation is minimal in so far as the results concerning circuits and cuts are concerned. Almost all the results presented earlier in this section have their equivalents in the directed case. In fact, we can view all the results on undirected graphs presented thus far as special cases of the results to be presented next.

TERMINOLOGY: A *circuit*, *cut*, or *spanning tree* in a directed graph G is a subset of edges that constitutes a circuit, cut, or spanning tree, respectively, in the underlying graph of G.

Circuit and Cut Vectors and Matrices

DEFINITIONS

D28: A circuit in a directed graph can be traversed in one of two directions, clockwise or counter-clockwise (relative to a plane drawing of the circuit). The traversal direction we choose is called the ***circuit orientation***.

D29: Let C be a circuit in a directed graph and $e = (v_i, v_j)$ an edge in C directed from v_i to v_j. Given an orientation of C, edge e is said to ***agree with the circuit orientation*** if the traversal of e specified by that orientation is from its tail v_i to its head v_j.

D30: A cut (V_a, V_b) in a directed graph can be traversed in one of two directions, from V_a to V_b or from V_b to V_a. The direction chosen is called the ***cut orientation***.

D31: Given an orientation of a cut in a directed graph, an edge $e = (v_{i,} v_j)$ in the cut is said to ***agree with the cut orientation*** if the traversal of e specified by that orientation is from v_i to v_j.

D32: Let G be a directed graph with edge-set $E = \{e_1, e_2, \ldots, e_m\}$, and let C be a circuit in G with a given orientation. The ***circuit vector*** representing C is the m-vector $(x_1, x_2, \ldots, x_m)$, where

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ agrees with the orientation of } C \\ -1, & \text{if edge } e_i \text{ does not agree with the orientation of } C \\ 0, & \text{if edge } e_i \text{ is not in } C \end{cases}$$

D33: Let G be a directed graph with edge-set $E = \{e_1, e_2, \ldots, e_m\}$, and let S be a cut in G with a given orientation. The ***cut vector*** representing S is the m-vector $(x_1, x_2, \ldots, x_m)$, where

$$x_i = \begin{cases} 1, & \text{if edge } e_i \text{ agrees with the orientation of } S \\ -1, & \text{if edge } e_i \text{ does not agree with the orientation of } S \\ 0, & \text{if edge } e_i \text{ is not in } S \end{cases}$$

D34: Let G be a directed graph with edge-set $E = \{e_1, e_2, \ldots, e_m\}$. Let $C_1, C_2, \ldots, C_t$ and $S_1, S_2, \ldots, S_r$ be the circuits and cuts of G, respectively, each with a given traversal orientation. The ***circuit matrix*** of G is the $t \times m$ matrix whose ith row is the circuit vector representing circuit C_i. The ***cut matrix*** of G is the $r \times m$ matrix whose ith row is the cut vector representing cut S_i.

The Fundamental Circuit, Fundamental Cutset, and Incidence Matrices

Next, we define two special matrices corresponding to the fundamental circuits and cutsets relative to a given spanning tree in a directed graph and a third matrix corresponding to the incidence vectors of the vertices.

REMARK

R9: The definitions of these three matrices depend on how the associated circuits and cuts are oriented. The orientations of each fundamental circuit and each fundamental cut are usually chosen to agree with the defining chord and branch, respectively, and we adopt that convention here. Also, for the cut consisting of the set of edges incident on a vertex v (i.e., the incidence set of v), we assume that the orientation is away from vertex v. Accordingly, the incidence vector of vertex v is given by $(x_1, x_2, \ldots, x_m)$, where

$$ {}_i = \begin{cases} 1, & \text{if edge } e_i \text{ is directed } \textit{from } v \quad (v \text{ is the tail of edge } e_i) \\ -1, & \text{if edge } e_i \text{ is directed } \textit{to } v \quad (v \text{ is the head of edge } e_i) \\ 0, & \text{if edge } e_i \text{ is not incident on } v \end{cases} $$

DEFINITIONS

D35: Let T be a spanning tree of a connected directed graph. The ***fundamental circuit matrix*** of the graph with respect to T, denoted by $\boldsymbol{B}_f$, is the $(m - n + 1)$-rowed submatrix of the circuit matrix whose rows are the fundamental circuit vectors. Similarly, the ***fundamental cutset*** matrix with respect to T, denoted by Q_f, is the $(n-1)$-rowed submatrix of the cut matrix whose rows are the fundamental cutset vectors.

D36: The ***incidence matrix*** of a given directed graph, denoted A_c, is the n-rowed submatrix of the cut matrix whose rows are the incidence vectors of the directed graph. The submatrix of the incidence matrix containing any $n - 1$ of the incidence vectors is called a ***reduced incidence matrix*** and is denoted by A.

D37: A matrix of real numbers is ***unimodular*** if the determinant of every square submatrix of the matrix is equal to 1, -1 or 0.

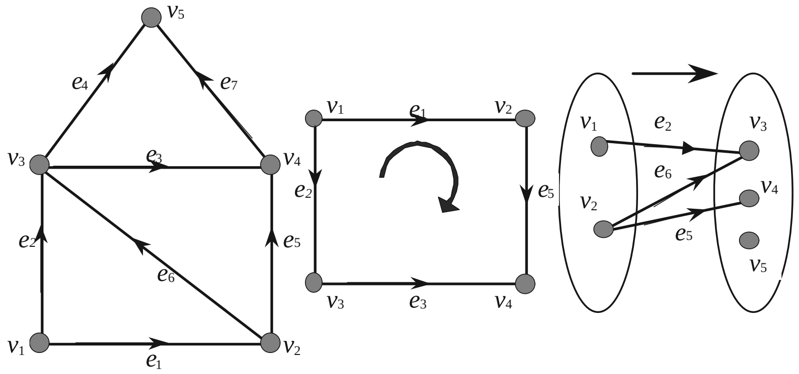
EXAMPLES

E12: Consider the directed graph of Figure 6.4.9(a) below. A circuit and a cut with orientations are shown in Figures 6.4.9(b) and (c), respectively. The corresponding circuit and cut vectors are $(1 \quad -1 \quad -1 \quad 0 \quad 1 \quad 0 \quad 0)$ and $(0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0)$, respectively.

E13: Consider the spanning tree T of the graph of Figure 6.4.9(a) consisting of the edges e_1, e_2, e_3, and e_4. The fundamental circuit and the fundamental cutset matrices with respect to T, and the incidence matrix of this graph with the column i in each matrix corresponding to edge e_i are:

Fundamental Circuit Matrix:

$$ \begin{array}{l} \text{Chord } e_5 \\ \text{Chord } e_6 \\ \text{Chord } e_7 \end{array} \begin{pmatrix} 1 & -1 & -1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 \end{pmatrix} $$



(a) A directed graph. (b) A circuit with orientation (c) A cut with orientation

Figure 6.4.9 **A directed graph, a circuit, and a cut with orientations.**

Fundamental Cutset Matrix:

$$\begin{array}{r} \text{Branch } e_1 \\ \text{Branch } e_2 \\ \text{Branch } e_3 \\ \text{Branch } e_4 \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Incidence Matrix:

$$\begin{array}{r} \text{Node } v_1 \\ \text{Node } v_2 \\ \text{Node } v_3 \\ \text{Node } v_4 \\ \text{Node } v_5 \end{array} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 \end{pmatrix}$$

Orthogonality and the Matrix Tree Theorem

TERMINOLOGY: A directed edge that is in both a circuit and a cut is said to have the ***same relative orientation*** with respect to the circuit and the cut if the edge either agrees or disagrees with the assigned orientations of both the circuit and the cut.

FACTS

F33: A circuit and a cut in a connected graph have an even number of common edges. If a circuit and a cut have $2k$ common edges, then these edges can be partitioned into two sets, each of size k, such that each edge in one set has the same relative orientation with respect to the circuit and the cut, and that each edge in the other set agrees with one of the two assigned orientations (circuit or cut) and disagrees with the other assigned orientation.

NOTATION: Let G be a directed graph and suppose that each of the circuits and cuts has been given an orientation.
(a) The collection of all circuit vectors of G and their linear combinations over the real field is denoted by $\hat{C}(G)$.
(b) The collection of all cut vectors of G and their linear combinations over the real field is denoted by $\lambda(G)$.

F34: In a directed graph every circuit vector is orthogonal to every cut vector over the real field.

F35: In a connected directed graph, every circuit vector can be expressed as a linear combination of fundamental circuit vectors with respect to a spanning tree of the graph. The coefficients in the linear combination are 1 or -1. Similarly, every cut vector in a connected directed graph can be expressed as a linear combination of fundamental cutset vectors with respect to a spanning tree of the graph. The coefficients in the linear combination are 1 or -1.

F36: In a directed graph G, $\hat{C}(G)$ and $\lambda(G)$ are vector spaces over the real field and are orthogonal to each other. $\hat{C}(G)$ and $\lambda(G)$ are called the ***circuit space*** and the ***cutset space***, respectively.

F37: The fundamental circuit vectors and the fundamental cutset vectors with respect to a spanning tree of a connected directed graph G form a basis of the circuit space and a basis of the cutset space, respectively. The dimension of the circuit space is equal to $m-n+p$, the nullity of G, and the dimension of the cutset space is equal to $n-p$, the rank of G, where p is the number of components of G.

F38: Any set of $n-1$ incidence vectors of a connected directed graph forms a basis of the cutset space of the graph.

F39: The fundamental cutset and the fundamental circuit matrices of a connected directed graph are unimodular.

F40: Consider a spanning tree T of a connected directed graph G with branches $b_1, b_2, \ldots\ldots, b_{n-1}$ and chords $c_1, c_2, c_3, \ldots, c_{m-n+1}$. Suppose that the edges of G are labeled so that $e_1, e_2, \ldots, e_m = b_1, b_2, \ldots, b_{n-1}, c_1, c_2, \ldots, c_{m-n+1}$, respectively. Then the fundamental circuit matrix B_f has the form $B_f = [B_{ft}|U_{m-n+1}]$, where U_{m-n+1} is the identity matrix of size $m-n+1$ and B_{ft} is the submatrix of B_f consisting of the columns corresponding to the branches $b_1, b_2, \ldots, b_{n-1}$ of T. Similarly, the fundamental cutset matrix Q_f has the form $Q_f = [U_{n-1}|Q_{fc}]$, where U_{n-1} is the identity matrix of size $n-1$ and Q_{fc} is the submatrix of Q_f consisting of the columns corresponding to the chords $c_1, c_2, \ldots, c_{m-n+1}$ of T. Moreover, $Q_{fc} = -B_{ft}^t$.

F41: The columns of the cut matrix of a connected directed graph G are linearly independent if and only if they correspond to the branches of a spanning tree. Similarly, the columns of the circuit matrix are linearly independent if and only if they correspond to the chords of a cospanning tree.

F42: (*Matrix Tree Theorem*) For a connected directed graph, each cofactor of the matrix $A_c A_c^t$ equals the number of spanning trees of the graph.

EXAMPLE

E14: The matrices in Example 13 illustrate Facts 33 through 41.

REMARKS

R10: By simply replacing -1 by 1 in all the matrices defined for directed graphs, we get the corresponding matrices for undirected graphs.

R11: The rank and the nullity of the cut matrix of a connected graph are $(n-1)$ and $(m-n+1)$, respectively. This motivated the definitions of the rank and nullity of a graph (see Definition 13).

R12: The matrix $A_c A_c^t$ is called the *degree matrix* of the graph. It can be verified that the diagonal entry (i, i) of the degree matrix is equal to the degree of vertex v_i and the off-diagonal entry (i, j) is equal to the negative of the number of edges connecting vertex v_i and vertex v_j (regardless of the orientations of these edges). A proof of Fact 42 may be found in [ThSw92]. A weighted version of the degree matrix plays an important role in electrical circuit analysis [SwTh81].

Minty's Painting Theorem

TERMINOLOGY: Two directed edges in a circuit or cutset are said to have the ***same direction*** (relative to that circuit or cutset) if both edges agree with the same orientation of that circuit or cutset.

DEFINITIONS

D38: A ***directed circuit*** is a circuit whose edges all have the same direction relative to it.

D39: A ***directed cutset*** is a cutset whose edges all have the same direction relative to it.

D40: A ***painting*** of a directed graph G is a partitioning of the edges of the graph into three sets R, Y, and B and the distinguishing of one element of the set Y. We can visualize this as coloring of the edges of G with three colors, each edge being painted red, yellow, or blue, and exactly one yellow edge being colored dark yellow.

FACTS

F43: (*Painting Theorem*) [Mi66] Let G be a directed graph. For any painting of the edges of G, exactly one of the following holds:

1. There exists a circuit containing the dark yellow edge but no blue edges, in which all the yellow edges have the same direction as the dark yellow edge.
2. There exists a cutset containing the dark yellow edge but no red edges, in which all the yellow edges have the same direction as the dark yellow edge.

F44: Each edge of a directed graph is in a directed circuit or in a directed cutset, but no edge belongs to both.

REMARK

R13: Minty's painting theorem (also known as the ***arc coloring lemma***) has profound applications in electrical circuit theory. This theorem is also true for orientable matroids (see [ThSw92]). Fact 44 is a corollary of Fact 43. Other related works by Minty of considerable significance in electrical circuit theory are [Mi60, Mi61]. Some applications of the arc coloring lemma to problems in electrical circuit theory may be found in [VaCh80, ChGr76, Wo70].

6.4.6 Two Circ/Cut-Based Tripartitions of a Graph

In §6.4.4 we presented a result on the decomposition of a graph into a circ and a cut. But such circs and cuts may not be disjoint and hence they may not form a partition of the edge set of the graph. We now present two ways to partition a graph. These partitions are both tripartitions and are again based on circs and cuts.

Bicycle-Based Tripartition

DEFINITION

D41: A subgraph that is in the intersection of the circuit and cutset subspaces of an undirected graph is called a ***bicycle***. That is, a bicycle is a circ as well as a cut.

EXAMPLE

E15: The edges e_1, e_2, e_3, and e_5 in the graph of Figure 6.4.8 form both a cut and a circuit.

FACT

F45: [RoRe78] Any edge e of a graph G is of one of the following types:

1. e is in a circ that becomes a cut when e is removed from it.
2. e is in a cut that becomes a circ when e is removed from it.
3. e is in a bicycle.

TERMINOLOGY: The partition of the edges defined by Fact 45 is called the ***bicycle-based tripartition***.

REMARK

R14: Rosenstiehl and Read [RoRe78] have proved several interesting results relating to circuits and cuts and their relationship. A proof of Fact 45 may also be found in [Pa94].

A Tripartition Based on Maximally Distant Spanning Trees

DEFINITIONS

D42: The ***tree distance***, $d(T_1, T_2)$, between any two spanning trees T_1 and T_2 is defined as $d(T_1, T_2) = |E(T_1) - E(T_2)| = |E(T_2) - E(T_1)|$.

D43: Two spanning trees T_1 and T_2 are ***maximally distant*** if $d(T_1, T_2) \geq d(T_i, T_j)$ for every pair of spanning trees T_i and T_j.

NOTATION: The maximum distance between any two spanning trees of a connected graph is denoted by d_m.

D44: Given a pair of maximally distant spanning trees T_1 and T_2 of a connected graph G. Suppose c is a common chord of T_1 and T_2. The ***k-subgraph*** $\boldsymbol{G_c}$ of G with respect to c is the edge-induced subgraph constructed as follows:

1. Let L_1 be the set of all the edges in the fundamental circuit with respect to T_1 defined by c.
2. Let L_2 be the union of the sets of edges in all the fundamental circuits with respect to T_2 defined by every edge in L_1.

3. Repeating the above, we can obtain a sequence of sets of edges L_1, $L_2, \ldots$ until we arrive at a set $L_{k+1} = L_k$. Then the induced subgraph on the edge set L_k is called the *k-subgraph* G_c with respect to c.

D45: The ***k*-subgraph G_b** with respect to a common branch b can be constructed in a dual manner as in Definition 44.

D46: The ***principal subgraph*** G_1 with respect to the common chords (of a pair of maximally distant spanning trees T_1 and T_2) is the union of the k-subgraphs with respect to all the common chords. The ***principal subgraph*** G_2 with respect to the common branches is the union of the k-subgraphs with respect to all the common branches.

FACTS

F46: [KiKa69] Let T_1 and T_2 form a pair of maximally distant spanning trees of a connected graph G.

1. The fundamental circuit of G with respect to T_1 or T_2 defined by a common chord of T_1 and T_2 contains no common branches of these spanning trees.
2. The fundamental cutset of G with respect to T_1 or T_2 defined by a common branch of T_1 and T_2 contains no common chords of these spanning trees.

F47: [KiKa69] Consider a graph $G = (V, E)$. Let E_1 and E_2 denote the edge-sets of the principal subgraphs G_1 and G_2, respectively, and let $E_0 = E(G) - (E_1 \cup E_2)$. Then E_0, E_1, and E_2 form a partition of the edge-set $E(G)$. The partition (E_0, E_1, E_2) is called the ***principal partition*** of G and is independent of the maximally distant trees used to construct it.

EXAMPLE

E16: It can be verified that $T_1 = \{e_2, e_{3,} e_4, e_7\}$ and $T_2 = \{e_1, e_{3,} e_5, e_6\}$ are a pair of maximally distant spanning trees for the graph in Figure 6.4.10 and that the associated principal partition is: $E_1 = \{e_6, e_7, e_8\}$, $E_2 = \{e_1, e_2, e_3\}$, and $E_0 = \{e_4, e_5\}$.

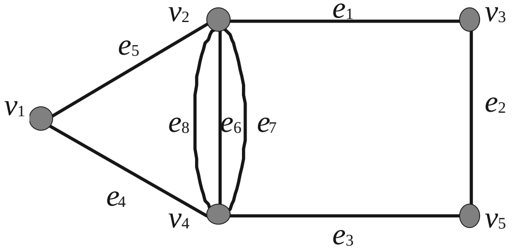


Figure 6.4.10

REMARKS

R15: In electrical circuit analysis one is interested in solving for all the current and the voltage variables. The circuit method of analysis (also known as the loop analysis) requires solving for only $m-n+1$ independent current variables. The remaining current variables and all the voltage variables can then be determined using these $m - n + 1$ independent current variables. The cutset method of analysis requires solving for only $n - 1$ independent voltage variables. A question that intrigued circuit theorists for a long time was whether one could use a hybrid method of analysis involving some current variables and some voltage variables and reduce the size of the system of equations to be solved to less than both $n-1$ and $m-n+1$, the rank and nullity of the graph of the circuit. Ohtsuki, Ishizaki, and Watanabe [OhIsWa70] studied this problem and showed that d_m, the maximum distance between any two spanning trees of the graph of the circuit is, in fact, the minimum number of variables required in the hybrid method of analysis. They also showed that the variables can be determined using the principal partition of the graph. The works by Kishi and Kajitani [KiKa69] on principal partition and by Ohtsuki, Ishizaki, and Watanabe [OhIsWa70] on the hybrid method of analysis are considered landmark results in electrical circuit theory. Swamy and Thulasiraman [SwTh81] give a detailed exposition of the principal partition concept and the hybrid and other methods of circuit analysis.

R16: Lin [Li76] presented an algorithm for computing the principal partition of a graph. Bruno and Weinberg [BrWe71] extended the concept of principal partition to matroids.

6.4.7 Realization of Circuit and Cutset Spaces

In the application of graph theory to the electrical circuit synthesis problem, one encounters a certain matrix of integers modulo 2 and seeks to determine if this matrix is the cutset or the circuit matrix of an undirected graph. The complete solution to this problem was given by Tutte [Tu59]. Cederbaum [Ce58] and Gould [Go58] considered this problem before Tutte provided the solution. We now present the main result on the necessary and sufficient conditions for the realizability of a matrix of integers modulo 2 as the circuit or the cutset matrix of an undirected graph. Related results leading to this main result are also presented. Seshu and Reed [SeRe61] discuss these results in considerable detail, except for the proof of the sufficiency of Tutte's realizability condition.

DEFINITIONS

D47: The graphs in Figure 6.4.11 are called **Kuratowski graphs**.

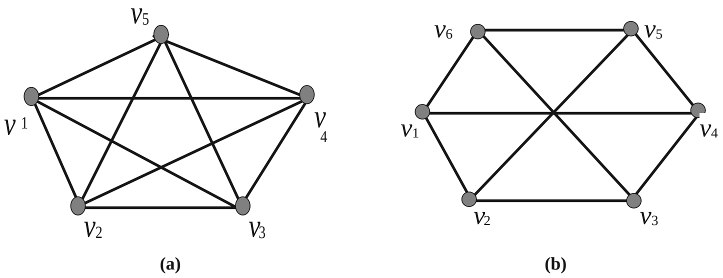


Figure 6.4.11 The two Kuratowski graphs.

D48: A matrix F of the form $F = [F' \mid U]$, where U is the identity matrix, is said to be in ***normal form***.

D49: A matrix F of real integers in normal form is a ***regular matrix*** if for every linear combination X of the rows of M with coefficients -1, 1, and 0 we have the following:

1. The elements of X are 1, -1, and 0, or
2. There exists another such linear combination Y (with coefficients 1, -1, and 0) that has 1 and -1 for nonzero elements and these are at a (not necessarily proper) subset of the positions in which X has nonzero elements.

D50: A matrix of integers mod 2 is ***regular*** if the replacement of a suitable set of 1's by -1's makes it regular.

FACTS

F48: For a connected directed graph G, the fundamental cutset and fundamental circuit matrices with respect to a spanning tree T of G and the reduced incidence matrix A of G are all regular matrices.

F49: A regular matrix in normal form is unimodular.

F50: Given a regular matrix F of integers 1, -1, and 0 in normal form, the replacement of -1's by 1's will leave the ranks of the submatrices unaltered (where the rank of the derived matrix is with respect to modulo 2 arithmetic).

F51: A matrix F of integers mod 2 is regular if and only if no normal form of F contains either the matrix N_0 or its transpose, where

$$N_0 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

F52: A matrix F of integers mod 2 is realizable as the cutset matrix of an undirected graph if and only if it is regular and no normal form of F contains the circuit matrix of either of the two Kuratowski graphs shown in Figure 6.4.11.

F53: A matrix F of integers mod 2 in normal form is realizable as the circuit matrix of an undirected graph if and only if it is regular and no normal of F contains the cutset matrix of either of the two Kuratowski graphs shown in Figure 6.4.11.

REMARKS

R17: Mayeda [Ma70] gave an alternate proof of Tutte's realizability condition, shorter than Tutte's original proof, which is 27 pages long.

R18: Early works on algorithms for constructing graphs having specified circuit or cutset matrices are in [Tu60, Tu64]. Bapeswara Rao [Ba70] defined the tree-path matrix of an undirected graph which is essentially the non-unit submatrix of the fundamental circuit matrix and presented an algorithm for constructing a graph with a prescribed tree-path matrix. This is also an algorithmic solution to the cutset and the circuit matrix realization problems. A detailed presentation of Bapeswara Rao's algorithm is given in [SwTh81].

R19: The circuit and the cutset matrix realization problems arise in the design of multi-port resistance networks. It was in the context of this application that Cederbaum [Ce58, Ce59] encountered the realization problem. Interestingly, Bapeswara Rao [Ba70] and Boesch and Youla [BoYo65] presented circuit-theoretic approaches to the realization of a matrix as the cutset or circuit matrix of a directed graph. Details of Bapeswara Rao's algorithm based on this approach may also be found in [SwTh81].

Whitney and Kuratowski

We believe that it is appropriate to conclude this section with a reference to two classic works by Whitney [Wh33] and Kuratowski [Ku30] relating to duality. While the results of this section bring out the duality between circuits and cutsets, Whitney introduced the concept of duality between graphs. His original definition was an algebraic one (see also [ThSw92]) relating the nullity and rank of certain corresponding subgraphs of dual graphs. Definition 51 is an equivalent one.

DEFINITION

D51: A graph G_2 is a ***dual*** of a graph G_1 if there is a one-to-one correspondence between their edge-sets such that a set of edges in G_2 is a circuit vector of G_2 if and only if the corresponding set of edges in G_1 is a cutset vector of G_1.

FACTS

F54: It follows from the duality between circuits and cutsets that if G_2 is a dual of G_1, then G_1 is a dual of G_2.

F55: [Wh33] A graph has a dual if and only if it is planar.

F56: In another classic work, Kuratowski [Ku30] proved that a graph is planar if and only if it does not contain a subdivision of a Kuratowski graph.

REMARK

R20: See [We01] for a proof of Kuratowski's theorem. It is quite interesting to see the role of the Kuratowski graphs in Tutte's realizability conditions for the cutset and the circuit matrix realization problems.

References

[Ba70] V. V. Bapeswara Rao, *The Tree-Path Matrix of a Network and Its Applications*, Ph.D. Thesis, Department of Electrical Engineering, Indian Institute of Technology, Madras, India, 1970.

[BoYo65] F. T. Boesch and D. C. Youla, Synthesis of resistor n-port networks, *IEEE Trans. Circuit Theory* 12 (1965), 515–520.

[BrWe71] J. Bruno and L. Weinberg, The principal minors of a matroid, *Linear Algebra and Its Applications* 4 (71), 17–54.

[Ce58] I. Cederbaum, Matrices all of whose elements and subdeterminants are 1, -1 or 0, *J. Math. and Phys.* 36 (58), 351–361.

[Ce59] I. Cederbaum, Applications of matrix algebra to network theory, *IRE Trans. Circuit Theory* 6 (1959), 127–137.

[Ch71a] W. -K. Chen, On vector spaces associated with a graph, *SIAM J. Appl. Math.* 20 (1971), 526–529.

[Ch71b] W. -K. Chen, *Applied Graph Theory*, North Holland, Amsterdam, 1971.

[ChGr76] L. O. Chua and D. M. Greene, Graph-theoretic properties of dynamic nonlinear networks, *IEEE Trans. Circuits and Systems* 23 (1976), 292–312.

[De74] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, 1974.

[Eu36] L. Euler, Solutios Problematis ad Geometriam Situs Pertinantis, *Academimae Petropolitanae* 8 (1736), 128–140.

[Go58] R. L. Gould, Graphs and vector spaces, *J. Math. and Phys.* 38 (1958), 193–214.

[GrYe99] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*, CRC Press, 1999.

[KiKa69] G. Kishi and Y. Kajitani, Maximally distant trees and principal partition of a linear graph, *IEEE Trans. Circuit Theory* 16 (1969), 323–330.

[Ku30] C. Kuratowski, Sur le probleme des Courbes Gauches en topologie, *Fund. Math.* 15 (1930), 271–283.

[Li76] P. M. Lin, An improved algorithm for principal partition of graphs, *Proc. IEEE Intl. Symp. Circuits and Systems* (1976), 145–148.

[Ma70] W. Mayeda, A proof of Tutte's realizability condition, *IEEE Trans. Circuit Theory* 17 (1970), 506–511.

[Mi60] G. J. Minty, Monotone networks, *Proc. Roy. Soc., A* 257(1960), 194–212.

[Mi61] G. J. Minty, Solving steady-state nonlinear networks of 'monotone' elements, *IRE. Trans. Circuit Theory* 8 (1961), 99–104.

[Mi66] G. J. Minty, On the axiomatic foundations of the theories of directed linear graphs, electrical networks and network programming, *J. Math. and Mech.* 15 (1966), 485–520.

[OhIsWa70] T. Ohtsuki, Y. Ishizaki, and H. Watanabe, Topological degrees of freedom and mixed analysis of electrical networks, *IEEE Trans. Circuit Theory* 17 (1970), 491–499.

[Pa94] K. R. Parthasarathy, *Basic Graph Theory*, Tata McGraw-Hill Publishing Company, New Delhi, India, 1994.

[Re61] M. B. Reed, The seg: a new class of subgraphs, *IEEE Trans. on Circuit Theory* CT-8 (1961), 17–22.

[RoRe78] P. Rosenstiehl and R. C. Read, On the principal edge tripartition of a graph, *Annals of Discrete Mathematics* 3 (1978), 195–226.

[SeRe61] S. Seshu and M. B. Reed, *Linear Graphs and Electrical Networks*, Addison Wesley, 1961.

[SwTh81] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms*, Wiley (Interscience), 1981.

[ThSw92] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*, Wiley (Interscience), 1992.

[Tu59] W. T. Tutte, Matroids and graphs, *Trans. of the Amer. Math. Soc.* 90 (1959), 527–552.

[Tu60] W. T. Tutte, An algorithm for determining whether a given binary matroid is graphic, *Proc. Amer. Math. Soc.* 11 (1960), 905–917.

[Tu64] W. T. Tutte, From matrices to graphs, *Canad. J. Math.* 56 (1964), 108–127.

[VaCh80] J. Vandewalle and L. O. Chua, The colored branching theorem and its applications in circuit theory, *IEEE Trans. Circuits and Systems* 27 (1980), 816–825.

[Ve31] O. Veblen, *Analysis Situs*, Amer. Math. Soc., 1931.

[We01] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2001.

[Wh33] H. Whitney, Planar graphs, *Fund. Math.* 21 (1933), 73–84.

[WiMa71] T. W. Williams and L. M. Maxwell, The decomposition of a graph and the introduction of a new class of subgraphs, *SIAM J. Appl. Math.* 20 (1971), 385–389.

[Wo70] D. H. Wolaver, Proof in graph theory of the no gain property of resistor networks, *IEEE Trans. Circuits and Systems* 17 (1970), 436–437.

6.5 SPECTRAL GRAPH THEORY

Michael Doob, University of Manitoba, Canada

6.5.1 Basic Matrix Properties
6.5.2 Walks and the Spectrum
6.5.3 Line Graphs, Root Systems, and Eigenvalue Bounds
6.5.4 Distance-Regular Graphs
6.5.5 Spectral Characterization
6.5.6 The Laplacian
References

Introduction

Spectral graph theory involves the investigation of the relationship of the usual (topological) properties of a graph with the (algebraic) spectral properties of various matrices associated with it. By far the most common matrix investigated has been the 0-1 adjacency matrix. The subject had its genesis with the paper by L. Collatz and U. Sinogowitz [CoSi57] in 1957. Since that time the subject has steadily grown and has shown surprising interrelationships with other mathematical areas.

Throughout this section, graphs are assumed to be simple.

6.5.1 Basic Matrix Properties

Many spectral properties of graphs follow from direct application or easy extensions of known results in matrix theory. An older but compact and useful reference is [MiMa64]. A more encyclopedic one is [Ga60].

DEFINITIONS

D1: The ***adjacency matrix*** of a (simple) graph G is a square matrix A (or A_G) with rows and columns corresponding to the vertices.

$$A_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{otherwise} \end{cases}$$

D2: The ***characteristic polynomial of a graph*** is the determinant $\det(xI - A)$ of its adjacency matrix.

D3: The ***eigenvalues of a graph*** are the roots to the characteristic polynomial.

D4: The ***algebraic multiplicity*** of an eigenvalue λ is the number of times it occurs as a root of the characteristic polynomial.

D5: The ***spectrum of a graph*** is the multiset of eigenvalues. For a graph with n vertices, there are n eigenvalues.

D6: The ***geometric multiplicity*** of an eigenvalue λ is the dimension of the *eigenspace* $\{x \mid Ax = \lambda x\}$.

FACTS

F1: The eigenvalues of a graph are real. This follows because the adjacency matrix is real and symmetric, which implies that it is *Hermitian*, i.e., each entry $a_{i,j}$ equals the complex conjugate of $a_{j,i}$. (The eigenvalues of any Hermitian matrix are real.)

F2: The geometric and algebraic multiplicities of each eigenvalue are equal.

F3: The eigenvalues have a corresponding set of eigenvectors that is orthonormal.

F4: The adjacency matrix A of a graph can be diagonalized, that is, there is a square matrix U (of eigenvectors) such that $UAU^T = UAU^{-1}$ is a diagonal matrix with the eigenvalues as diagonal entries.

F5: The *trace* of the adjacency matrix of a graph, that is, the sum of the eigenvalues of a graph, is 0, since the adjacency matrix is 0 on the diagonal.

F6: The spectrum of a graph is the union of the spectra of its connected components, since connected components of a graph are just blocks down the diagonal of the adjacency matrix.

F7: If r is the largest eigenvalue of a graph, then $|\lambda| \leq r$ for any eigenvalue λ of that graph. (Since the adjacency matrix has nonnegative entries and connectivity implies irreducibility, this follows from the well-known Perron-Frobenius theorem – see [MiMa64].)

F8: If a graph is connected, then the largest eigenvalue has multiplicity 1. It has an eigenvector with all entries positive. (This fact is another consequence of the Perron-Frobenius theorem.)

Since the adjacency matrix is symmetric, being imprimitive is equivalent to the graph being bipartite.

F9: Let r be the largest eigenvalue of a graph. Then a graph is bipartite if and only if $-r$ is also an eigenvalue.

F10: Whether or not a graph is bipartite can be determined by its spectrum. (This follows immediately from Fact 9.)

F11: A graph is bipartite if and only if the spectrum is symmetric around 0, that is, λ is an eigenvalue if and only if $-\lambda$ is an eigenvalue.

REMARK

R1: Because of Fact 6, for most results in this chapter the graphs under consideration may be assumed to be connected, with no loss of generality.

EXAMPLES

The spectrum of a graph is given as a set of eigenvalues with the multiplicities as exponents (and thus, the determinant is taken as the product of the set entries).

E1: The complete graph K_n: $\{(n-1)^1, -1^{n-1}\}$.

E2: The complete bipartite graph $K_{m,n}$: $\{\sqrt{mn}^1, 0^{m+n-2}, -\sqrt{mn}^1\}$.

E3: The path with n vertices: P_n: $\{2\cos(k\pi/n+1)^1, k=1,\ldots,n\}$.

E4: The circuit with n vertices C_n: $\{2\cos(2k\pi/n)^1, k=1,\ldots,n\}$.
Notice that these eigenvalues are not distinct. The eigenvalues 2 and, when n is even, -2 are simple and all others have multiplicity 2.

E5: A ***cocktail party graph*** $\mathrm{CP}(n)$ is a complete graph on $2n$ vertices with a 1-factor deleted. Spectrum: $\{(2n-2)^1, 0^n, -2^{n-1}\}$.

E6: The d-dimensional hypercube Q_d: $\{(d-2k)^{\binom{d}{k}}, k=0,\ldots,d\}$.

E7: A ***wheel*** W_n (with $n+1$ vertices) is the join of an n-cycle C_n and an additional vertex. Spectrum: $\{2\cos(2k\pi/n)^1, k=1,\ldots,n-1\} \cup \{1 \pm \sqrt{1+n}\}$.

E8: The platonic graphs

tetrahedral graph K_4: $\{3, -1^3\}$.

cube graph Q_3: $\{3^1, 1^3, -1^3, -3^1\}$.

octahedral graph $\mathcal{O}_3(\cong \mathrm{CP}(3))$: $\{4^1, 0^3, -2^2\}$.

dodecahedral graph: $\{3^1, \sqrt{5}^3, 1^5, 0^4, -2^4, -\sqrt{5}^3\}$.

icosahedral graph: $\{5^1, \sqrt{5}^3, -1^5, -\sqrt{5}^3\}$.

6.5.2 Walks and the Spectrum

Walks and the Coefficients of the Characteristic Polynomial

DEFINITIONS

D7: An ***elementary figure*** in a graph is a subgraph that is isomorphic to K_2 or to a cycle graph C_k.

D8: A ***basic figure*** is a vertex-disjoint union of elementary figures.

FACTS

F12: If A is the adjacency matrix of a graph, then ${A^k}_{i,j}$ is the number of walks of length k from the vertex v_i to v_j.

F13: If A is the adjacency matrix of a graph, then ${A^k}_{j,j}$ is the number of closed walks of length k from the vertex v_j to v_j.

F14: The trace of A^k is the total number of closed walks of length k in the graph.

F15: The trace of A^2 is twice the number of edges in the graph.

F16: The trace of A^3 is six times the number of triangles in the graph.

F17: [Sa64] If $a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$ is the characteristic polynomial of a graph, then $a_n = 1$ and $a_{n-1} = 0$. Also, $-a_{n-2}$ is the number of edges and $-a_{n-3}$ is twice the number of triangles in the graph. (This fact follows readily from an expansion of the determinant $\det(xI - A)$.)

F18: [Sa64] If $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ is the characteristic polynomial of a graph, then $a_k = \sum(-1)^{\text{comp}(B)} 2^{\text{circ}(B)}$, where the sum is over all basic figures with $n-k$ vertices, comp(B) is the number of connected components of B, and circ(B) is the number of circuits of B.

REMARK

R2: Fact 18 is sometimes called the *coefficients theorem*. Its interesting history is given in [CvDoSa95]. It extends the idea of Fact 17 to other coefficients of the characteristic polynomial. The coefficient of x^k will be determined by permutations with exactly k fixed points (vertices). For the other vertices, the permutations will have cycles of length 2 (corresponding to an edge in the graph) or cycles of length greater than two (corresponding to a circuit in the graph). Thus, in order to determine the coefficient of x^k we need to count all of the basic figures that have $n-k$ vertices. Furthermore, within each basic figure, a permutation corresponding to a circuit contributes 2 to the determinant (once clockwise, once counterclockwise) and an edge contributes 1 to the determinant.

Walks and the Minimal Polynomial

DEFINITIONS

D9: The ***minimal polynomial*** of a graph G is the monic polynomial $q(x)$ of smallest degree, such that $q(A_G) = 0$.

D10: The ***eigenvalues-diameter (lower) bound*** for the number of eigenvaues of a graph G is diam$(G) + 1$.

FACTS

F19: The minimal polynomial of a graph is $m(x) = \prod(x - \lambda_i)$ where the product is taken over all distinct eigenvalues.

F20: Given two vertices v_i and v_j at distance t in a graph with adjacency matrix A, we have $A^k_{i,j} = 0 \leq k < t$, and $A^t_{i,j} \neq 0$.

F21: If a graph has diameter d and has m distinct eigenvalues, then $m \geq d+1$. This substantiates the eigenvalues-diameter bound. It follows from Fact 20.

F22: The degree of the minimal polynomial is larger than the diameter of a graph.

F23: The complete graph is the only (connected) graph with exactly two distinct eigenvalues.

F24: The complete graph K_n is determined by its spectrum. (This follows from Fact 23, since the total number of eigenvalues — taking multiplicities into account — equals the number of vertices.)

EXAMPLES

E9: Note that in our previous examples, the graphs K_n, $K_{m,n}$, CP(n), P_n, C_n, and Q_n all attain the eigenvalues-diameter bound.

E10: The wheel W_n has approximately $n/2$ distinct eigenvalues and diameter 2.

OPEN PROBLEM

P1. Characterizing those graphs meeting the eigenvalues-diameter bound remains an open question. Of the 31 connected graphs with 5 or fewer vertices, there are 12 that meet the bound.

Regular Graphs

DEFINITION

D11: The ***Hoffman polynomial*** for a regular, connected graph of degree r is the polynomial

$$h(x) = n \prod \frac{(x - \lambda_i)}{(r - \lambda_i)}$$

the product being taken over all distinct eigenvalues not equal to r.

NOTATION: Let J denote a square matrix with every entry equal to 1.

FACTS

F25: The largest eigenvalue of a regular graph of degree r is r itself. A corresponding eigenvector is $(1, 1, \ldots, 1)^T$. (This follows from the Perron-Frobenius theorem.)

F26: Any eigenvector corresponding to an eigenvalue other than r has coordinates that sum to 0. (This is because eigenvectors from different eigenvalues are orthogonal.)

F27: The multiplicity of the eigenvalue r is the number of connected components. (Each connected component contributes 1 to the multiplicity of r.)

F28: A graph is regular if and only if A and J commute.

F29: The complement of a regular graph with n vertices has adjacency matrix equal to $J - A - I$. Hence the eigenvalues for the complement of a regular connected graph are $n - r - 1$ and $-\lambda_i - 1$, where λ_i runs over the eigenvalues of A not equal to r.

F30: [Ho63] For a regular connected graph with adjacency matrix A and Hoffman polynomial $h(x)$. Then $h(A) = J$.

6.5.3 Line Graphs, Root Systems, and Eigenvalue Bounds

An early problem in spectral graph theory was bounding the eigenvalues of a graph from below. One of the basic tools for bounding eigenvalues comes from matrix theory and is called the *interlacing theorem*. Other uses for the interlacing theorem are given in [Ha95].

DEFINITIONS

D12: A ***principal submatrix*** of an $n \times n$ square matrix is obtained by deleting the i^{th} row and the i^{th} column for some $1 \leq i \leq n$.

D13: The ***line graph*** of a graph G, denoted $L(G)$, has the edges of G as vertices with two vertices in $L(G)$ adjacent if, as edges of G, they have an endpoint in common.

D14: The ***vertex-edge incidence matrix*** of a graph G has rows corresponding to its vertices and columns corresponding to its edges. An entry is 1 if the vertex corresponding to the row is incident to the edge corresponding to the column, and is 0 otherwise. It is denoted by $K(G)$ or, simply, by K.

D15: Given a graph G with n vertices and nonnegative integers $a_1, \ldots, a_n$, the ***generalized line graph*** $L(G; a_1, \ldots, a_n)$ is formed as follows: first, take disjoint copies of the line graph $L(G)$ and cocktail party graphs $\mathrm{CP}(a_1), \ldots, \mathrm{CP}(a_n)$. In addition, if a vertex in $L(G)$ corresponds to the edge joining v_i to v_j in G, then join it to all vertices in $\mathrm{CP}(a_i)$ and $\mathrm{CP}(a_j)$.

NOTATIONS

NOTATION: $\lambda(G)$ denotes the smallest eigenvalue of a graph G.

NOTATION: $\Lambda(G)$ denotes the largest eigenvalue of a graph G.

FACTS ABOUT INTERLACING

F31: Let A be a real symmetric matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, having a principal submatrix with eigenvalues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_{n-1}$. Then

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \cdots \geq \lambda_{n-1} \geq \mu_{n-1} \geq \lambda_n$$

F32: If H is an induced subgraph of G, if $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$ are the eigenvalues of H, and if $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ are the eigenvalues of G, then

$$\lambda_i \geq \mu_i \geq \lambda_{i+n-m} \quad \text{for } i = 1, \ldots, m$$

FACTS ABOUT THE SMALLEST EIGENVALUE $\lambda(G)$

F33: If H is an induced subgraph of G, then $\lambda(H) \leq \lambda(G)$.

F34: The least eigenvalue of a connected graph is always nonpositive. It equals zero if and only if the graph is K_1.

F35: No graph has a least eigenvalue between 0 and -1.

F36: The only connected graphs with least eigenvalue -1 are the complete graphs with two or more vertices.

F37: There are no graphs with least eigenvalue between -1 and $-\sqrt{2}$.

F38: $K_{1,2}$ is the only connected graph whose least eigenvalue equals $-\sqrt{2}$.

F39: There are infinitely many connected graphs with their least eigenvalues between $-\sqrt{2}$ and -2. (This follows from Example 3.)

Line Graphs and Generalized Line Graphs

FACTS ABOUT THE LINE GRAPH $L(G)$

F40: If K is the vertex-edge incidence matrix of a graph G, then $KK^T = 2I + A(L(G))$.

F41: If K is the vertex-edge incidence matrix of a graph G, then the matrix KK^T is positive semidefinite, and hence has nonnegative eigenvalues.

F42: [Ho75] For any graph G, $\lambda(L(G)) \geq -2$.

F43: [Do70] A graph G satisfies $\lambda(L(G)) > -2$ if and only if G is a tree or G has exactly one circuit, that circuit being odd.

F44: There are infinitely many graphs G with $-2 \leq \lambda(G) < -\sqrt{2}$.

FACTS ABOUT THE GENERALIZED LINE GRAPH $\lambda(L(G; a_1, \ldots, a_n)$

The results for generalized line graphs are similar to those for line graphs. Form the matrix K' by appending columns and rows to the vertex-edge incidence matrix. For each a_i, append a_i pairs of new columns and each of which has two nonzero entries. For each pair there is a 1 in the row corresponding to v_i and a new row is added with one 1 and one -1 in the new columns. All other entries are 0. Use $K'K'^T$ as before.

F45: [CDS81] For any graph G with n vertices $\lambda(L(G; a_1, \ldots, a_n) \geq -2$.

F46: [CDS81] A graph G satisfies the lower bound $\lambda(L(G; a_1, \ldots, a_n) > -2$ if and only if $G = L(T; 1, 0, \ldots, 0)$ where T is a tree or $G = L(H)$ where H is a tree and H has exactly one cycle, that cycle being odd.

F47: [Ho75] If G is a regular connected graph of degree r with n vertices and eigenvalues $r = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_n$ then the eigenvalues of $L(G)$ are $\lambda_i + r - 2$, $i = 1, \ldots, n$ plus -2 of (additional) multiplicity $n(r-2)/2$.

EXAMPLES

E11: The line graph of a complete graph is called a ***triangular graph***. From the spectrum of K_n (Example 1) we see that the spectrum of the triangular graph $L(K_n)$ is $\{2n-4^1, n-4^{n-1}, -2^{n(n-3)/2}\}$.

E12: The line graph of a regular complete bipartite graph is called a ***lattice graph***. From the spectrum of $K_{n,n}$ (Example 2), it follows that the spectrum of the lattice graph $L(K_{n,n})$ is $\{2n-2^1, n-2^{2n-2}, -2^{(n-1)^2}\}$.

Root Systems

The converse of Fact 46 is almost true. The exact statement involves the *root systems* used in the classification of semisimple Lie algebras (found in [Ca89], for example). ***Root systems*** are sets of vectors used to form the columns of the matrices K' as above.

FACTS ABOUT ROOT SYSTEMS

The possible real root systems are denoted A_n, D_n, E_6, E_7, and E_8. These root systems have $n(n+1)/2$, $n(n-1)$, 36, 63, and 120 vectors, respectively. They also satisfy $A_{n-1} \subseteq D_n$ and $E_6 \subseteq E_7 \subseteq E_8$. One of the most beautiful results in spectral graph theory relates root systems with eigenvalues of graphs.

F48: [GCSS76] $\lambda(H) \geq -2$ if and only if there is a matrix K' whose columns are taken from the root systems A_n, D_n, E_6, E_7, or E_8 such that $A = 2I + K'K'^T$.

F49: [GCSS76] $H = L(G)$ where G is bipartite if and only if K' can be formed from vectors in the root system A_n.

F50: [GCSS76] $H = L(G; a_1, \ldots, a_n)$ if and only if K' can be formed from vectors in the root system D_n.

F51: [GCSS76] With only a finite number of exceptions, $\lambda(H) \geq -2$ implies that H is a generalized line graph.

The exceptional graphs from the last fact are those constructed from (the finite number of) vectors in E_6, E_7, and E_8 by using them as columns of a matrix K' as described in Fact 48. Using a variety of techniques (both computer-assisted and otherwise) much is known about these exceptional graphs.

F52: If $\lambda(H) > -2$, then either H can be formed by vectors in the root system D_n, or H is one of 20 graphs with six vertices, 110 graphs with seven vertices, or 443 graphs with eight vertices.

A type of characterization of graphs with $\lambda(G) > -1 - \sqrt{2}$ by a different generalization of line graphs has been given in [WoNu95].

EXAMPLE

E13: Let $\{e_1, \ldots, e_n\}$ be the canonical basis for $\mathbb{R}^n$. Consider the following set of vectors: $\{e_i - e_{i+1}, i = 1, \ldots, n-1\} \cup \{e_1 + e_2, e_{n-1} + e_n\}$. In fact, these vectors are in the root system D_n. Use these vectors as columns for K' and define a graph, also denoted by D_n, from the matrix equation $A = 2I + KK'^T$. The graph then appears as in Figure 6.5.1.

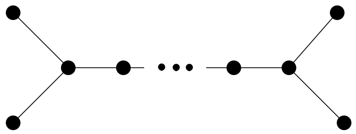


Figure 6.5.1 **The graph D_n.**

Eigenvalue Bounds

FACTS ABOUT THE LARGEST EIGENVALUE $\Lambda(G)$

Many of the above ideas can also be used to obtain upper bounds on the eigenvalues of graphs.

F53: If H is an induced subgraph of G, then $\Lambda(H) \leq \Lambda(G)$. If H is a proper subgraph of G, then $\Lambda(H) < \Lambda(G)$.

F54: If $\bar{r}$ is the average degree and $r_{\max}$ is the maximum degree of a graph G, then $\bar{r} \leq \Lambda(G) \leq r_{\max}$. Equality is attained if and only if the graph is regular.

F55: [CvDoGu82] If $\Lambda(G) < 2$, then G is P_n, $T(1,1,n)$, $T(1,2,4)$, $T(1,2,3)$, or $T(1,2,2)$. The graph $T(i,j,k)$ is formed by taking three paths $u_0, \ldots, u_i$, $v_0, \ldots, v_j$, and $w_0, \ldots, w_k$ and identifying the vertices u_o, v_0, and w_0. It clearly has one vertex of degree three, three vertices of degree one, and all other vertices of degree two.

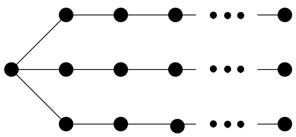


Figure 6.5.2 **The graph $T(i,j,k)$.**

F56: [CvDoGu82] If $\Lambda(G) = 2$ then G is C_n, $K_{1,4}$, $T(2,2,2)$, $T(3,3,1)$, $T(5,2,1)$ or D_n.

Similar results are available for graphs with $\Lambda(G) < (2+\sqrt{5})^{1/2}$ (see [CvDoGu82]).

FACTS ABOUT THE SECOND LARGEST EIGENVALUE

Bounds on the second largest eigenvalue of a graph, $\lambda_2(G)$, have also been of considerable interest.

F57: [Sm70] $\lambda_2(G) \leq 0$ if and only if G is a complete multipartite graph.

F58: [Li94] $\lambda_2(G)$ is dense in the interval $\left((2+\sqrt{5})^{1/2}, \infty\right)$ but nowhere dense in $(-\infty, \sqrt{2}-1)$.

Graphs with $\lambda_2(G) \leq \sqrt{2}-1$ and $\lambda_2(G) \leq (\sqrt{5}-1)/2$ are described in [CvSi95], [Li94], and [Si95].

FURTHER READING

A more general discussion on different bounds on the eigenvalues of graphs can be found in [PeRa01].

6.5.4 Distance-Regular Graphs

Eigenvalues are crucial for understanding the properties of distance-regular graphs.

DEFINITIONS

D16: A graph of diameter d is ***distance-regular*** with parameters

$$\{p_{i,j}^k \mid 0 \leq i,j,k \leq d\}$$

if for each triple (i,j,k) and for any pair of vertices such that distance between them is k, the number of vertices at distance i from the first and distance j from the second is $p_{i,j}^k$. Each of these numbers $p_{i,j}^k$ is independent of the particular choice of vertices.

D17: In the ***Hamming graph*** $H(d,n)$, the vertices are all d-tuples $(x_1, \ldots, x_d)$ with $1 \leq x_i \leq n$. Two vertices are joined if, as d-tuples, they agree in all but one coordinate. The distance between two vertices is then the number of coordinates in which, as d-tuples, they differ.

D18: In the ***Johnson graph*** $J(d,n)$, the vertices are the $\binom{n}{d}$ subsets of $\{1, \ldots, n\}$. Two vertices are joined if, as subsets, their intersection has cardinality $d-1$. The distance bewteen two vertices is k if, as subsets, they have an intersection of cardinality $d-k$.

EXAMPLES

E14: $H(d,2)$ is isomorphic to the hypercube graph Q_d. It is distance regular with diameter d.

E15: $J(2,n)$ is isomorphic to the triangular graph $L(K_n)$. It is distance regular with diameter 2.

Distance-Regular Graphs and the Hoffman Polynomial

A distance-regular graph is regular (from $p^0_{1,1}$) and connected, and so it has a Hoffman polynomial.

DEFINITIONS

D19: The l^{th}***-order adjacency matrix*** is defined with $A_0 = I$ as the identity matrix, $A_1 = A$ as the usual adjacency matrix, and A_l as the matrix with 1 in the (i, j) position if the corresponding vertices are at distance l from each other and 0 otherwise.

D20: The l^{th}***-order parameter matrix*** is the matrix P_k with the distance-regularity parameter $p^i_{j,k}$ in the (i, j) entry.

FACTS ABOUT THE MATRIX A_l

F59: For a distance-regular graph with diameter d, we have $A_iA_j = \sum_{k=0}^{d} p^k_{i,j}A_k$. Also, $\sum_{k=0}^{d} A_k = J$, the all-one matrix.

F60: The algebra generated by $\{A_0, \ldots, A_d\}$ is of dimension $d + 1$ (since matrices A_i and A_j commute and the A_i are linearly independent). All the matrices A^k are in this algebra for $k = 0, 1, \ldots, d$.

F61: The number of distinct eigenvalues of the adjacency matrix A of a distance-regular graph of diameter d is $d + 1$.

F62: The Hoffman polynomial of any distance-regular graph of diameter d is a polynomial of degree d.

F63: Any distance-regular graph meets the eigenvalues-diameter bound.

FACTS ABOUT THE PARAMETERS $p^k_{i,j}$

F64: $P_iP_j = \sum_{k=0}^{d} p^k_{i,j}P_k$, and so the commutative algebra generated by $\{P_0, \ldots, P_d\}$ is isomorphic to the one generated by $\{A_0, \ldots, A_d\}$.

F65: The minimal polynomial for A_1 and P_1 is the same, and so A_1 and P_1 have the same distinct eigenvalues.

F66: The eigenvalues of P_1 are *simple*. That is, they occur with multiplicity one.

F67: The parameters of a distance-regular graph determine the spectrum.

Strongly Regular Graphs

DEFINITION

D21: A ***strongly regular graph*** is a distance-regular graph of diameter 2. The parameters are (n, r, λ, μ) where n is the number of vertices, r is the degree, $\lambda = p^1_{1,1}$ and $\mu = p^2_{1,1}$. To avoid trivialities, K_n and its complement are not strongly regular.

EXAMPLES

E16: Triangular graphs: $L(K_n)$ has parameters $(n(n - 1)/2, 2n - 4, n - 2, 4)$.

E17: Lattice graphs: $L(K_{n,n})$ has parameters $(n^2, 2n - 2, n - 2, 2)$.

E18: Paley graphs: $P(p^n)$ has as vertices the elements of the finite field $\mathrm{GF}(p^n)$ with two vertices adjacent if, as field elements, their difference is a quadratic residue (for this relation to be symmetric p^n must be 1 mod 4). The Paley graph has parameters $(p^n, (p^n-1)/2, (p^n-5)/4, (p^n-1)/4)$.

FACTS

The parameters of a strongly regular graph are not independent. Pick a vertex and count the number of paths of length two starting at that vertex and ending at a different one. There are $n-r-1$ vertices at distance two from our given vertex, and each one contributes μ such paths. Also, for each of the r vertices adjacent to the given vertex, there are $r-1-\lambda$ choices for a second edge to get a desired path.

F68: For a strongly regular graph with parameters (n,r,λ,μ), we have $\mu(n-r-1) = r(r-1-\lambda)$.

F69: Since there are only three types of entries in A, A^2, and I, for a strongly regular graph (corresponding to equal, adjacent, and nonadjacent vertices) it's easy to recognize the Hoffman polynomial and hence the eigenvalues for a strongly regular graph. In particular, for a strongly regular graph with parameters (n,r,λ,μ), we have

$$A^2 + (\mu-\lambda)A + (\mu-r)I = \mu J.$$

F70: The eigenvalues of a strongly regular graph with parameters (n,r,λ,μ) are r and the two roots of the polynomial $x^2+(\mu-\lambda)x+(\mu-r)$.

F71: A regular connected graph is strongly regular if and only if it has three distinct eigenvalues.

It is also easy to compute the multiplicities of the eigenvalues, since r is a simple eigenvalue, the sum of the multiplicities is n, and the trace of A is 0.

F72: For a strongly regular graph with parameters (n,r,λ,μ), the eigenvalues are $\lambda_1 = r$, $\lambda_2 = (\lambda-\mu)/2+\Delta^{1/2}$ and $\lambda_3 = (\lambda-\mu)/2-\Delta^{1/2}$, where $\Delta = \mu^2-2\mu\lambda+\lambda^2-4\mu-4\lambda$. The respective multiplicities are 1, m_2, and m_3 where $m_2+m_3 = n-1$ and $m_2\lambda_2+m_3\lambda_3 = -r$. Since $\lambda_2 \neq \lambda_3$, the solution for m_2 and m_3 is unique.

F73: If Δ in Fact 72 is not a square, then $m_2 = m_3$. Such a graph is called a *conference graph.*

F74: One test to see if a potential set of parameters (n,r,λ,μ) is actually attained by a graph is to see if the multiplicities m_2 and m_3 are integers.

EXAMPLE

E19: For the Paley graph, $x^2+(\mu-\lambda)x+(\mu-r) = x^2+x-(p^n+1)/4$, $\{\lambda_2,\lambda_3\} = \{\frac{1}{2}(-1\pm p^{n/2})\}$, and $m_2 = m_3 = (p^n-1)/4$.

FURTHER READING

Further details and results concerning strongly regular graphs can be found in the excellent reference [GoRo01]. The encyclopedic reference for distance-regular graphs is [BrCoNu89].

6.5.5 Spectral Characterization

One of the earliest and continuing questions in spectral graph theory asks the following: when is a graph characterized by its spectrum? Finding pairs of nonisomorphic graphs with the same spectrum can pinpoint properties of a graph that cannot be determined spectrally.

EXAMPLES

E20: Figure 6.5.3 shows the two smallest graphs with the same spectrum, which is $\{2^1, 0^3, -2^1\}$. This example implies that the number of quadrilaterals (unlike the number of triangles) can not be determined from the spectrum. Similarly, neither the degree sequence nor the connectivity can be determined by the spectrum.

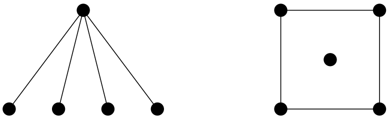


Figure 6.5.3 **The smallest pair of cospectral graphs.**

E21: Figure 6.5.4 shows the two smallest connected graphs with the same spectrum. The characteristic polynomial of both graphs is $(x-1)(x^3-x^2-5x+1)(x+1)^2$.

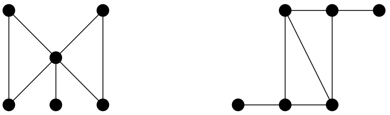


Figure 6.5.4 **The smallest pair of connected cospectral graphs.**

E22: Figure 6.5.5 shows two cospectral trees with the smallest possible number of vertices. The spectrum is $\pm\frac{1}{2} \pm \frac{\sqrt{13}}{2}$ (all four simple) and 0 with multiplicity 4. The characteristic polynomial in this case is $x^4(x^2+x-3)(x^2-x-3)$.

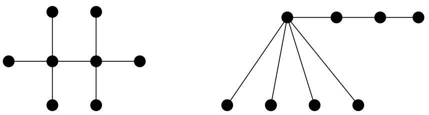


Figure 6.5.5 **Cospectral trees with the minimum number of vertices.**

E23: Figure 6.5.6 shows a pair of strongly regular cospectral graphs with 16 vertices and spectrum $\{6^1, 2^6, -2^9\}$. Some interpretation is necessary. The graph on the left is actually drawn on the torus, that is, the vertices on the outside edges in the same row or column are identified. In the graph on the right, any pair of vertices in the same row or column are joined. The graph on the right is actually $L(K_{4,4})$ and hence is strongly regular with parameters $(16, 6, 2, 2)$. Being cospectral, the one on the left (called the *Shrikhande graph*) must be strongly regular with the same parameters.

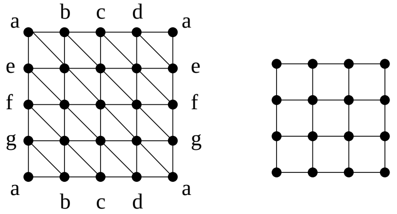


Figure 6.5.6 **A pair of strongly regular cospectral graphs.**

Eigenvalues and Graph Operations

One method for constructing cospectral graphs is by using various graph operations.

DEFINITIONS

D22: The ***cartesian product*** of two graphs G_1 and G_2, denoted $G_1 \otimes G_2$, has as vertex set all possible pairs (v_1, v_2) where v_1 is a vertex of G_1 and v_2 is a vertex of G_2. Two vertices are joined if, as ordered pairs, they are identical in one coordinate and adjacent in the other.

D23: The ***coalescence*** of two (disjoint) graphs G_1 and G_2 with distinguished vertices v_1 and v_2, denoted $G_1 \cdot G_2$, is formed by identifying v_1 and v_2, that is, the vertices v_1 and v_2 are replaced by a single vertex v adjacent to the same vertices in G_1 as v_1 and the same vertices in G_2 as v_2.

FACTS

F75: The straightforward extension of the cartesian product to an iterated product of more than two graphs is associative.

F76: The eigenvalues of the product graph $G_1 \otimes G_2$ are precisely all possible sums $\lambda_1 + \lambda_2$, where λ_1 is an eigenvalue of G_1 and λ_2 is an eigenvalue of G_2.

F77: Let G_1 and G_2 be a pair of cospectral nonisomorphic graphs. For $t = 0, 1, \ldots, m$, we define H_t by taking the Cartesian product $G_1 \otimes \cdots \otimes G_1 \otimes G_2 \otimes \cdots \otimes G_2$ using t copies of G_1 and $m - t$ copies of G_2. Then the graphs H_t are pairwise cospectral and nonisomorphic.

F78: [Sc73] Let $P_H(x)$ be the characteristic polynomial of the graph H. Then

$$P_{G_1 \cdot G_2}(x) = P_{G_1}(x)P_{G_2 - v_2}(x) + P_{G_1 - v_1}(x)P_{G_2}(x) - xP_{G_1 - v_1}P_{G_2 - v_2}$$

F79: [Sc73] As the number of vertices gets large, the probability that a tree has a cospectral mate goes to 1.

EXAMPLES

The first example of this type of construction was given by A. J. Hoffman.

E24: (A. J. Hoffman in [Mo72]) Let G_1 and G_2 be a pair of regular cospectral nonisomorphic graphs. Define H_t by taking t copies of G_1 and $m-t$ copies of G_2 and taking the complement, for $t = 0, 1, \ldots, m$. Then the H_t are all cospectral, connected, regular and nonisomorphic.

E25: The Hamming graph $H(d, n)$ is simply $K_n \otimes \cdots \otimes K_n$, where the number of factors is d.

REMARK

R3: The cartesian product has been generalized to the NEPS graph by D. Cvetković [CvDoSa95].

6.5.6 The Laplacian

The Laplacian is an alternative to the adjacency matrix for describing the adjacent vertices of a graph. It has many interesting properties, and, although not as much information is known about the spectral properties of the Laplacian of a graph, more recent work has indicated that there is much more to be found.

DEFINITIONS

D24: The ***Laplacian*** of a graph is a square matrix whose rows and columns correspond to the vertices of a graph. A diagonal entry is the degree of the corresponding vertex; an off-diagonal entry is -1 if the corresponding vertices are adjacent and 0 otherwise. In other words, $L = D - A$, where D is the diagonal matrix of degrees of the vertices and A is the usual adjacency matrix.

If the graph is regular, then $D = rI$, and the eigenvalues of A and L are obtainable from each other. Thus for regular graphs the study of the adjacency matrix and the Laplacian are identical.

D25: [Fi73] The ***algebraic connectivity*** of a connected graph whose Laplacian L has eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ is defined to be λ_2.

FACTS

The oldest result about the Laplacian concerns the number of spanning trees of a graph. Let $\tau = \tau(G)$ be the number of spanning trees of a graph. Let $L_{i,j}$ be the matrix obtained by deleting the i-th row and j-th column from L. Also, let $\operatorname{adj} L$ be the adjoint of L.

F80: $\tau(G) = (-1)^{i+j} \det(L_{i,j})$.

F81: $\operatorname{adj}(L) = \tau J$.

F82: The multiplicity of 0 as an eigenvalue of L is the number of connected components in the graph.

F83: $\prod_{i=2}^{n} \lambda_i = n\tau(G)$.

F84: Since $\lambda_2 = \lambda_3 = \cdots = \lambda_n = n$ for K_n, $\tau(K_n) = n^{n-2}$.

F85: The algebraic connectivity is positive if and only if the graph is connected.

F86: $\lambda_2(G_1 \otimes G_2) = \min\{\lambda_2(G_1), \lambda_2(G_2)\}$.

F87: [Ne00] If a graph G has diameter d, then $\lambda_2(G) \geq \frac{1}{nd}$.

EXAMPLES

E26: $\lambda_2(P_n) = 2(1 - \cos(\pi/n))$.

E27: $\lambda_2(C_n) = 2(1 - \cos(2\pi/n))$.

E28: $\lambda_2(Q_n) = 2$.

E29: $\lambda_2(K_n) = n$.

E30: $\lambda_2(K_{m,n}) = \min\{m, n\}$.

FURTHER READING

A good introduction to further properties of the Laplacian is given by B. Mohar [Mo92]. Another excellent synopsis is by M. Newman [Ne00].

References

[BrCoNe89] A. E. Brouwer, A. M. Cohen, and A. Neumaier, *Distance-Regular Graphs*, Springer-Verlag, 1989.

[Ca89] R. W. Carter, *Simple Groups of Lie Type*, John Wiley & Sons, 1989.

[CoSi57] L. Collatz and U. Sinogowitz, Spektren endlicher Grafen, *Ahb. Math. Sem. Univ. Hamburg* 21 (1957), 27–56.

[CvDoGu82] D. Cvetković, M. Doob, and I. Gutman, On Graphs Whose Spectral Radius Does Not Exceed $(2+\sqrt{5})^{1/2}$, *Ars Combiatorica* **14** (1982), 225–239.

[CDS81] D. Cvetković, M. Doob, S. Simić, Generalized line graphs, *J. Graph Theory* **5** (1981), 385–399.

[CvDoSa95] D. Cvetković, M. Doob, and H. Sachs, *Spectra of Graphs*, Johann Ambrosius Barth, 1995.

[CvSi95] D. Cvetković and S. Simić, On graphs whose second largest eigenvalue does not exceed $(\sqrt{5}-1)/2$, *Discrete Math.* **139** (1995), 213–227.

[Da95] E. R. van Dam, Regular graphs with four eigenvalues, *Lin. Alg. and its Appl.* 226-228 (1995), 139–162.

[DaHa03] E. R. van Dam and W. H. Haemers, Which graphs are determined by their spectrum?, *Lin. Alg. and Its Appl.*, to appear.

[Do70] M. Doob, A geometric interpretation of the least eigenvalue of a line graph, *Proc. Second Conference on Comb. Math. and Appl., Chapel Hill, NC* (1970), 126–135.

[Fi73] M. Fiedler, Algebraic connectivity of graphs, *Czechsolvak. Math. J.* **23** (1973), 298–305.

[Ga60] F. R. Gantmacher, *The Theory of Matrices,* Vols. I, II, Chelsea, 1960.

[GCSS76] J. M. Goethals, P. Cameron, J. Seidel, E. Shult, Line graphs, roots systems and elliptic geometry, *J. Algebra* **43** (1976), 305–327.

[Go93] C. Godsil, *Algebraic Combinatorics*, Chapman and Hall, 1993.

[GoRo01] C. Godsil and G. Royle, *Algebraic graph theory*, Springer-Verlag, 2001.

[Ha95] W. Haemers, Interlacing eigenvalues and graphs, *Lin. Alg. and its Appl.* **226-228** (1995), 593–616.

[Ho63] A. J. Hoffman, On the polynomial of a graph, *Amer. Math. Monthly* **70** (1963), 30–36.

[Ho75] A. J. Hoffman, Eigenvalues of graphs, pp. 225–245 in *Studies in Graph Theory, II*, D. R. Fulkerson, Ed., Mathematical Assoc. of America (1970).

[Li94] J. Li, *Subdominant eigenvalues of graphs*, Ph. D. Thesis, University of Manitoba, 1994.

[MiMa64] H. Minc and M. Marcus, *A Survey of Matrix Theory and Matrix Inequalities*, Prindle, Weber & Schmidt, 1964.

[Mo92] B. Mohar, Laplace eigenvalues of graphs—a survey, *Discrete Math.* **109**(1992), 171–183.

[Mo72] A. Mowshowitz, The characteristic polynomial of a graph, *J. Comb. Theory (B)* (1972), 177–193.

[Ne00] M. Newman, *The Laplacian spectrum of graphs,* Masters Thesis, University of Manitoba, 2000.

[PeRa01] M. Petrović and Z. Radosavljević, *Spectrally constrained graphs*, Faculty of Science, University of Kragujevac, 2001.

[Sa64] H. Sachs, Beziehungen zwischen den in einem Graphen enthaltenen Kreisen und seinem charakteristischen Polynomialen, *Publ. Math. Debrecen* **11** (1964), 119–134.

[Sc73] A. J. Schwenk, *Almost all trees are cospectral*, in New Directions in the Theory of Graphs (Ed: F. Harary), Academic Press, 1973, 275–307.

[Si95] S. Simić, Some notes on graphs whose second largest eigenvalue is less than $(\sqrt{5}-1)/2$, *Linear and Multilinear Algebra* **39** (1995), 59–71.

[Sm70] J. H. Smith, Some properties of the spectrum of a graph, pp. 403–406 in *Combinatorial Structures and their Applications*, Eds: R. Guy, H. Hanani, N. Sauer, and J. Schönheim, Gordon and Breach (1970).

[WoNu95] On graphs whose smallest eigenvalue is at least $-1-\sqrt{2}$, *Lin. Alg. and its Appl.* 226–228 (1995), 577–591.

6.6 MATROIDAL METHODS IN GRAPH THEORY

James Oxley, Louisiana State University

6.6.1 Matroids: Basic Definitions and Examples
6.6.2 Alternative Axiom Systems
6.6.3 The Greedy Algorithm
6.6.4 Duality
6.6.5 Matroid Union and Its Consequences
6.6.6 Fundamental Operations
6.6.7 2- and 3-Connectedness for Graphs and Matroids
6.6.8 Graphs and Totally Unimodular Matrices
6.6.9 Excluded-Minor Characterizations
6.6.10 Wheels, Whirls, and the Splitter Theorem
6.6.11 Removable Circuits
6.6.12 Minimally k-connected Graphs and Matroids
6.6.13 Conclusion
References

Introduction

Every graph gives rise to a matroid, so every theorem for matroids has an immediate consequence for graphs, although many of these are easy to derive directly. On the other hand, numerous results for graphs have analogs or generalizations to matroids. This link between graph theory and matroid theory is so close that the famous graph theorist W. T. Tutte (1917–2002) wrote [Tu79]: “If a theorem about graphs can be expressed in terms of edges and circuits only it probably exemplifies a more general theorem about matroids.” This section provides an overview of the rich interaction between graph theory and matroid theory.

6.6.1 Matroids: Basic Definitions and Examples

The edge-sets of cycles in a graph and the minimal linearly dependent sets of columns in a matrix share many similar properties. Hassler Whitney (1907–1989) aimed to capture these similarities when he defined matroids in 1935 [Wh35].

DEFINITIONS

D1: A ***matroid*** M is a pair comprising a finite set $E(M)$ (the ***ground set*** of M) and a collection $\mathcal{C}(M)$ of nonempty incomparable subsets of E called ***circuits***, such that if C_1 and C_2 are distinct members of $\mathcal{C}(M)$ and $e \in C_1 \cap C_2$, then there is a member C_3 of $\mathcal{C}(M)$ such that $C_3 \subseteq (C_1 \cup C_2) - \{e\}$.

NOTATION: Frequently, $E(M)$ and $\mathcal{C}(M)$ are abbreviated to E and $\mathcal{C}$.

D2: A subset of E is ***dependent*** if it contains a member of $\mathcal{C}$, and it is ***independent*** otherwise.

D3: A ***basis*** (or ***base***) is a maximal independent set.

D4: The matroid M_1 is ***isomorphic*** to the matroid M_2, written $M_1 \cong M_2$, if there is a 1-1 function ϕ from $E(M_1)$ onto $E(M_2)$ such that C is a circuit of M_1 if and only if $\phi(C)$ is a circuit of M_2.

NOTATION: The collections of independent sets and bases of M are denoted by $\mathcal{I}(M)$ and $\mathcal{B}(M)$, respectively.

REMARK

R1: It follows easily from the definition of a matroid that all bases are of the same cardinality.

EXAMPLES

E1: Three different classes of examples of matroids are given in Table 6.6.1.

Table 6.6.1 Examples of matroids.

MATROID M	GROUND SET $E(M)$	CIRCUITS $\mathcal{C}(M)$	INDEPENDENT SETS, $\mathcal{I}(M)$	BASES $\mathcal{B}(M)$
$M(G)$, **cycle matroid** of graph G	$E(G)$, edge-set of G	edge-sets of cycles	$\{I \subseteq E(G)$: I contains no cycle$\}$	For conn G: edge-sets of spanning trees
$M[A]$, **vector matroid** of matrix A over field F	column labels of A	minimal linearly dependent multisets of columns	$\{I \subseteq E : I$ labels a linearly independent multiset of columns$\}$	maximal linearly independent sets of columns
Uniform matroid, $U_{m,n}$ $(0 \leq m \leq n)$	$\{1, 2, \ldots, n\}$	$\{C \subseteq E :$ $\lvert C\rvert = m+1\}$	$\{I \subseteq E :$ $\lvert I\rvert \leq m\}$	$\{B \subseteq E :$ $\lvert B\rvert = m\}$

E2: Let M be the matroid with $E(M) = \{1, 2, \ldots, 6\}$ and $\mathcal{C}(M) = \{\{1\}, \{5, 6\}, \{3, 4, 5\}, \{3, 4, 6\}\}$. Then $M = M(G_1) = M(G_2)$, where G_1 and G_2 are the graphs shown in Figure 6.6.1. Also $\mathcal{B}(M) = \{\{2, 3, 4\}, \{2, 3, 5\}, \{2, 3, 6\}, \{2, 4, 5\}, \{2, 4, 6\}\}$, and $M = M[A]$, where A is the following matrix over $\mathbf{R}$.

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$
$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

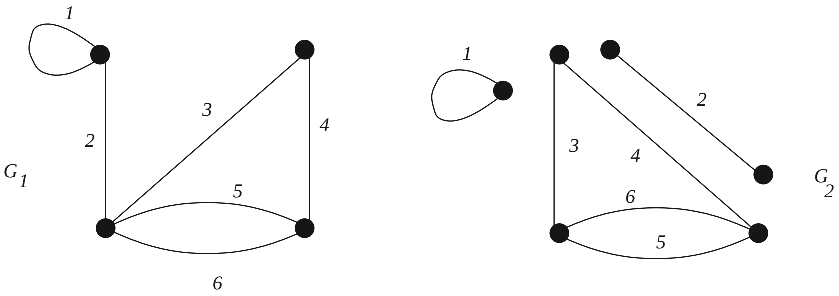


Figure 6.6.1 **Graphs G_1 and G_2 yield the same matroid M.**

E3: In Table 6.6.2, several classes of matroids are defined. A matroid M is in the specified class if it satisfies the indicated condition.

Table 6.6.2 Some classes of matroids.

Class	Condition
graphic	$M \cong M(G)$ for some graph G
representable over **F**	$M \cong M[A]$ for some matrix A over the field **F**
binary	representable over $GF(2)$, the 2-element field
ternary	representable over $GF(3)$
regular	representable over all fields

FACTS

In each of the following, M is a matroid.

F1: The unique smallest non-graphic matroid is $U_{2,4}$.

F2: If M is a graphic matroid, then $M \cong M(G)$ for some connected graph G.

F3: (Whitney's 2-Isomorphism Theorem [Wh33]) Two graphs have isomorphic cycle matroids if and only if one can be obtained from the other by a sequence of the following operations: (i) choose one vertex from each of two components and identify the chosen vertices; (ii) the reverse of (i); (iii) in a graph that can be obtained from the disjoint union of two graphs G_1 and G_2 by identifying vertices u_1 and v_1 of G_1 with vertices u_2 and v_2 of G_2, one can ***twist*** the graph by identifying, instead, u_1 with v_2 and u_2 with v_1.

The following is an immediate consequence of the last fact.

F4: A 3-connected loopless graph is uniquely determined by its cycle matroid.

F5: If M is a graphic matroid, then M is regular.

F6: A matroid M is regular if and only if M can be represented over the real numbers by a ***totally unimodular matrix***, a matrix for which all subdeterminants are in $\{0, 1, -1\}$.

6.6.2 Alternative Axiom Systems

Matroids can be characterized by numerous different axiom systems. Two examples of these systems follow. Others may be found, for example, in [Ox92]. Throughout, E is a finite set and 2^E is its set of subsets.

Independent set axioms. A subset $\mathcal{I}$ of 2^E is the set of independent sets of a matroid on E if and only if

(I1) the empty set is in $\mathcal{I}$;

(I2) every subset of a member of $\mathcal{I}$ is in $\mathcal{I}$ ($\mathcal{I}$ is *hereditary*); and

(I3) if X and Y are in $\mathcal{I}$ and Y has more elements than X, then there is an element e of $Y - X$ such that $X \cup \{e\}$ is in $\mathcal{I}$.

Basis axioms. A subset $\mathcal{B}$ of 2^E is the set of bases of a matroid on E if and only if

(B1) $\mathcal{B}$ is nonempty; and

(B2) if B_1 and B_2 are in $\mathcal{B}$ and $x \in B_1 - B_2$, then there is an element y of $B_2 - B_1$ such that $(B_1 - \{x\}) \cup \{y\} \in \mathcal{B}$.

DEFINITIONS

In all of the following, M is a matroid with ground set E.

D5: If $A \subseteq E$, then all maximal independent subsets of A have the same cardinality, which is called the ***rank of the subset*** A, and is denoted $r(A)$ (or $r_M(A)$).

D6: The ***rank*** $r(M)$ ***of the matroid*** M is the rank $r(E)$ of its ground set.

D7: A ***spanning set*** of M is a subset of E of rank $r(M)$.

D8: A ***hyperplane*** of M is a maximal nonspanning set.

D9: The ***closure*** $\text{cl}(X)$ of X is $\{x \in E : r(X \cup \{x\}) = r(X)\}$.

D10: A set Y is a ***flat*** (or ***closed set***) if $\text{cl}(Y) = Y$.

D11: A ***loop*** of M is an element e such that $\{e\}$ is a circuit.

D12: If $\{f, g\}$ is a circuit, then f and g are called ***parallel elements***.

D13: A ***simple matroid*** (or ***combinatorial geometry***) is a matroid that has no loops and no parallel elements.

FACT

F7: If X is a set of edges of a graph G and $G[X]$ is the subgraph of G induced by X, then $r_{M(G)}(X) = |V(G[X])| - k(G[X])$ where $k(G[X])$ is the number of components of $G[X]$.

EXAMPLE

E4: For the graphs G_1 and G_2 shown in Figure 6.6.1, $r(M(G_1)) = |V(G_1)| - 1 = 3$ and $r(M(G_2)) = |V(G_2)| - 3 = 3$. In each matroid, $\text{cl}(\{2, 5\}) = \{1, 2, 5, 6\}$ and the last set is a flat of rank 2.

6.6.3 The Greedy Algorithm

Matroids have a fundamental relationship to the greedy algorithm that makes them important in optimization problems. Kruskal's algorithm for finding a minimum-cost spanning tree in a connected graph G is one of the best-known efficient algorithms in graph theory. This algorithm works precisely because the spanning trees of G form the bases of a matroid.

Algorithm 6.6.1: Greedy Algorithm for $(\mathcal{I}, w)$

Let E be a finite set and $\mathcal{I}$ be a nonempty hereditary subset of 2^E. Let w be a real-valued function on E, and, for $X \subseteq E$, let the ***weight*** $w(X)$ be $\sum_{x \in X} w(x)$.

```
(i) Set $X_0 = \emptyset$ and $j = 0$.
(ii) If $E - X_j$ contains an element $e$ such that $X_j \cup \{e\} \in \mathcal{I}$, choose such an element $e_{j+1}$ of maximum weight, let $X_{j+1} = X_j \cup \{e_{j+1}\}$, and go to (iii); otherwise let $B_G = X_j$ and go to (iv).
(iii) Add 1 to $j$ and go to (ii).
(iv) Stop.
```

EXAMPLE

E5: Let G be a connected graph with each edge e having a cost $c(e)$. Define $w(e) = -c(e)$. Then the greedy algorithm is just Kruskal's algorithm, and the result B_G is the edge-set of a spanning tree of minimum cost.

FACT

F8: A nonempty hereditary set $\mathcal{I}$ of subsets of a finite set E is the set of independent sets of a matroid on E if and only if, for all real-valued weight functions w on E, the set B_G produced by the greedy algorithm is a maximal member of $\mathcal{I}$ of maximum weight.

6.6.4 Duality

Matroid theory has an attractive theory of duality that extends both the concept of a planar dual of a plane graph and the notion of orthogonality in vector spaces. This duality means that every graph gives rise to another matroid in addition to its cycle matroid.

DEFINITIONS

D14: The ***dual*** M^* ***of a matroid*** M is the matroid on $E(M)$ having the set

$$\mathcal{B}^*(M) = \{E(M) - B : B \in \mathcal{B}(M)\}$$

as its set of bases (see Fact 9 below).

D15: The ***cocircuits***, ***cobases***, ***coloops***, and ***coindependent sets*** of a matroid M are the circuits, bases, loops, and independent sets of the dual matroid M^*.

D16: For a graph G, $(M(G))^*$ is called the ***bond matroid*** of G and is denoted by $M^*(G)$. A matroid M is ***cographic*** if $M \cong M^*(G)$ for some graph G.

D17: A class of matroids is ***closed under duality*** if the dual of every member of the class is also in the class.

EXAMPLES

E6: Table 6.6.3 specifies the duals of certain matroids.

Table 6.6.3 Duals of some basic examples.

Matroid	$M(G)$ for G plane	$U_{m,n}$	$M[I_r \| D]$ for $r \times n$ matrix $[I_r \| D]$
Dual	$M(G^*)$ where G^* is the dual of G	$U_{n-m,n}$	$M[-D^T \| I_{n-r}]$, same order of column labels as $[I_r \| D]$

E7: The graph G_1^* in Figure 6.6.2 is the planar dual of the graph G_1 in Figure 6.6.1. Observe that $M(G_1^*)$ is isomorphic to $M(G_1)$ under the permutation of $E(G_1)$ that interchanges 1, 3, and 4 with 2, 5, and 6, respectively.

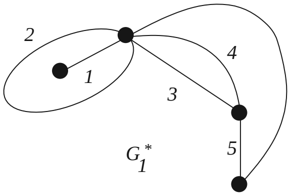


Figure 6.6.2 **Graph G_1^* is the planar dual of G_1.**

FACTS

For all matroids M:

F9: The set $\mathcal{B}^*(M) = \{E(M) - B : B \in \mathcal{B}(M)\}$ is the set of bases of a matroid on $E(M)$.

F10: $(M^*)^* = M$.

F11: The rank function of M^* is $r^*(X) = |X| - r(M) + r(E - X)$.

F12: The cocircuits of M are the minimal sets having nonempty intersection with every basis of M.

F13: The cocircuits of M are the minimal nonempty sets C^* such that $|C^* \cap C| \neq 1$ for every circuit C of M.

F14: For a graph G, the circuits of $M^*(G)$ are the ***bonds*** or minimal edge-cuts of G. In particular, the loops of $M^*(G)$ are the isthmuses of G.

F15: A graphic matroid is cographic if and only if it is ***planar*** (isomorphic to the cycle matroid of a planar graph).

F16: Every row of the matrix $[I_r|D]$ is orthogonal to every row of $[-D^T|I_{n-r}]$.

F17: The following classes of matroids are closed under duality: uniform matroids, matroids representable over a fixed field $\mathbf{F}$, planar matroids, and regular matroids. The classes of graphic and cographic matroids are not closed under duality.

Table 6.6.4 Special sets and their complements in M and M^*.

X	basis of M	independent set of M	circuit of M
$E - X$	basis of M^*	spanning set of M^*	hyperplane of M^*

REMARKS

R2: The fact that both the cycles and bonds of a graph are the circuits of a matroid means that cycles and bonds share many common properties.

R3: Matroids in general do not have vertices. In a 2-connected loopless graph G, the set of edges meeting a vertex forms a bond of G and hence a cocircuit of $M(G)$. Although $M(G)$ will usually have many cocircuits that do not arise in this way, in many contexts, an appropriate matroid analog of a vertex is a cocircuit.

6.6.5 Matroid Union and Its Consequences

The operation of matroid union, which was introduced by Nash-Williams [Na66], led to very straightforward proofs of two graph results whose original proofs were quite intricate.

DEFINITION

D18: The ***union of the matroids*** $M_1, M_2, \ldots, M_n$ on a common ground set E is the matroid whose independent sets are all subsets of E of the form $I_1 \cup I_2 \cup \ldots \cup I_n$ such that $I_i \in \mathcal{I}(M_i)$ for all i. (See Fact 18.) It is denoted by $M_1 \vee M_2 \vee \ldots \vee M_n$.

FACTS

F18: Let $M_1, M_2, \ldots, M_n$ be matroids on a common ground set E. Then the collection of subsets of E of the form $I_1 \cup I_2 \cup \ldots \cup I_n$ such that $I_i \in \mathcal{I}(M_i)$ for all i is the collection of independent sets of a matroid. Thus, matroid union is well-defined.

F19: If M_i has rank r_i, then the rank of X in $M_1 \vee M_2 \vee \ldots \vee M_n$ is

$$\min\{\sum_{i=1}^{n} r_i(Y) + |X - Y| : Y \subseteq X\}$$

Covering and Packing Results

The following covering and packing results for matroids are easily proved by taking the union of a matroid with itself multiple times, although the original proofs preceded the introduction of the operation of matroid union. Facts 22 and 23 are obtained by applying Facts 20 and 21, respectively, to graphs. Notice that Facts 20 and 21 are duals of each other.

FACTS

F20: [Ed65] A matroid M has k disjoint bases if and only if, for every subset X of $E(M)$,

$$kr(X) + |E(M) - X| \geq kr(M)$$

F21: [Ed65] A matroid M has k independent sets whose union is $E(M)$ if and only if, for every subset X of $E(M)$,

$$kr(X) \geq |X|$$

F22: [Tu61, Na61] A connected graph G has k edge-disjoint spanning trees if and only if, for every partition π of $V(G)$, the number of edges joining vertices in different classes of the partition is at least $k(|\pi| - 1)$ where $|\pi|$ is the number of classes in π.

F23: [Tu61] The edge-set of a graph G can be partitioned into k disjoint forests if and only if, for all subsets X of $V(G)$,

$$|E(G[X])| \geq k(|X| - 1)$$

F24: [Ed65] Let G be a connected graph. Players B and C alternately tag edges of G where an edge is destroyed if it is tagged by C and made invulnerable to destruction if it is tagged by B. The goal for B is to tag all the edges of some spanning tree of G and the goal for C is to prevent B from achieving this goal (by tagging all the edges of some bond). The following are equivalent:

(i) Player C plays first and B can win against all possible strategies of C.

(ii) G has two edge-disjoint spanning trees.

(iii) For all partitions π of $V(G)$, the number of edges joining vertices in different classes of the partition is at least $2(|\pi| - 1)$.

6.6.6 Fundamental Operations

Duality is one of the three basic operations for matroids. Two other basic operations, ***deletion*** and ***contraction***, are defined in Table 6.6.5 below, along with the operation of ***direct sum***, a special case of matroid union which generalizes the operation of direct sum of vector spaces. Each of these operations generalizes an operation for graphs.

DEFINITIONS

D19: The matroids $M\backslash T$ and M/T are also written as $M|(E-T)$ and $M.(E-T)$ and are called the ***restriction*** and ***contraction of*** M ***to*** $E - T$.

D20: A matroid N is a ***minor*** of a matroid M if N can be obtained from M by a sequence of deletions and contractions. The minor N is ***proper*** if $N \neq M$.

D21: A graph H is a ***minor*** of a graph G if H can be obtained from G by a sequence of edge deletions, edge contractions, and deletions of isolated vertices.

NOTATION: For an element e of a matroid M, the matroids $M\backslash\{e\}$ and $M/\{e\}$ are frequently written as $M\backslash e$ and M/e.

Table 6.6.5 Three basic matroid constructions.

Matroid	Ground Set	$\mathcal{C}$	$\mathcal{I}$
$M\backslash T$, the **deletion** of T from M	$E(M)-T$	$\{C \subseteq E(M)-T : C \in \mathcal{C}(M)\}$	$\{I \subseteq E(M)-T : I \in \mathcal{I}(M)\}$
M/T, the **contraction** of T from M	$E(M)-T$	minimal non-empty members of $\{C-T : C \in \mathcal{C}(M)\}$	$\{I \subseteq E(M)-T : I \cup B_T \in \mathcal{I}(M)$ for some B_T in $\mathcal{B}(M\|T)\}$
$M_1 \oplus M_2$, **direct sum** of M_1 and M_2, $E(M_1) \cap E(M_2) = \emptyset$	$E(M_1) \cup E(M_2)$	$\mathcal{C}(M_1) \cup \mathcal{C}(M_2)$	$\{I_1 \cup I_2 : I_j \in \mathcal{I}(M_j)\}$

EXAMPLES

E8: $M(G)\backslash e = M(G\backslash e)$, where $G\backslash e$ is obtained from the graph G by deleting edge e.

E9: $M(G)/e = M(G/e)$, where G/e is obtained from the graph G by contracting the edge e, that is, by identifying the endpoints of e and then removing e.

E10: $U_{m,n}\backslash e = U_{m,n-1}$ unless $m = n$ when $U_{m,n}\backslash e = U_{m-1,n-1}$.

E11: $U_{m,n}/e = U_{m-1,n-1}$ unless $m = 0$ when $U_{m,n}/e = U_{m,n-1}$.

E12: $M[A]\backslash e$ is the vector matroid of the matrix obtained by deleting column e from matrix A.

E13: If e corresponds to a unit vector in A, then $M[A]/e$ is the vector matroid of the matrix obtained by deleting both the column e and the row containing the one of e.

E14: If G_1 and G_2 are vertex-disjoint graphs, then $M(G_1) \oplus M(G_2)$ is the cycle matroid of the graph that is obtained by taking the disjoint union of G_1 and G_2. Moreover, if v_1 is a vertex of G_1 and v_2 is a vertex of G_2, then $M(G_1) \oplus M(G_2)$ is also the cycle matroid of the graph that is obtained by identifying v_1 and v_2, this graph being a 1-***sum*** of G_1 and G_2.

FACTS

In the following, M, M_1, and M_2 are matroids and $E(M_1) \cap E(M_2) = \emptyset$.

F25: $(M/T)^* = M^*\backslash T$; and $(M\backslash T)^* = M^*/T$. (Deletion and contraction are dual operations.)

F26: If X and Y are disjoint subsets of $E(M)$, then $M\backslash X\backslash Y = M\backslash(X \cup Y) = M\backslash Y\backslash X$; $M/X/Y = M/(X \cup Y) = M/Y/X$; and $M\backslash X/Y = M/Y\backslash X$.

F27: If $X \subseteq E(M) - T$, then $r_{M/T}(X) = r_M(X \cup T) - r_M(T)$.

F28: $M_1 \oplus M_2 = M_2 \oplus M_1$.

F29: Let N_1 and N_2 be the rank-zero matroids on $E(M_1)$ and $E(M_2)$. Then $M_1 \oplus M_2 = (M_1 \oplus N_2) \vee (N_1 \oplus M_2)$.

6.6.7 2- and 3-Connectedness for Graphs and Matroids

Although connectedness for graphs does not carry over to matroids, 2-connectedness and 3-connectedness do.

DEFINITION

D22: A ***2-connected matroid*** is a matroid M such that, for every two distinct elements e and f of $E(M)$, there is a circuit containing $\{e, f\}$.

TERMINOLOGY: For matroids, the terms "2-connected" and "connected" are used interchangeably. Another synonym that is also used is "non-separable".

EXAMPLE

E15: In Figure 6.6.1, the graph G_1 is connected and G_2 is not, and yet, $M(G_1) = M(G_2)$. Thus, in general, one cannot tell from its matroid $M(G)$ whether or not a graph G is connected.

FACTS

F30: Let G be a graph without loops or isolated vertices and assume that $|V(G)| \geq 3$. Then G is 2-connected if and only if, for every two distinct edges e and f of G, there is a cycle of G containing $\{e, f\}$.

F31: A matroid M is 2-connected if and only if M cannot be written as the direct sum of two matroids with nonempty ground sets.

F32: A matroid is 2-connected if and only if its dual is 2-connected.

F33: [Tu65] If M is 2-connected and $e \in E(M)$, then $M\backslash e$ or M/e is 2-connected.

F34: [Le64] If M is 2-connected, then M is uniquely determined by the set of circuits containing some fixed element of $E(M)$.

By combining Facts 30 and 32, one obtains the following:

F35: Let G be a graph without loops or isolated vertices and assume that $|V(G)| \geq 3$. Then G is 2-connected if and only if, for every two distinct edges e and f of G, there is a bond of G containing $\{e, f\}$.

Bounds on the Number of Elements

NOTATION: For a matroid M having a circuit and a cocircuit, let $c(M)$ and $c^*(M)$ be the sizes of, respectively, a largest circuit and a largest cocircuit of M.

NOTATION: If $e \in E(M)$ and e is not a loop or a coloop, let $c_e(M)$ and $c^*_e(M)$ be the sizes of, respectively, a largest circuit of M containing e and a largest cocircuit of M containing e; and let $d_e(M)$ and $d^*_e(M)$ be the sizes of a smallest circuit of M containing e and a smallest cocircuit of M containing e.

F36: [LeOx01] Let M be a 2-connected matroid with at least two elements.

(i) If e is an element of M, then $|E(M)| \leq (c_e(M) - 1)(c^*_e(M) - 1) + 1$.

(ii) $|E(M)| \leq \frac{1}{2}c(M)c^*(M)$.

F37: (Length-width inequality [Le79]) Let M be a regular matroid, and suppose that $e \in E(M)$ and that e is not a loop or a coloop. Then

$$|E(M)| \geq (d_e(M) - 1)(d^*_e(M) - 1) + 1$$

The next two facts for graphs are immediate consequences of Fact 36.

F38: Let u and v be distinct vertices in a 2-connected loopless graph G. Then $|E(G)|$ cannot exceed the product of the length of a longest $u-v$ path and the size of a largest bond separating u from v.

F39: [Wu97] Let G be a 2-connected loopless graph with circumference c and let c^* be the size of a largest bond. Then

$$|E(G)| \leq \frac{1}{2}cc^*$$

Wu [Wu00] showed that the graphs attaining equality in the last bound are certain series-parallel graphs, including cycles. Wu's bound is sometimes better and sometimes worse than the following bound, whose hypotheses are slightly different.

F40: (Erdős and Gallai) [ErGa59] Let G be a simple graph with circumference c. Then

$$|E(G)| \leq \frac{1}{2}c(|V(G)| - 1)$$

The last bound motivated the question whether Fact 36(ii) is true for matroids. This question was answered for graphs before it was answered for all matroids. The following is a generalization of Fact 39.

F41: [NeRiUr99] Every 2-connected loopless graph with circumference c has a collection of c bonds such that every edge lies in at least two of them.

The (matroid) dual of the last result is also true.

F42: [Mc04?] Every 2-connected loopless graph whose largest bond has size c^* has a family of c^* cycles so that every edge lies in at least two of them.

The last result was proved as a partial answer to the following problem of Vertigan (in [Ox01]), which remains open in general.

OPEN PROBLEM

P1: Let M be a 2-connected matroid with at least two elements. Does M have a family of $c(M)$ cocircuits such that every element is in at least two of them?

2-sums and 3-sums

As noted in Example E14, matroid direct sum generalizes the operation of 1-sum for graphs. The graph operation of 2-sum generalizes to all matroids, while 3-sum generalizes to binary matroids.

DEFINITIONS

D23: The ***2-sum of two 2-connected matroids*** M_1 and M_2 on disjoint sets, each with at least three elements, ***with respect to*** $p_1 \in E(M_1)$ and $p_2 \in E(M_2)$ is the matroid with ground set $(E(M_1) - \{p_1\}) \cup (E(M_2) - \{p_2\})$, for which the circuits are all circuits of M_1 avoiding p_1, all circuits of M_2 avoiding p_2, and all sets of the form $(C_1 - \{p_1\}) \cup (C_2 - \{p_2\})$ where C_i is a circuit of M_i containing p_i. It is denoted by $M_1 \oplus_2 M_2$.

D24: A ***3-connected matroid*** is a matroid that is 2-connected and cannot be written as a 2-sum.

D25: Let M_1 and M_2 be binary matroids each having at least seven elements. Suppose that $E(M_1) \cap E(M_2) = T$ where T is a 3-element circuit in each of M_1 and M_2 and T does not contain a cocircuit of M_1 or M_2. Then the ***3-sum of matroids*** M_1 and M_2 is the matroid on $(E(M_1) \cup E(M_2)) - T$ whose flats are those sets $F - T$ such that $F \cap E(M_i)$ is a flat of M_i for each i.

EXAMPLES

E16: Let G_1 and G_2 be 2-connected loopless graphs and p_i be an edge of G_i for each i. Let G be one of the two graphs that can be obtained from G_1 and G_2 by identifying p_1 with p_2 and then deleting the identified edge, that is, G is a 2-sum of the graphs G_1 and G_2. Then $M(G_1) \oplus_2 M(G_2) = M(G)$.

E17: Let G_1 and G_2 be the graphs in Figure 6.6.3, where $E(G_1) \cap E(G_2) = \{1, 2, 3\}$. Then the graph G obtained by sticking G_1 and G_2 together across the 3-cycle $\{1, 2, 3\}$ and then deleting $\{1, 2, 3\}$ is the 3-sum of the graphs G_1 and G_2. The matroid $M(G)$ is the 3-sum of the matroids $M(G_1)$ and $M(G_2)$.

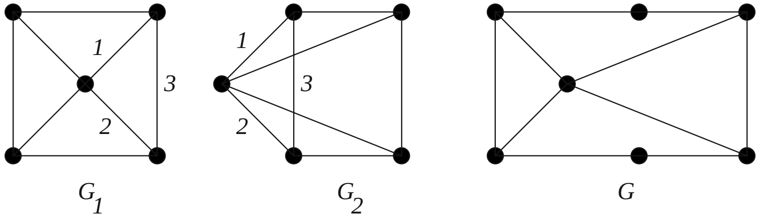


Figure 6.6.3 $M(G)$ **is the 3-sum of** $M(G_1)$ **and** $M(G_2)$**.**

FACTS

F43: $(M_1 \oplus_2 M_2)^* = M_1^* \oplus M_2^*$.

F44: A matroid M is 3-connected if and only if M^* is 3-connected.

F45: Let G be a graph without isolated vertices and suppose that $|V(G)| \geq 4$. Then $M(G)$ is 3-connected if and only if G is 3-connected and simple.

REMARK

R4: Tutte [Tu66] defined a notion of k-connectedness for matroids for all $k \geq 2$ that includes the cases considered above. It has the advantage of being preserved under matroid duality but the disadvantage that it departs from graph k-connectedness when $k \geq 4$. Several authors [Cu81, InWe81, Ox81] introduced the notion of vertical k-connectedness for matroids, which generalizes k-connectedness for graphs but is no longer closed under duality.

6.6.8 Graphs and Totally Unimodular Matrices

One of the most significant achievements of matroid theory is Seymour's result showing that all totally unimodular matrices are obtainable from graphs and one additional special matroid. This result leads to a polynomial-time algorithm to test whether a given matroid is totally unimodular, which is particularly useful in combinatorial optimization (see, for example, [Sc86]). Recall that a matroid is regular if and only if it is represented by a totally unimodular matrix.

EXAMPLES

E18: Let G be a graph. Arbitrarily orient the edges of G and let D be the vertex-edge incidence matrix of the resulting directed graph. Then D is a totally unimodular matrix that represents $M(G)$ over all fields.

E19: [Bi77] Consider the matrix A over $GF(2)$ whose columns consist of the ten 5-tuples with exactly three ones. Let R_{10} be the matroid represented by A. Then $R_{10}^* \cong R_{10}$. Moreover, if e is an element of R_{10}, then $R_{10} \backslash e \cong M(K_{3,3})$ and $R_{10}/e \cong M^*(K_{3,3})$.

FACT

F46: [Se80] The class of regular matroids is the class of matroids that can be constructed by direct sums, 2-sums, and 3-sums from graphic matroids, cographic matroids, and copies of R_{10}.

6.6.9 Excluded-Minor Characterizations

The Kuratowski-Wagner Theorem [Ku30, Wa37] that a graph is planar if and only if it has no minor isomorphic to K_5 or $K_{3,3}$ has a number of extensions for graphs and matroids. The search for such results is currently the most active area of research in matroid theory.

DEFINITIONS

D26: A class of matroids is ***minor-closed*** if every minor of a member of the class is also in the class.

D27: An ***excluded minor*** of a minor-closed class of matroids is a matroid N that is not in the class, such that every proper minor of N is in the class.

FACTS

F47: The following classes of matroids are minor-closed: graphic matroids, cographic matroids, uniform matroids, matroids representable over a fixed field, regular matroids, and planar matroids.

F48: [RoSe04?] For every minor-closed class of graphs, the set of excluded minors is finite.

F49: [La58] For every field **F** of characteristic 0 so, in particular, for **Q**, **R**, and **C**, the class of matroids representable over **F** has an infinite set of excluded minors.

EXAMPLES

E20: Given a finite set E of points in the plane and a collection of lines (subsets of E with at least three elements), no two of which share more than one common point, there is a matroid with ground set E whose circuits are all sets of three collinear points and all sets of four points no three of which are collinear. *Geometric representations* of two such matroids are shown in Figure 6.6.4, where the reader is cautioned that these diagrams are *not* to be interpreted as graphs. Each matroid depicted has ground set $\{1, 2, \ldots, 7\}$. On the left is the non-Fano matroid, F_7^-. It differs from the Fano matroid, F_7, on the right by the collinearity through $4, 5$, and 6 in the latter. Neither of these two matroids is graphic.

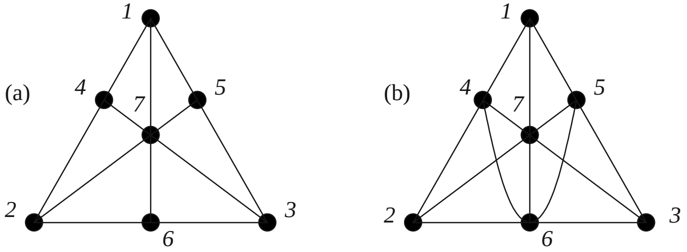


Figure 6.6.4 **(a) The non-Fano matroid, F_7^-. (b) The Fano matroid, F_7.**

E21: Table 6.6.6 specifies the collections of excluded minors for certain classes of matroids. The results in the last two rows of the table were proved in three landmark papers of Tutte [Tu58, Tu58a, Tu59]. The characterization of ternary matroids was proved independently by Bixby [Bi79] and Seymour [Se79].

Table 6.6.6 Excluded-minor characterizations of certain classes of matroids.

Class	Excluded minors	Class	Excluded minors
uniform	$U_{0,1} \oplus U_{1,1}$	ternary	$U_{2,5}, U_{3,5}, F_7, F_7^*$
binary	$U_{2,4}$	graphic	$U_{2,4}, F_7, F_7^*, M^*(K_5), M^*(K_{3,3})$
regular	$U_{2,4}, F_7, F_7^*$	cographic	$U_{2,4}, F_7, F_7^*, M(K_5), M(K_{3,3})$

E22: The class of simple matroids is not minor-closed since it contains the cycle matroid of a 3-edge cycle but not the cycle matroid of a 2-edge cycle.

CONJECTURES

The following two conjectures are the main unsolved problems in matroid theory.

C1: (Rota's Conjecture [Ro71]) For all finite fields $\mathbf{F}$, there is a finite set of excluded minors for the class of $\mathbf{F}$-representable matroids.

C2: For all finite fields $\mathbf{F}$, if $\mathcal{M}$ is some minor-closed class of matroids all of which are $\mathbf{F}$-representable, then there is a finite set of excluded minors for $\mathcal{M}$.

REMARKS

R5: Fact 48 is probably the deepest result ever proved in graph theory, appearing in the twentieth paper of a very difficult series. Fact 49 shows that Fact 48 does not extend to matroids. The two conjectures above propose two natural classes of matroids to which Fact 48 may be extendable.

R6: From Table 6.6.6, if $q \in \{2, 3\}$, then the set of excluded minors for the class of $GF(q)$-representable matroids is finite. Geelen, Gerards, and Kapoor [GeGeKa00] proved that the same is true for $GF(4)$, there being exactly seven excluded minors in this case. Rota's Conjecture remains open for all prime powers q exceeding 4. Some recent progress on this and on Conjecture C2 has been made by Geelen and Whittle [GeWh02] and by Geelen, Gerards, and Whittle [GeGeWh02].

6.6.10 Wheels, Whirls, and the Splitter Theorem

Tutte [Tu61] identified wheels as the basic building blocks of 3-connected simple graphs. Subsequently, he generalized that result to matroids [Tu66]. The Splitter Theorem, a powerful generalization of the last result, was proved for matroids by Seymour [Se80] and, independently, for graphs by Negami [Ne82].

DEFINITIONS

D28: For $n \geq 2$, the ***wheel*** $\mathcal{W}_n$ is the graph that is formed from an n-cycle C_n by adding a new vertex and joining this by a single edge (a ***spoke***) to every vertex of the ***rim*** C_n.

D29: For $r \geq 2$, the ***rank-r whirl*** $\mathcal{W}^r$ is the matroid on the edge set of $\mathcal{W}_r$ whose set of circuits consists of all the cycles of $\mathcal{W}_r$ except the rim, together with all sets of edges consisting of the rim plus a single spoke.

D30: Let M and N be matroids. Then M ***has an N-minor*** if M has a minor isomorphic to N.

D31: An ***n-spike*** with tip p is a rank-n matroid whose ground set is the union of n 3-element circuits $C_1, C_2, \ldots, C_n$ all containing a common point p such that, for all $k \leq n-1$, the union of any k of $C_1, C_2, \ldots, C_n$ has rank $k+1$.

EXAMPLES

E23: Figure 6.6.5 shows the graph $\mathcal{W}_3$, which is clearly isomorphic to K_4, together with geometric representations of the matroids $M(\mathcal{W}_3)$ and $\mathcal{W}^3$. The line $\{4, 5, 6\}$ in $M(\mathcal{W}_3)$ corresponds to the rim of $\mathcal{W}_3$.

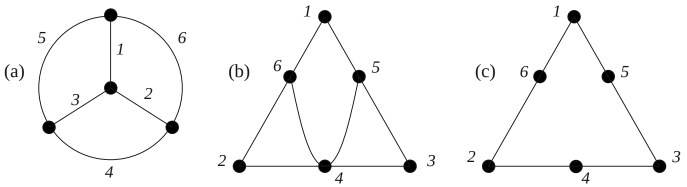


Figure 6.6.5 **(a) The graph $\mathcal{W}_3$. The matroids (b) $M(\mathcal{W}_3)$ and (c) $\mathcal{W}^3$.**

E24: Both the Fano and non-Fano matroids are examples of 3-spikes.

E25: The unique rank-r binary spike is the vector matroid of the matrix $[I_r|I_r^c|\mathbf{1}]$ over $GF(2)$ where I_r^c is the matrix obtained by interchanging the zeros and ones in the $r \times r$ identity matrix I_r, and $\mathbf{1}$ is the column of all ones.

FACTS

F50: [Tu61] Let G be a simple 3-connected graph. The following two conditions are equivalent:

(i) For every edge e of G, neither $G\backslash e$ nor G/e is both simple and 3-connected.

(ii) G is isomorphic to a wheel with at least three spokes.

F51: [Tu66] Let M be a 3-connected matroid. The following two conditions are equivalent:

(i) For every element e of M, neither $M\backslash e$ nor M/e is 3-connected.

(ii) M is isomorphic to $M(\mathcal{W}_r)$ or $\mathcal{W}^r$ for some $r \geq 3$.

F52: (The Splitter Theorem [Se80]) Let M and N be 3-connected matroids such that N is a minor of M, with $|E(N)| \geq 4$, and M is neither a whirl nor the cycle matroid of a wheel. Suppose that if $N \cong \mathcal{W}^2$, then M has no $\mathcal{W}^3$-minor; and if $N \cong M(\mathcal{W}_3)$, then M has no $M(\mathcal{W}_4)$-minor. Then there is a sequence of 3-connected matroids

$$M_0, M_1, \ldots, M_n$$

such that $M_0 = M; M_n \cong N$; and, for all i in $\{1, 2, \ldots, n\}$, M_i is a single-element deletion or a single-element contraction of M_{i-1}.

The statement above of the Splitter Theorem is a slight strengthening, due to Coullard [Co85] (see also [CoOx92]), of Seymour's original result. The Splitter Theorem has numerous applications for both graphs and matroids. It played a key role in the proof of Fact 46 and can also be used to derive the following results, the first two of which preceded the Splitter Theorem.

F53: [Wa60] Let G be a simple 3-connected graph having no K_5-minor. Then either G has no H_8-minor or $G \cong H_8$, where H_8 is the 4-rung Möbius ladder shown in Figure 6.6.6.

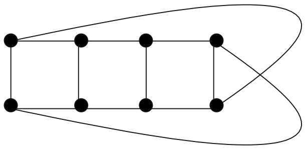


Figure 6.6.6 **The 4-rung Möbius ladder, H_8.**

F54: [Ha43] Let G be a simple 3-connected graph. Then G has no $K_{3,3}$-minor if and only if either G is planar or $G \cong K_5$.

F55: [Se80] Let M be a 3-connected binary matroid. Then M has no F_7-minor if and only if M is regular or $M \cong F_7^*$.

F56: Let G be a simple 3-connected graph. Then G has no $\mathcal{W}_4$-minor if and only if $G \cong \mathcal{W}_3$.

F57: [Ox89] Let G be a simple 3-connected graph. Then G has no $\mathcal{W}_5$-minor if and only if

(i) G is isomorphic to a simple 3-connected minor of one of the four graphs in Figure 6.6.7; or

(ii) for some $k \geq 3$, the graph G is obtained from $K_{3,k}$ by adding up to three edges joining distinct pairs of vertices in the 3-vertex class of the bipartition.

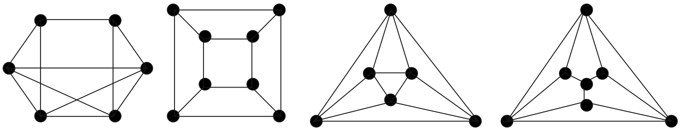


Figure 6.6.7 **Four graphs with no 5-wheel minor.**

The last fact and a result of Gubser [Gu93] motivated the following result, which, in turn, motivated the result following it.

F58: [OpOxTh93] For every integer $n \geq 3$, there is an integer N such that every 3-connected simple graph with at least N edges has a minor isomorphic to $\mathcal{W}_n$ or $K_{3,n}$.

F59: [DiOpOxVe97] For every integer $n \geq 3$, there is an integer N such that every 3-connected matroid with at least N elements has a minor isomorphic to

$$U_{2,n}, U_{n-2,n}, M(K_{3,n}), M^*(K_{3,n}), M(\mathcal{W}_n), \mathcal{W}^n$$

or to an n-spike.

6.6.11 Removable Circuits

A result of Mader gave conditions under which a simple k-connected graph has a cycle whose edges can be deleted without destroying k-connectedness. The natural matroid generalization of this fails for $k = 2$ even for cographic matroids. However, loose analogs of Mader's result hold for 2- and 3-connected matroids and these give new results for graphs.

DEFINITIONS

D32: A cycle C of a k-connected graph G is *removable* if the graph obtained from G by deleting all the edges of C is k-connected.

D33: For k in $\{2,3\}$, a circuit D of a k-connected matroid M is *removable* if $M \backslash D$ is k-connected.

FACT

F60: (Mader) [Ma74] If G is a simple k-connected graph with minimum degree at least $k+2$, then G has a removable cycle.

The hypothesis of Mader's theorem implies that $|E(G)| \geq \frac{1}{2}(k+2)|V(G)|$. The next two facts show that imposing appropriate lower bounds on the number of elements in a matroid guarantees the existence of removable circuits.

F61: [LeOx99] Let M be a 2-connected matroid with at least two elements and C' be a largest circuit of M. If $|E(M)| \geq 3r(M) + 3 - c(M)$, then M has a circuit C that is disjoint from C' such that $M \backslash C$ is 2-connected and $r(M \backslash C) = r(M)$. In particular, if $r(C') = r(M)$ and $|E(M)| \geq 2r(M) + 2$, then M has a removable circuit.

F62: [LeOx99a] Let M be a 3-connected matroid with at least two elements and C' be a largest circuit of M. If

$$|E(M)| \geq \begin{cases} 3r(M) + 1 & \text{when } c(M) = r(M) + 1, \\ 4r(M) + 1 - c(M) & \text{otherwise} \end{cases}$$

then M has a circuit C that is disjoint from C' such that $M \backslash C$ is 3-connected and $r(M \backslash C) = r(M)$.

The next two facts are obtained by applying the previous two results to graphs.

F63: Let G be a 2-connected loopless graph and C' be a largest cycle in G. If $|E(G)| \geq 3|V(G)| - c(G)$, then G has a removable cycle having no common edges with C'. In particular, if G is hamiltonian and $|E(G)| \geq 2|V(G)|$, then G has a removable cycle.

F64: Let G be a simple 3-connected graph and C' be a largest cycle of G. Suppose that

$$|E(G)| \geq \begin{cases} 3|V(G)| - 2 & \text{if } G \text{ is hamiltonian}, \\ 4|V(G)| - 3 - c(G) & \text{otherwise} \end{cases}$$

Then G has a cycle C that has no common edges with C' such that $G \backslash C$ is 3-connected.

F65: [GovaMc97] Let G be a 2-connected graph with minimum degree at least four. If G has no minor isomorphic to the Petersen graph, then G has two edge-disjoint removable cycles.

F66: [Mc04?a] Let G be a 2-connected graph that is not a multiple edge. If G has no minor isomorphic to K_5, then G has a bond C^* such that G/C^* is 2-connected.

For 2-connected graphs, the condition that the graph is simple in Mader's result (F60) can be replaced by a higher bound on the minimum degree.

F67: [Si98] Let G be a 2-connected graph with minimum degree at least five. Then G has a removable cycle.

Example E29 below shows that the previous result does not generalize to all matroids. The next result implies that it generalizes to regular and hence cographic matroids and prompts the problem as to whether it extends to binary matroids.

F68: [GoJa99] Let M be a 2-connected binary matroid in which every cocircuit has at least five elements. If M does not have minors isomorphic to both F_7 and F_7^*, then M has a removable circuit C such that $r(M \backslash C) = r(M)$.

REMARKS

R7: The last sentence of Fact 63 is easily deduced directly, but the result in the case when G is non-hamiltonian seems far less obvious.

R8: Facts 61 and 62 can also be applied to the bond matroids of, respectively, 2-connected loopless graphs and 3-connected simple graphs, to give necessary conditions for such a graph G to have a bond C^* for which G/C^* is, respectively, 2-connected and loopless, or 3-connected and simple.

R9: Arthur Hobbs provided much of the impetus for the study of removable cycles by asking whether every 2-connected Eulerian graph with minimum degree at least four contains a removable cycle.

EXAMPLES

E26: [LeOx99] Consider the simple graph that is constructed as follows: begin with $K_{5,5}$ having as its two vertex classes $\{1,2,3,4,5\}$ and $\{6,7,8,9,10\}$; for every 3-element subset X of $\{1,2,3,4,5\}$ and of $\{6,7,8,9,10\}$, add two new vertices v_X and w_X each joined to all the members of X and to nothing else. Then the resulting graph G is 2-connected having every cycle of length at least four and having every bond of size at least three. Thus $M^*(G)$ is simple and 2-connected having every cocircuit of size at least four. But $M^*(G)$ has no removable circuit because G has no bond C^* for which G/C^* is 2-connected. Thus the generalization of Fact 60 to cographic matroids fails when $k = 2$.

E27: Jackson [Ja80] and, independently, Robertson (in [Ja80]) answered Hobbs's question (R9) negatively by producing the modified Petersen graph in Figure 6.6.8(a).

E28: [GovaMc97] For the dual problem, the graph G in Figure 6.6.8(b) is 2-connected but has no bond C^* such that G/C^* is 2-connected. This motivated a conjecture, which McGuinness proved in Fact 66.

E29: For $r \geq 3$, the uniform matroid $U_{r,2r}$ is 2-connected, has all its cocircuits and circuits of cardinality $r+1$, and has no removable circuits.

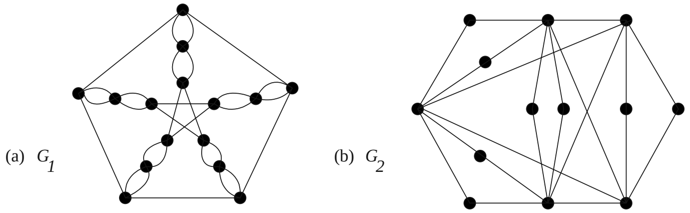


Figure 6.6.8 Neither $M(G_1)$ nor $M^*(G_2)$ has a removable circuit.

PROBLEMS

P2(i): [GoJa99] Is there an integer t such that every 2-connected binary matroid in which every cocircuit has at least t elements has a removable circuit?

P2(ii): [GoJa99] If M is a 2-connected binary matroid in which every cocircuit has at least 5 elements, then does M have a removable circuit?

6.6.12 Minimally k-Connected Graphs and Matroids

For $k \geq 2$, a k-connected graph for which no single-edge deletion is k-connected has many vertices of degree k. For k in $\{2, 3\}$, this fact has some matroid analogs that lead to new graph results.

DEFINITIONS

D34: For $k \geq 2$, a k-connected graph G is ***minimally k-connected*** if no single-edge deletion of G is k-connected.

D35: For k in $\{2, 3\}$, a k-connected matroid M is ***minimally k-connected*** if no single-element deletion of M is k-connected.

D36: Let M be a 2-connected matroid. A cocircuit C^* of M is ***nonseparating*** if $M \backslash C^*$ is 2-connected.

EXAMPLES

E30: If $m \geq k \geq 2$, then $K_{k,m}$ is minimally k-connected. For all $n \geq 3$, the n-spoked wheel $\mathcal{W}_n$ is minimally 3-connected. The cycle matroids of $K_{3,m}$ and $\mathcal{W}_n$ are minimally 3-connected matroids.

E31: The duals of the matroids F_7 and F_7^- are both minimally 3-connected.

FACTS FOR ARBITRARY CONNECTIVITY

F69: [Ma72] For all $k \geq 2$, every cycle of a minimally k-connected graph meets a vertex of degree k.

F70: [Ma79] For all $k \geq 2$, the number of vertices of degree k in a minimally k-connected graph G is at least

$$\frac{(k-1)|V(G)| + 2k}{2k-1}$$

F71: [Ox81b] For all $k \geq 2$, the number of vertices of degree k in a minimally k-connected graph G is at least

$$\frac{|E(G)| - |V(G)| + 1}{k-1}$$

REMARKS

R10: Fact 70 was proved when $k = 2$ by Dirac [Di67] and Plummer [Pl68], independently, and when $k = 3$ by Halin [Ha69]. The same authors proved Fact 73 below.

R11: The bound in Fact 71, which was obtained from Fact 69 by using an elementary matroid argument, frequently sharpens the bound in Fact 70.

R12: [Ma96] The graph that is obtained from a path P of length three by adding three vertices joined to each vertex of the path is minimally 4-connected but has a cycle meeting only one vertex of degree 4. Thus Fact 69 cannot be sharpened in general although it can be improved for $k \leq 3$.

FACTS FOR SMALL CONNECTIVITY

F72: If M is a 3-connected matroid and $M = M(G)$ for some loopless graph G without isolated vertices, then a subset C^* of $E(M)$ is a nonseparating cocircuit of M if and only if C^* is the set of edges meeting at some vertex of G.

F73: [Di67, Pl68, Ha69] For k in $\{2, 3\}$, every cycle of a minimally k-connected graph meets at least two vertices of degree k.

F74: [Ox81a, Ox81b] For k in $\{2, 3\}$, let M be a minimally k-connected matroid with at least four elements. Then

(i) every circuit of M meets at least two k-element cocircuits; and

(ii) M has at least $\dfrac{r^*(M) + (k-1)}{k-1}$ k-element cocircuits.

F75: [Wu98] Let M be a minimally 3-connected binary matroid with at least four elements. Then

(i) every circuit of M meets at least two 3-element nonseparating cocircuits; and

(ii) M has at least $\dfrac{r^*(M) + 2}{2}$ 3-element nonseparating cocircuits.

F76: [Ha69a] Let G be a minimally 3-connected graph. Then $\dfrac{3|V(G)|}{2} \le |E(G)|$. Moreover,

$$|E(G)| \le \begin{cases} 2|V(G)| - 2 & \text{if } |V(G)| \le 6; \\ 3|V(G)| - 9 & \text{if } |V(G)| \ge 7. \end{cases}$$

The only graphs attaining equality in these bounds are $\mathcal{W}_n$ for $3 \le n \le 6$ and $K_{3,m}$ for $m \ge 4$.

F77: [Ox81a] Let M be a minimally 3-connected matroid with at least four elements. Then

$$|E(M)| \le \begin{cases} 2r(M) & \text{if } r(M) \le 5; \\ 3r(M) - 6 & \text{if } r(M) \ge 6. \end{cases}$$

The only binary matroids attaining equality in these bounds are $M(\mathcal{W}_n)$ for $3 \le n \le 6$ and $M(K_{3,m})$ for $m \ge 4$.

On combining F73 and F74(ii) and using a small amount of additional argument, one gets the following:

F78: [Ox81b] Let G be a minimally 3-connected graph and G have n_3 vertices of degree 3. Then

$$n_3 \ge \begin{cases} \dfrac{2|V(G)| + 7}{5} & \text{when } \dfrac{3|V(G)|}{2} \le |E(G)| \le \dfrac{9|V(G)| - 3}{5}; \\[2ex] \dfrac{|E(G)| - |V(G)| + 3}{2} & \text{when } \dfrac{9|V(G)| - 3}{5} \le |E(G)| \le 3|V(G)| - 9. \end{cases}$$

Applying F74(ii) and F75(ii) to cographic matroids gives the following:

F79: Let G be a graph.

(i) If G is 2-connected and loopless having no single-edge contraction that is 2-connected and loopless, then the number of 2-cycles in G is at least $|V(G)|$.

(ii) Let G be a simple 3-connected graph for which no single-edge contraction is both simple and 3-connected. Then the number of 3-cycles C in G such that G/C is 2-connected and loopless is at least $\frac{|V(G)|+1}{2}$.

6.6.13 Conclusion

Many areas of the interaction between graphs and matroids have not been discussed above. The most notable omission relates to the Tutte polynomial and, in particular, to colorings and flows. (See the surveys in [BrOx92] and [We99].) In spite of this omission, numerous examples of this interaction are provided above. These include examples of matroid results that produce new graph results when applied to graphic or cographic matroids, graph results that have generalizations or analogs for matroids, and graph results that arise by viewing graphs from a matroid perspective. The already strong ties between matroid theory and graph theory are continuing to grow.

References

[Bi77] R. E. Bixby, Kuratowski's and Wagner's theorems for matroids, *J. Combin. Theory Ser. B* 22 (1977), 31–53.

[Bi79] R. E. Bixby, On Reid's characterization of the ternary matroids, *J. Combin. Theory Ser. B* 26 (1979), 174–204.

[BrOx92] T. Brylawski and J. Oxley, The Tutte polynomial and its applications, pp. 123–225 in *Matroid Applications* (ed. N. White), Cambridge University Press, 1992.

[Co85] C. R. Coullard, "Minors of 3-connected Matroids and Adjoints of Binary Matroids", Ph.D. thesis, Northwestern University, 1985.

[CoOx92] C. R. Coullard and J. G. Oxley, Extensions of Tutte's Wheels-and-Whirls Theorem, *J. Combin. Theory Ser. B* 56 (1992), 130–140.

[Cu81] W. H. Cunningham, On matroid connectivity, *J. Combin. Theory Ser. B* 30 (1979), 94–99.

[DiOpOxVe97] G. Ding, B. Oporowski, J. Oxley, and D. Vertigan, Unavoidable minors of large 3-connected matroids, *J. Combin. Theory Ser. B* 71 (1997), 244–299.

[Ed65] J. Edmonds, Lehman's switching game and a theorem of Tutte and Nash-Williams, *J. Res. Nat. Bur. Standards Sect. B* 69B (1965), 67–72.

[ErGa59] P. Erdős and T. Gallai, On maximal paths and circuits of graphs, *Acta Math. Acad. Sci. Hungar.* 10 (1959), 337–356.

[GeWh02] J. Geelen and G. Whittle, Branch-width and Rota's conjecture, *J. Combin. Theory Ser. B* 86 (2002), 315–330.

[GeGeKa00] J. F. Geelen, A. M. H. Gerards, and A. Kapoor, The excluded minors for $GF(4)$–representable matroids, *J. Combin. Theory Ser. B* 79 (2000), 247–299.

[GeGeWh02] J. Geelen, A. M. H. Gerards, and G. Whittle, Branch-width and well-quasi-ordering in matroids and graphs, *J. Combin. Theory Ser. B* 86 (2002), 148-171.

[GoJa99] L. A. Goddyn and B. Jackson, Removable circuits in binary matroids, *Combin. Probab. Comput.* 6 (1999), 539–545.

[GovaMc97] L. A. Goddyn, J. van den Heuvel, and S. McGuinness, Removable circuits in multigraphs, *J. Combin. Theory Ser. B* 71 (1997), 130–143.

[Gu93] B. S. Gubser, Planar graphs with no 6-wheel minor, *Discrete Math.* 120 (1993), 59–73.

[Ha43] D. W. Hall, A note on primitive skew curves, *Bull. Amer. Math. Soc.* 49 (1943), 435–437.

[Ha69] R. Halin, Untersuchungen über minimale n-fach zusammenhängende Graphen, *Math. Ann.* 182 (1969), 175–188.

[Ha69a] R. Halin, Zur Theorie der n-fach zusammenhängende Graphen, *Abh. Math. Sem. Univ. Hamburg* 33 (1969), 133–164.

[InWe81] T. Inukai and L. Weinberg, Whitney conectivity of matroids, *SIAM J. Alg. Discrete Methods* 2 (1981), 311–312.

[Ja80] B. Jackson, Removable cycles in 2-connected graphs of minimum degree at least four, *J. London Math. Soc. (2)* 21 (1980), 385–392.

[Ku30] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fund. Math.* 15 (1930), 271–283.

[La58] T. Lazarson, The representation problem for independence functions, *J. London Math. Soc.* 33 (1958), 21–25.

[Le64] A. Lehman, A solution to the Shannon switching game, *J. Soc. Indust. Appl. Math.* 12 (1964), 687–725.

[Le79] A. Lehman, On the width-length inequality, *Math. Programming* 17 (1979), 403–417.

[LeOx01] M. Lemos and J. Oxley, A sharp bound on the size of a connected matroid, *Trans. Amer. Math. Soc.* 353 (2001), 4039–4056.

[Ma72] W. Mader, Ecken vom Grad n in minimalen n-fach zusammenhängenden Graphen, *Arch. Math.* 23 (1972), 219–224.

[Ma74] W. Mader, Kreuzungfreie a, b-Wege in endliche Graphe, *Abh. Math. Sem. Univ. Hamburg* 42 (1974), 187–204.

[Ma79] W. Mader, Connectivity and edge-connectivity in finite graphs, pp. 66–95 in *Surveys in Combinatorics* (ed. B. Bollobás), Cambridge Univesity Press, 1979.

[Ma96] W. Mader, On vertices of degree n in minimally n-connected graphs and digraphs, in *Combinatorics, Paul Erdős is Eighty Vol. 2* (eds. D. Miklós, V. T. Sós, and T. Szőnyi), János Bolyai Math. Soc., Budapest, 1996, 423–449.

[Mc04?] S. McGuinness, Circuits through cocircuits in a graph with extensions to matroids, submitted.

[Mc04?a] S. McGuinness, Contractible bonds in graphs, submitted.

[Na61] C. St.J. A. Nash-Williams, Edge-disjoint spanning trees of finite graphs, *J. London Math. Soc.* 36 (1961), 445–450.

[Na66] C. St.J. A. Nash-Williams, An application of matroids to graph theory, in *Theory of Graphs* (Internat. Sympos., Rome), Dunod, Paris, 1966, 263–265.

[Ne82] S. Negami, A characterization of 3-connected graphs containing a given graph, *J. Combin. Theory Ser. B* 32 (1982), 69–74.

[NeRiUr99] V. Neumann-Lara, E. Rivera-Campo, and J. Urrutia, A note on covering the edges of a graph with bonds, *Discrete Math.* 197/198 (1999), 633–636.

[OpOxTh93] B. Oporowski, J. Oxley, and R. Thomas, Typical subgraphs of 3- and 4-connected graphs, *J. Combin. Theory Ser. B* 57 (1993), 239–257.

[Ox81] J. G. Oxley, On a matroid generalization of graph connectivity, *Math. Proc. Camb. Phil. Soc.* 90 (1981), 207–214.

[Ox81a] J. G. Oxley, On matroid connectivity, *Quart. J. Math. Oxford (2)* 32 (1981), 193–208.

[Ox81b] J. G. Oxley, On connectivity in matroids and graphs, *Trans. Amer. Math. Soc.* 265 (1981), 47–58.

[Ox89] J. G. Oxley, The regular matroids with no 5-wheel minor, *J. Combin. Theory Ser. B* 46 (1989), 292–305.

[Ox92] J. G. Oxley, *Matroid Theory*, Oxford University Press, 1992.

[Ox01] J. G. Oxley, On the interplay between graphs and matroids, *Surveys in Combinatorics, 2001* (ed. J. W. P. Hirschfeld), London Math. Soc. Lecture Notes 288, Cambridge University Press, 2001, 199–239.

[Ro71] G.-C. Rota, Combinatorial theory, old and new, in *Proc. Internat. Cong. Math.* (Nice, Sept. 1970), Gauthier-Villars, Paris, 1971, 229–233.

[RoSe04?] N. Robertson and P. D. Seymour, Graph minors. XX. Wagner's conjecture, *J. Combin. Theory Ser. B*, to appear.

[Sc86] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.

[Se79] P. D. Seymour, Matroid representation over $GF(3)$, *J. Combin. Theory Ser. B* 26 (1979), 159–173.

[Se80] P. D. Seymour, Decomposition of regular matroids, *J. Combin. Theory Ser. B* 28 (1980), 305–359.

[Si98] P. A. Sinclair, "Strong Snarks and the Removal of Edges from Circuits in Graphs", Ph. D. thesis, University of London, 1998.

[Tu58] W. T. Tutte, A homotopy theorem for matroids I, *Trans. Amer. Math. Soc.* 88 (1958), 144–160.

[Tu58a] W. T. Tutte, A homotopy theorem for matroids II, *Trans. Amer. Math. Soc.* 88 (1958), 161–174.

[Tu59] W. T. Tutte, Matroids and graphs, *Trans. Amer. Math. Soc.* **90** (1959), 527–552.

[Tu61] W. T. Tutte, On the problem of decomposing a graph into n connected factors, *J. London Math. Soc.* 36 (1961), 221–230.

[Tu65] W. T. Tutte, Lectures on matroids, *J. Res. Nat. Bur. Standards Sect. B* 69B (1965), 1–47.

[Tu66] W. T. Tutte, Connectivity in matroids, *Canad. J. Math.* 18 (1966), 1301–1324.

[Wa37] K. Wagner, Über eine Erweiterung eines Satzes von Kuratowski, *Deut. Math.* 2 (1937), 280–285.

[Wa60] K. Wagner, Bemerkungen zu Hadwigers Vermutung, *Math. Ann.* 141 (1960), 433–451.

[We99] D. Welsh, The Tutte polynomial, *Random Structures Algorithms* 15 (1999), 210–228.

[Wh33] H. Whitney, 2-isomorphic graphs, *Amer. J. Math.* 55 (1933), 245–254.

[Wh35] H. Whitney, On the abstract properties of linear dependence, *Amer. J. Math.* 57 (1935), 509–533.

[Wu98] H. Wu, On vertex-triads in 3-connected binary matroids, *Combin. Probab. Comput.* 7 (1998), 485–497.

[Wu97] P.-L. Wu, An upper bound on the number of edges of a 2-connected graph, *Combin. Probab. Comput.* 6 (1997), 107–113.

[Wu00] P.-L. Wu, Extremal graphs with prescribed circumference and cocircumference, *Discrete Math.* 223 (2000), 299–308.

GLOSSARY FOR CHAPTER 6

adjacency matrix – of a simple graph G: the 0-1 matrix A_G whose rows and columns correspond to the vertices of G, with an entry being 1 if and only if the corresponding row and column vertices are adjacent.

___, l^{th}**-order**: inductively defined with $A_0 = I$ as the identity matrix, $A_1 = A$ as the usual adjacency matrix, and A_l as the matrix with 1 in the (i, j) position if the corresponding vertices are at distance l from each other and 0 otherwise.

algebraic connectivity of a graph whose Laplacian has the eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$: the eigenvalue λ_2.

almost transitive automorphism group – for an infinite graph G: a group of automorphisms of G that acts with only finitely many orbits.

almost transitive graph: a graph G whose full automorphism group $\mathcal{A}ut(G)$ is almost transitive.

s**-arc**: a directed walk in a graph of length s in which consective edges are distinct.

arc-transitive graph: a graph G (undirected) whose automorphism group induces a transitive group action on the set of ordered edges of G.

asymmetric graph: a graph whose automorphism group is trivial.

automorphism group – of a graph G: the set of all automorphisms of a graph, made with the operation of composition into a group, usually denoted $\mathcal{A}ut(G)$.

automorphism – of a graph: an isomorphism of the graph onto itself.

___, **bounded** – for an infinite graph G: an automorphism such that there is a uniform bound on the distances between every vertex and its image.

basic figure: a vertex-disjoint union of elementary figures.

basis – of a matroid M: a maximal set containing no circuit of M.

bicycle – in a graph: a subgraph that is both a circ and a cut.

binary vector – representing a subset E' of edges in an undirected graph: a row vector, whose i^{th} is 1 if the i^{th} edge of the graph is in E' and is otherwise 0.

block – of objects under a permutation group action: a subset B of the set X of objects on which a permutation group $\mathcal{P}$ acts, such that for every permutation $\pi \in \mathcal{P}$, the image $\pi(B)$ either coincides with B or is disjoint from B.

___, **nontrivial**: a block other than $\emptyset$, a singleton set, or the entire set of objects on which a permutation group is acting.

bond – of a graph G: a minimal set of edges whose deletion from G increases the number of connected components.

branch$_1$ – at a vertex v of a tree T: a maximal subtree having v as a leaf.

branch$_2$ – of a spanning tree: an edge of the spanning tree.

cage: a smallest 3-valent graph with a given girth.

Catalan numbers: the sequence of numbers defined by the recursion
$C_0 = 1, \quad C_n \;=\; C_0C_{n-1} + C_1C_{n-2} + \cdots + C_{n-1}C_0 \quad \text{for } n \geq 1$

Cayley digraph – for a group $\mathcal{A}$ with *connection set* X of elements: the graph whose vertices are the elements of group $\mathcal{A}$ and such that, for each element $a \in \mathcal{A}$ and each connection $x \in X$, there is a directed edge from vertex a to vertex ax. Commonly denoted $C(\mathcal{A}, X)$.

Cayley graph$_1$: any graph isomorphic to the underlying undirected graph of a Cayley digraph.

Cayley graph$_2$: a Cayley digraph.

Cayley graph$_3$: where the connection set X is stipulated to be a generating set.

Cayley graph$_4$: where the connection set X is stipulated to be balanced, which means that $x \in X$ if and only if $x^{-1} \in X$.

characteristic polynomial – of a graph: the determinant $\det(xI - A)$ of its adjacency matrix.

chord – of a spanning tree: an edge of the cotree.

CI-graph: a Cayley graph $C(\mathcal{A}, \mathcal{X})$ such that whenever $C(\mathcal{A}, \mathcal{X}) \cong \mathcal{C}(\mathcal{A}, \mathcal{X}')$, there exists an automorphism $\alpha \in \mathcal{A}\mathrm{ut}(\mathcal{A})$ such that $X' = \alpha(X)$.

CI-group: a group $\mathcal{A}$ such that every Cayley graph on $\mathcal{A}$ is a CI-graph.

circ – in a graph: a circuit or union of edge-disjoint circuits of the graph.

circuit – in a graph: a subgraph isomorphic to any of the cycle graphs C_n.

___, **directed** – in a directed graph: a circuit in which all the edges are oriented in the same direction.

___, **removable** – of a k-connected matroid M: for k in $\{2, 3\}$, a circuit of M such that the deletion of C from M is k-connected.

circuit matrix – of a graph (directed or undirected): the matrix in which each row is a circuit vector, with one row for each circ in the graph.

circuit space – of a directed graph G: the set of all circuit vectors and their linear combinations over the real field, and is denoted by $\hat{C}(G)$.

circuit subspace – of an undirected graph G: the set of all circs of the graph; denoted by $\hat{C}(G)$.

circuit vector – of an undirected graph: the binary m-vector representing a circ of the graph.

circuit vector – in a directed graph: an m-vector representing a circ of the graph; the signs of the elements in the vector depend on the orientation assigned to each of the circuits in the circ.

circulant graph: an undirected Cayley graph on the cyclic group $\mathbb{Z}_n$.

closed under duality – of a class $\mathcal{M}$ of matroids: the dual of every member of $\mathcal{M}$ is also in $\mathcal{M}$.

closure – of a set X in a matroid M: the maximal subset of $E(M)$ that contains X and has the same rank as X.

coalescence – of two (disjoint) graphs G_1 and G_2, with distinguished vertices v_1 and v_1: the graph formed from their union by identifying the vertices v_1 and v_1; also called *amalgamation at a vertex.*

cocircuit – of a matroid M: a circuit of the dual matroid of M.

cocktail party graph CP(n): the regular graph of degree $2n - 2$ with $2n$ vertices; another name for the *n-dimensional octahedal graph.*

complement of a subgraph G' – in a graph G: the graph $G'' = (V(G), E(G) - E(G'))$.

component – of a graph: a maximal connected subgraph.

connected graph: a graph in which there is a path between every pair of vertices.

connection set: see *Cayley graph*.

contraction – of a set T from a matroid M: the matroid M/T on $E(M) - T$ whose circuits are the minimal nonempty sets in $\{C - T : C \in \mathcal{C}(M)\}$.

cospanning tree – of a graph G with respect to a spanning tree T: the complement of T in G; this is generally called the *cotree* of T.

cotree – of a spanning tree T in a graph G: the *complement* of T in G.

cut matrix – of a graph (directed or undirected): the matrix in which each row is a cut vector and its number of rows is equal to the number of cuts in the graph.

cut vector – in a directed graph: the m-vector representing a cut of the graph; the signs of the elements in the vector depend on the cut orientation.

cut vector – in an undirected graph: a binary m-vector representing a cut of the graph.

cut $\langle V_1, V_2\rangle$ – in a graph $G = (V, E)$: the set of edges with one end vertex in V_1 and the other in $V_2 = V - V_1$.

cutset$_1$ – in a connected graph G: a set of edges whose removal increases the number of components.

cutset$_2$ – in a connected graph G: a cut whose removal results in a graph with exactly two components.

___, **directed** – of a directed graph: a cut in which all the edges are oriented in the same direction.

cutset space or **cutset subspace$_1$** – of a directed graph G: the set of all cut vectors of G and their linear combinations over $GF(2)$.

cutset space or **cutset subspace$_2$** – of a directed graph G: the set of all cut vectors of G and their linear combinations over the reals.

cycle matroid – of a graph G: the matroid on the edge-set of G whose circuits are the edge-sets of the cycles of G.

degree – of a vertex: the number of edges incident on that vertex.

deletion – of a set T from a matroid M: the matroid $M\backslash T$ on $E(M) - T$ whose circuits are the circuits of M contained in $E(M) - T$.

dependent set – of a matroid M: a set containing a circuit of M.

digraph, labeled: a digraph with labels, typically $v_1, v_2, \ldots, v_n$, assigned to the vertices. Two labeled digraphs with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

direct sum – of matroids M_1 and M_2 on disjoint sets: the matroid on $E(M_1) \cup E(M_2)$ whose circuits consist of every set that is a circuit of M_1 or of M_2.

distance-regular graph – with parameters $p^k_{i,j}$, $0 \le i, j, k \le d$: a graph such that for any pair of vertices whose distance is k, the number of vertices at distance i from the first and distance j from the second is $p^k_{i,j}$.

divisor of a graph G – with respect to any *product operation* $\natural$: either of the coordinate factors A or B, when graph G is expressed as a graph product $A\natural B$.

___, **proper**: a *divisor* of a graph other than itself and the trivial graph K_1.

doubly transitive group: a permutation group that acts transitively on ordered pairs of elements.

dual$_1$ of a graph G (*Poincare dual*): a graph obtained from a *cellular imbedding* of G on a surface, by drawing a dual vertex in each region, and then drawing a dual edge thru each edge of G (see §7.6), so as to join dual vertices.

dual$_2$ of a graph G (*Whitney dual*): see Definition 51 of §6.4.

dual M^* – of a matroid M: the matroid on $E(M)$ whose set of bases consists of the set of complements of bases of M.

edge-automorphism: an edge-isomorphism from a graph to itself.

edge-group: the permutation group on the edge-set of a graph consisting of the edge-automorphisms.

edge-isomorphism: a bijection from the edge-set of one graph to the edge-set of another graph that maps every pair of adjacent edges to a pair of adjacent edges.

edge-transitive graph: a graph whose automorphism group induces a transitive group action on the edge set of the graph.

eigenvalues – of a graph: the roots of its characteristic polynomial.

eigenvalues-diameter (lower) bound – for the number of eigenvaues of a graph G: $\text{diam}(G) + 1$.

elementary figure: an subgraph isomorphic either to a K_2 or to a cycle graph C_r.

empty graph: a graph with no edges.

end – of an infinite graph: an equivalence class of rays such that no two subrays can be separated by a finite subgraph.

endomorphism – of a graph: a homomorphism of the graph to itself.

even graph: a graph in which the degree of every vertex is even. (Such graphs are more commonly called *eulerian graphs.*)

excluded minor – of a minor-closed class of matroids: a matroid that is not in the class but has all its proper minors in the class.

factor: see *divisor.*

flat – of a matroid M: a maximal subset of $E(M)$ of a fixed rank.

fundamental circuit matrix – of a connected graph with respect to a spanning tree: the $(m - n + 1)$-rowed submatrix of the circuit matrix in which each row is a fundamental circuit vector with respect to the spanning tree, and will be denoted by B_f; in a directed graph, the orientation of the fundamental circuit is chosen to agree with the orientation of the chord defining the fundamental circuit.

fundamental circuit – with respect to a chord c and a spanning tree T of a graph: the unique circuit produced by adding chord c to spanning tree T.

fundamental cutset matrix – of a connected graph with respect to a spanning tree: the $(n - 1)$-rowed submatrix of the cut matrix in which each row is a fundamental cutset vector with respect to the spanning tree; in a directed graph, the orientation of a fundamental cutset is chosen to agree with the orientation of the edge defining the fundamental cutset.

fundamental cutset – of a graph with respect to an edge: the unique cutset $< V_1, V_2 >$, where V_1 and V_2 are the sets of vertices of the two trees that result when the edge is removed from the spanning tree.

generalized dicyclic group: an abstract group generated by an abelian, but not elementary abelian, subgroup $\mathcal{A}$ of index 2 and an element b of order 4 such that conjugation by b inverts every element of $\mathcal{A}$.

geodesic: a geodetic double ray.

geodetic: said of a path, ray, or double ray that contains a shortest path joining any two of its vertices.

graph product$_1$ – of two graphs: the *cartesian product.*

graph product$_2$ – of two graphs: a graph that results from applying any graph *product operation* $\natural$.

graph, labeled: a graph with labels, typically $v_1, v_2, \ldots, v_n$, assigned to the vertices. Two labeled graphs with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

graphical regular representation of a group $\mathcal{G}$: a graph whose automorphism group is isomorphic to $\mathcal{G}$ and acts regularly on the vertex set of the graph.

growth – of an infinite graph G: $\liminf_{k\to\infty}[d(k)/a^k]$, where $d(k)$ is the number of vertices of G at distance k from a fixed vertex and $a > 1$ is a real number.

GRR: graphical regular representation.

half-transitive: vertex-transitive and edge-transitive, but not arc-transitive.

Hamilton decomposition – of a regular graph: a partition of the edge-set into Hamilton cycles (when the degree is even) or into Hamilton cycles and a 1-factor (when the degree is odd).

Hamilton-connected graph: a graph such that for any two vertices u, v, there is a Hamilton path whose terminal vertices are u and v.

Hamilton-laceable graph: a bipartite graph with parts A and B such that for any $u \in A$ and $v \in B$, there is a Hamilton path whose terminal vertices are u and v.

Hamming graph $H(d, n)$: the graph whose vertices are the d-tuples $(x_1, \ldots, x_d)$ with $1 \le x_i \le n$; two vertices are joined if, as d-tuples, they agree in all but one coordinate. (The distance between two vertices is then the number of coordinates in which, as d-tuples, they differ.)

hereditary collection – of sets: a collection $\mathcal{A}$ of sets such that every subset of a member of $\mathcal{A}$ is also in $\mathcal{A}$.

Hoffman polynomial – for an r-regular, connected graph: the polynomial $h(x) = n\prod\frac{(x-\lambda_i)}{(r-\lambda_i)}$, the product being taken over all distinct eigenvalues not equal to r.

homomorphism of general graphs G and H: a pair of mappings $f : V_G \to V_H$ and $f : E_G \to E_H$ such that the endpoint-set of each edge $e \in E_G$ is mapped onto the endpoint set of the image edge $f(e) \in E_H$.

homomorphism of simple graphs G and H: a mapping $f : V_G \to V_H$ such that whenever the vertices u and v are adjacent in G, the vertices $f(u)$ and $f(v)$ are adjacent in H.

hyperplane – of a matroid M: a maximal subset of $E(M)$ that does not contain a basis of M.

incidence matrix$_1$ – of a graph: a matrix whose rows correspond to the vertices and whose columns correspond to the edges; the ij entry is 2 if edge j is a self-loop and vertex i is its endpoint, 1 if edge j is a proper edge and vertex i is an endpoint, and is 0 otherwise.

incidence matrix$_2$ – of a graph: the n-rowed submatrix of the cut matrix in which each row is an incidence vector.

incidence set – of a vertex: the set of edges incident on that vertex.

incidence vector – for a directed graph: the cut vector representing the set of edges incident on a vertex of the graph, with the orientation of the cut chosen to be away from the vertex.

incidence vector – for an undirected graph: the binary cut vector representing the set of edges incident on a vertex of the graph.

independent set – of a matroid M: a set containing no circuit of M.

induced subgraph on an edge subset $E' \subset E(G)$: the subgraph of G with edge-set E' and vertex-set consisting of the endpoints of the edges in E'.

induced subgraph on a vertex subset $V' \subset V(G)$: the graph with vertex set V' and edge-set consisting of those edges whose endpoints are in V'.

infinite connectivity of a graph G – denoted $\kappa_\infty(G)$: the cardinality of a smallest set of vertices whose deletion leaves a graph with at least two infinite components.

isolated vertex: a vertex with degree zero.

isomorphic factorization – of a graph G: a partition of the edge set of G so that the subgraphs induced by the edges in each part are mutually isomorphic.

isomorphic matroids: matroids M_1 and M_2 for which there is a 1-1 function ϕ from $E(M_1)$ onto $E(M_2)$ such that C is a circuit of M_1 if and only if $\phi(C)$ is a circuit of M_2.

isomorphic permutation groups: a pair of isomorphic groups whose actions on their respective sets are the same, up to a bijection from one object set to the other.

isomorphism of general graphs G and H: a pair of bijections $f : V_G \to V_H$ and $f : E_G \to E_H$ such that the endpoint-set of each edge $e \in E_G$ is mapped onto the endpoint set of the image edge $f(e) \in E_H$.

isomorphism of simple graphs G and H: a bijection $f : V_G \to V_H$ such that vertices $f(u)$ and $f(v)$ are adjacent in H if and only if vertices u and v are adjacent in G.

Johnson graph $J(d,n)$: the graph whose vertices are the $\binom{n}{d}$ subsets of $\{1,\ldots,n\}$; two vertices are joined if, as subsets, their intersection has cardinality $d-1$. (The distance bewteen two vertices is k if, as subsets, they have an intersection of cardinality $d-k$.)

Kuratowski graph: either of the two graphs in Figure 6.4.11, which characterize nonplanarity.

Laplacian matrix: a square matrix whose rows and columns correspond to the vertices of a graph, such that a diagonal entry is the degree of the corresponding vertex; an off-diagonal entry is -1 if the corresponding vertices are adjacent and 0 otherwise.

line graph of a graph G, denoted by $L(G)$: a graph whose vertex-set is the edge-set of G, with two vertices in $L(G)$ adjacent if, as edges of G, they have an endpoint in common.

___, **generalized** $L(G; a_1, \ldots, a_n)$ – for a graph G with n vertices and nonnegative integers $a_1, \ldots, a_n$: the graph formed by taking disjoint copies of the line graph $L(G)$ and cocktail party graphs $\mathrm{CP}(a_1), \ldots, \mathrm{CP}(a_n)$; if a vertex in $L(G)$ corresponds to the edge joining v_i to v_j in G, it is joined to all vertices in $\mathrm{CP}(a_i)$ and $\mathrm{CP}(a_j)$.

loop – of a matroid M: an element e of $E(M)$ for which $\{e\}$ is a circuit.

matroid M: a finite set $E(M)$, the ground set of M, and a collection $\mathcal{C}(M)$ of nonempty incomparable subsets of $E(M)$ called the circuits of M such that if C_1 and C_2 are distinct members of $\mathcal{C}(M)$ and $e \in C_1 \cap C_2$, then there is a member C_3 of $\mathcal{C}(M)$ such that $C_3 \subseteq (C_1 \cup C_2) - \{e\}$.

___, **binary**: a matroid that is isomorphic to the vector matroid of a matrix over the 2-element field $GF(2)$.

___, **bond** – of a graph G: the matroid on the edge-set of G whose circuits are the bonds of G.

___, **cographic**: a matroid that is isomorphic to the bond matroid of some graph.

___, **2-connected**: a matroid in which, for every two distinct elements, there is a circuit containing both.

___, **3-connected**: a 2-connected matroid that cannot be written as a 2-sum.

___, **graphic**: a matroid that is isomorphic to the cycle matroid of some graph.

___, **planar**: a matroid that is isomorphic to the cycle matroid of a planar graph.

___, **regular**: a matroid that is representable over all fields.

___, **representable** – over a field $\mathbb{F}$: a matroid that is isomorphic to the vector matroid of some matrix over $\mathbb{F}$.

___, **simple**: a matroid in which all circuits have at least three elements.

___, **uniform** $U_{m,n}$: for $0 \le m \le n$, the matroid on $\{1, 2, \ldots, n\}$ in which the circuits consist of all $(m+1)$-element subsets.

maximally distant trees: two spanning trees T_1 and T_2 such that $d(T_1, T_2) \geq d(T_i, T_j)$, for every pair of spanning trees T_i and T_j.

metric ray (double ray): a *ray* (*double ray*) with positive *straightness*.

metric type: describes a *ray* in an infinite graph that is an α-*essential* ray for some automorphism α.

minimally k-connected graph: a k-connected graph for which no deletion of an edge remains k-connected.

minimally k-connected matroid: for k in $\{2, 3\}$, a k-connected matroid for which no single-element deletion is k-connected.

minimum polynomial of a graph G: the monic polynomial $q(x)$ of smallest degree, such that $q(A_G) = 0$.

minor – of a graph G: a graph that can be obtained from G by a sequence of edge deletions, edge contractions, and deletions of isolated vertices.

minor – of a matroid M: a matroid that can be obtained from M by a sequence of deletions and contractions.

___, **proper** – of a matroid M: a minor of M that is not equal to M.

minor-closed – class of matroids: one in which every minor of a member of the class is also in the class.

nonseparating cocircuit – of a 2-connected matroid M: a cocircuit whose deletion from M remains 2-connected.

null graph: a graph with no vertices and hence no edges.

nullity – of a graph G having n vertices, m edges and p components: nullity is equal to $m - n + p$ and is denoted $\mu(G)$.

orientation of a cut $\langle V_1, V_2\rangle$ – in a directed graph: the direction, either from V_1 to V_2 or from V_2 to V_1, that we choose for the cut.

orientation of a circuit – in a directed graph: the direction we choose to traverse the circuit.

orthogonal complements – of a vector space: two subspaces whose intersection is the zero vector.

orthogonal subspaces of a vector space: subspaces such that the inner product of every vector in one subspace with every vector in the other subspace is equal to zero.

painting – of a graph: a partitioning of the edges into three sets R (red), Y (yellow), and B (blue), and the distinguishing of one edge in the set Y.

Paley graph: a Cayley graph formed on the additive group of a finite field $GF(q)$, $q \equiv 1 (\text{mod } 4)$, where the connection set is the set of quadratic residues in $GF(q)$.

parallel elements e and f of a matroid M: elements such that $\{e, f\}$ is a circuit of M.

parameter matrix, l^{th}-order: the matrix P_k with the distance-regularity parameter $p^i_{j,k}$ in the (i, j) entry.

permutation group, doubly transitive: a permutation group that acts transitively on ordered pairs of objects.

permutation group, primitive: a transitive permutation group whose only blocks are trivial.

permutation group, regular: a permutation group that is both transitive and semiregular.

permutation group, regular: a transitive permutation group P acting on a finite set X, satisfying any one of the following three equivalent conditions:

- the only element of $\mathcal{P}$ fixing an element of X is the identity permutation;
- $|\mathcal{G}|$=$|X|$;
- for any $x_1, x_2 \in X$, there is a unique element $\pi \in \mathcal{P}$ satisfying $\pi(x_1) = x_2$.

permutation group: a nonempty set $\mathcal{P}$ of permutations (on the same set X of objects), such that $\mathcal{P}$ is closed under composition and inversion.

___, **semiregular**: a permutation group all of whose vertex-stabilizers are trivial.

___, **transitive**: a permutation group such that for any two objects of the set on which it acts, some permutation maps one object onto the other.

prime graph – under a given *product operation*: a graph having no proper *divisor*.

primitive graph: a graph whose automorphism group acts as a primitive permutation group on the vertex-set.

primitive group: a transitive permutation group that has no nontrivial blocks.

principal subgraphs G_1 and G_2 – of a graph G: see Definition 46 of §6.4.

product operation – on two graphs G and H: any operation $\natural$ such that the vertex-set $G\natural H$ is the cartesian product of V_G and V_H, and such that the edge-set is determined exclusively by the adjacency relations in G and Hs.

rank – of a graph G having n vertices and p components: the number of edges in the complement of a spanning forest; the rank is equal to $n - p$ and is denoted by $\rho(G)$; usually called the *cycle rank.*

rank – of a set A in a matroid: the cardinality of a maximal independent subset of A. The rank of a matroid M is the cardinality of a maximal independent subset of $E(M)$.

ray – in an infinite graph: a one-way infinite path.

___, **α-essential** – in an infinite graph: a ray that is mapped onto one of its subrays by a positive power of the automorphism α.

___, **double** – in an infinite graph: a two-way infinite path.

reduced incidence matrix – of a graph: the submatrix of the incidence matrix containing any $(n-1)$ incidence vectors.

regular action – of a permutation group: see *permutation group, regular.*

regular matrix: see Definitions 49 and 50 of §6.4.

s-regular graph: a graph that contains at least one s-arc and whose automorphism group acts regularly on its set of s-arcs.

relatively prime graphs: graphs having no common proper divisor.

removable cycle – of a k-connected graph G: a cycle of G such that the deletion of the edges of C from G leaves a k-connected graph.

ring sum of two sets E_1 and E_2: the set consisting of elements that belong to E_1 or to E_2, but not to both E_1 and E_2; denoted by $E_1 \oplus E_2$.

ring sum of two vectors $(x_1, x_2, x_3, \ldots, x_i, \ldots, x_m)$ and $(y_1, y_2, y_3, \ldots, y_i, \ldots, y_m)$: the vector $Z = (z_1, z_2, z_3, \ldots, z_i, \ldots, z_m)$, where $z_i = x_i \otimes y_i$ and $\otimes$ is the logical exclusive-or operation ($1 \otimes 0 = 1$, $0 \otimes 1 = 1$, $0 \otimes 0 = 0$, and $1 \otimes 1 = 0$).

semisymmetric graph: an edge-transitive graph with constant valence (i.e., a regular graph) that is not vertex-transitive.

spanning forest – of a graph G having p components: a collection of p spanning trees, one for each component of G.

spanning set – of a matroid M: a subset of $E(M)$ containing a basis of $E(M)$.

spanning tree – of a connected graph: a tree that contains all the vertices of the graph.

spectrum of a graph: the multiset of eigenvalues; for a graph with n vertices, there are n eigenvalues.

stabilizer of a vertex u of a graph G: the subgroup of $\mathcal{A}ut$ consisting of the permutations that fix vertex u.

straightness of a ray or double ray D: the number $\liminf_{d_D(u,v)\to\infty} d(u,v)/d_D(u,v)$, where u, v are vertices of D.

strip: a connected graph G that admits a connected subgraph H and an automorphism α such that ∂H and $H - \alpha(H)$ are finite and $\alpha(H \cup \partial H) \subseteq H$.

strongly regular graph – with parameters (n, r, λ, μ): an r-regular n-vertex graph such that any pair of adjacent vertices is mutually adjacent to λ other vertices, and such that any pair of nonadjacent vertices is mutually adjacent to μ other vertices.

2-sum of matroids: for 2-connected matroids M_1 and M_2 on disjoint sets each having at least three elements, let p_i be an element of M_i; the 2-sum with respect to p_1 and p_2 is the matroid on $(E(M_1) - \{p_1\}) \cup (E(M_2) - \{p_2\})$ whose circuits are the circuits of M_1 avoiding p_1, the circuits of M_2 avoiding p_2, and all sets of the form $(C_1 - \{p_1\}) \cup (C_2 - \{p_2\})$ where C_i is a circuit of M_i containing p_i.

3-sum of matroids: for binary matroids M_1 and M_2 each having at least seven elements such that $E(M_1) \cap E(M_2)$ is a 3-element circuit T of M_1 and M_2 that does not contain a cocircuit of either matroid; the 3-sum is the matroid on $(E(M_1) \cap E(M_2)) - T$ whose flats are those sets $F - T$ such that $F \cap E(M_i)$ is a flat of M_i for each i.

symmetric difference – of two sets E_1 and E_2: the set consisting of only elements that belong to E_1 or to E_2, but not to both E_1 and E_2; denoted by $E_1 \oplus E_2$.

symmetric group S_n: the group of all permutations acting on the set $\{1, 2, \ldots, n\}$.

system of imprimitivity: collection of images of a nontrivial *block* under the action of a transitive permutation group.

ternary matroid: a matroid that is isomorphic to the vector matroid of a matrix over the 3-element field $GF(3)$.

torsion subgroup: a subgroup of an infinite group, all of whose elements have finite order.

totally unimodular matrix: a matrix over the real numbers for which the determinant of every square submatrix is in $\{0, 1, -1\}$.

tournament: a digraph in which, for each pair u, v of distinct vertices, either there exists an arc from u to v or an arc from v to u but not both.

___, **strong**: short for *strongly connected tournament.*

___, **strongly connected**: a tournament such that for each pair u, v of vertices, there exist directed paths from u to v and from v to u.

transitive action – of a permutation group: see *permutation group, transitive.*

s-**transitive graph**: a graph that contains at least one s-arc and whose automorpism group acts transitively on its set of s-arcs.

translation: an endomorphism of a graph that fixes no finite nonempty subset of the vertex set.

tree labeled is a tree in which labels, typically $v_1, v_2, \ldots, v_n$, have been assigned to the vertices. Two labeled trees with the same set of labels are considered the same only if there is an isomorphism from one to the other that preserves the labels.

tree – in a graph: a connected subgraph of the graph containing no circuits.

___, **1-4**: a tree in which each vertex has degree 1 or 4.

___, **1-rooted 1-4**: a 1-4 tree rooted at a vertex of degree 1.

___, **binary**: a root vertex and at most two principal subtrees that are themselves binary trees. Each principal subtree must be specified as either the left subtree or the right subtree.

___, **homeomorphically reduced**: a tree with no vertices of degree 2.

___, **left-right**: a binary tree in which each vertex has either 0 or 2 children.

___, **ordered**: a root vertex and a sequence $t_1, t_2, \ldots,\ t_m$ of $m \geq 0$ principal subtrees that are themselves ordered trees. The root vertex of an ordered tree is joined by an edge to the root of each principal subtree.

___, **reduced**: short for *tree, homeomorphically reduced.*

___, **rooted**: a tree in which one vertex, the root, is distinguished. Two rooted trees are considered the same only if there is an isomorphism from one to the other that maps the root of the first to the root of the second.

trivial graph: a graph with a single vertex and no edge.

unimodular matrix: a matrix of real numbers, the determinant of every square submatrix of which is equal to 1, -1, or 0.

union – of matroids $M_1, M_2, \ldots, M_n$ on a common set E: the matroid on E whose independent sets consist of all sets of the form $I_1 \cup I_2 \cup \ldots \cup I_n$ where I_j is an independent set of M_j for all j.

vector matroid of a matrix: the matroid on the set of column labels of the matrix whose circuits are the minimal linearly dependent multisets of columns.

vector space of a graph G: the set of all subsets of edges of G; also, the set of all vectors representing the subsets of edges of G; more commonly called the *edge space* of G.

vertex-edge incidence matrix – of a graph: see *incidence matrix.*

vertex-transitive graph: a graph whose automorphism group acts transitively on its vertex set.

wheel$_1$ W_n: for $n \geq 2$, the graph with $n+1$ vertices that is obtained by joining each vertex of an n-cycle, called the "rim", to one newly added vertex called the "hub"by an edge, called a "spoke".

wheel$_2$ W_n: a graph with n vertices, of which $n-1$ form a cycle (the *rim*), with the remaining vertex (the *hub*) adjacent to all the rim vertices.

whirl, $\mathcal{W}^r$: for $r \geq 2$, the matroid on the set of edges of $\mathcal{W}_r$ whose circuits are all the cycles of $\mathcal{W}_r$ except the rim along with all sets consisting of the rim plus a single spoke.

wreath product – of permutation groups G and H acting on sets S and T, resp.: a permutation group on $S \times T$ of which each element π satisfies $\pi(s,t) = (\alpha(s), \beta_{\alpha(s)}(t))$, where $\alpha \in G$ and $\beta_t \in H$ for each $t \in T$.

Chapter 7

TOPOLOGICAL GRAPH THEORY

7.1 GRAPHS ON SURFACES

Tomaž Pisanski, University of Ljubljana, Slovenia
Primož Potočnik, University of Ljubljana, Slovenia

7.2 MINIMUM AND MAXIMUM IMBEDDINGS

Jianer Chen, Texas A&M University

7.3 GENUS DISTRIBUTIONS

Jonathan L. Gross, Columbia University

7.4 VOLTAGE GRAPHS

Jonathan L. Gross, Columbia University

7.5 THE GENUS OF A GROUP

Thomas W. Tucker, Colgate University

7.6 MAPS

Andrew Vince, University of Florida

7.7 REPRESENTATIVITY

Dan Archdeacon, University of Vermont

7.8 TRIANGULATIONS

Seiya Negami, Yokohama National University, Japan

7.9 GRAPHS AND FINITE GEOMETRIES

Arthur T. White, Western Michigan University

GLOSSARY

7.1 GRAPHS ON SURFACES

Tomaž Pisanski, University of Ljubljana, Slovenia
Primož Potočnik, University of Ljubljana, Slovenia

7.1.1 Surfaces
7.1.2 Polygonal Complexes
7.1.3 Imbeddings
7.1.4 Combinatorial Descriptions of Maps
References

Introduction

The need to imbed (draw) finite graphs on surfaces arises in various aspects of mathematics and science. Often the simplest surface in which such a graph can be imbedded is sought. Some generalizations of surfaces are briefly considered.

7.1.1 Surfaces

2-Manifolds and 2-Pseudomanifolds

DEFINITIONS

D1: The ***open unit disk***, the ***closed unit disk*** and the ***unit half-disk*** are the respective subsets
$\{(x, y) \mid x^2 + y^2 < 1\}$, $\{(x, y) \mid x^2 + y^2 \le 1\}$ and $\{(x, y) \mid x \ge 0, x^2 + y^2 < 1\}$
of the Euclidean plane, together with the inherited Euclidean topology.

D2: An ***open disk***, a ***closed disk*** and a ***half disk*** are any topological spaces homeomorphic, respectively, to the open unit disk, the closed unit disk, or to the unit half-disk. A ***disk*** usually means a closed disk.

D3: A ***pinched open disk*** is a topological space obtained from k copies of open disks by identifying their respective centers to a single vertex, as shown in Figure 7.1.1.

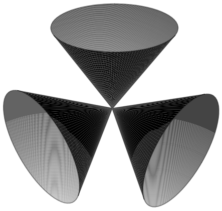


Figure 7.1.1 Three disks pinched together.

D4: A ***2-manifold*** is a topological space in which each point has a neighborhood that is homeomorphic either to an open disk or to a half-disk.

D5: The ***boundary*** of a 2-manifold M is the subspace of those points in M that do not have neighborhoods homeomorphic to open disks.

D6: A ***surface*** is a 2-manifold, often taken in context to be connected.

D7: A ***closed surface*** is a compact surface without boundary.

D8: If we relax the definition of a 2-manifold to allow the neighborhoods to be homeomorphic not only to open disks or half-disks but also to pinched open disks, then the resulting topological space is called a ***2-pseudomanifold***.

D9: A ***pseudosurface*** is a 2-pseudomanifold (usually taken to be connected). It may be obtained from a 2-manifold by successively identifying finitely many pairs of vertices.

FACTS

F1: The boundary components of a compact surface are closed curves. That is, each boundary component is homeomorphic to the unit circle.

F2: Every pseudosurface can be obtained from some 2-manifold by iteratively identifying finitely many pairs of points.

EXAMPLES

E1: The Euclidean plane is a non-compact surface.

E2: The closed disk is a compact surface with a nonempty boundary.

E3: The half-disk is a non-compact surface with a nonempty boundary.

E4: The ***pinched torus*** is a pseudosurface obtained from a sphere by identifying two of its points, as at the left of Figure 7.1.2.

E5: A ***jellyfish pseudosurface*** (also called the ***spindle pseudosurface***) is obtained from two spheres by pairwise identifying some number n of points on one sphere with n points on the other, as shown at the right of Figure 7.1.2.

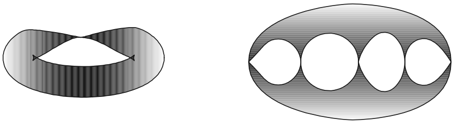


Figure 7.1.2 **The pinched torus and a jellyfish pseudosurface.**

Some Standard Surfaces

DEFINITIONS

D10: A ***sphere*** (usually denoted by S_0) is any surface homeomorphic to the ***unit sphere*** $\{(x, y, z) \mid x^2 + y^2 + z^2 = 1\}$.

D11: A ***cylinder*** (or ***annulus***) is any surface which is homeomorphic to the ***unit cylinder*** $\{(x, y, z) \mid x^2 + y^2 = 1, -1 \leq z \leq 1\}$.

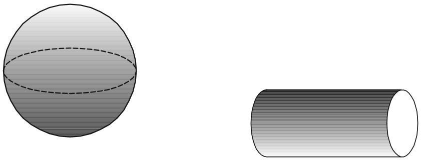


Figure 7.1.3 **The sphere S_0 and the cylinder.**

D12: A ***projective plane*** (usually denoted by N_1) is a closed surface homeomorphic to the surface obtained from the closed unit disk by identifying pairs of boundary points that are diametrically opposite relative to the center of the disk.

D13: A ***Möbius band*** (or ***Möbius strip***) is any surface that is homeomorphic to the surface obtained from a unit square $\{(x, y)| - 1 \leq x \leq 1, -1 \leq y \leq 1\}$ by pasting the vertical sides together with the matching $(-1, y) \rightarrow (1, -y)$.

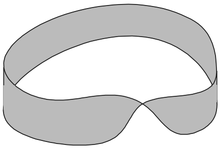


Figure 7.1.4 **A Möbius band is a non-orientable surface with boundary.**

D14: A ***torus*** (usually denoted by S_1) is a closed surface homeomorphic to the subset of the Euclidean three-dimensional space obtained by rotating a circle $\{(x, y, z) \mid (x - 2)^2 + y^2 = 1, z = 0\}$ around the y-axis.

D15: The ***Klein bottle*** (usually denoted by N_2) is a closed surface homeomorphic to the surface obtained from the unit cylinder $\{(x, y, z) \mid x^2 + y^2 = 1, -1 \leq z \leq 1\}$ by identifying the pairs of point $\{(x, y, -1), (x, -y, 1)\}$ on the two boundary components.

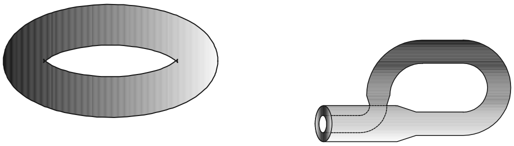


Figure 7.1.5 **The torus S_1 and the Klein bottle N_2.**

FACTS

F3: An annulus can be obtained by excising the interior of a disk from a sphere.

F4: A Möbius band can be obtained by excising the interior of a disk from a projective plane.

F5: The torus can be obtained by identifying the pairs of points $\{(x, y, -1), (x, y, 1)\}$ on the two boundary components of the unit cylinder

$$\{(x, y, z) \mid x^2 + y^2 = 1, -1 \leq z \leq 1\}.$$

EXAMPLES

E6: The sphere, the torus, and the Klein bottle are closed surfaces.

E7: A closed disk is a compact surface with one boundary component, while an annulus is a compact surface with two boundary components.

E8: A Möbius strip is a compact surface with one boundary component.

Surface Operations and Classification

DEFINITIONS

D16: The ***connected sum*** $S\#S'$ of two surfaces S and S' is obtained by excising the interior of a closed disk in each surface and then gluing the corresponding boundary curves.

D17: ***Adding an orientable handle*** to a surface S means forming the connected sum $S\#S_1$.

D18: The ***orientable surface with g handles*** or the ***g-torus*** is the connected sum of g copies of a torus. It is denoted by S_g.

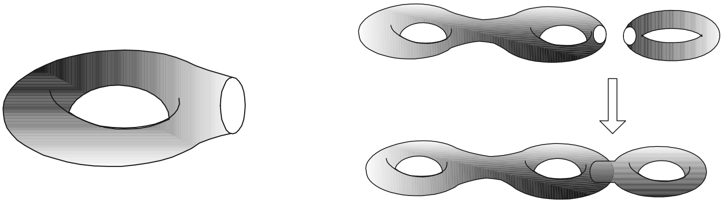


Figure 7.1.6 **Adding a handle to S_2 to obtain S_3.**

D19: ***Adding a crosscap*** to a surface S means forming the connected sum $S\#N_1$.

D20: The ***non-orientable surface with k crosscaps***, denoted by N_k, is the connected sum of k copies of the projective plane N_1.

D21: A 2-manifold is ***non-orientable*** if it contains a subspace that is homeomorphic to the Möbius band. Otherwise it is ***orientable***.

D22: The ***genus*** $\gamma(S)$ of a closed orientable surface homeomorphic to S_g is the number g of handles.

D23: The ***crosscap number*** or ***non-orientable genus*** $\tilde{\gamma}(S)$ of a closed nonorientable surface homeomorphic to N_k is k.

D24: The ***Euler characteristic*** $\chi(S)$ of a closed surface is defined by these formulas:

$$\chi(S_g) = 2 - 2g \qquad \text{and} \qquad \chi(N_k) = 2 - k.$$

D25: A ***surface with k holes*** is obtained by removing the interiors of k disjoint disks from a closed surface.

D26: A simple closed curve on a surface is ***separating*** if its excision splits the surface into two components. Otherwise it is ***non-separating***.

D27: A separating curve C on a surface S is ***contractible*** if the closure of one of the components of $S - C$ is a disk.

D28: A curve C on a surface S is ***essential*** if it is not contractible.

D29: A simple closed curve on a surface is ***orientation reversing*** if its regular neighborhood is a Möbius band.

FACTS

F6: An equivalent way to add a handle to an orientable surface is to remove the interiors of two disjoint disks and then to match the boundary components of a cylinder to the resulting boundary components, so as to preserve the orientation.

F7: The connected sum is well defined (up to homeomorphism of topological spaces) and is commutative and associative.

F8: A 2-manifold is orientable if and only if it is homeomorphic to a two-sided subspace of Euclidean 3-space.

F9: A closed 2-manifold is orientable if and only if it is homeomorphic to a surface in Euclidean 3-space.

F10: **Classification of Closed Surfaces**: Each closed surface is homeomorphic to one and only one of the following surfaces: $S_g, g \geq 0$, or $N_k, k \geq 1$.

F11: **Classification of Compact Surfaces**: Each compact surface with nonempty boundary components is isomorphic to a closed surface with holes. Each compact surface is completely specified by its orientability, an integer giving the genus or crosscap number, and the number b of holes.

F12: $S_g \# S_{g'} \cong S_{g+g'}$, $N_k \# N_{k'} \cong N_{k+k'}$, and $N_k \# S_g \cong N_{k+2g}$

F13: There are four mutually exclusive types of closed curves on surfaces:

(a) separating and contractible

(b) separating and non-contractible

(c) non-separating and orientation preserving

(d) non-separating and orientation reversing

EXAMPLES

E9: The sphere and the torus are orientable surfaces. Both are realizable in 3-space.

E10: Since the Klein bottle and the projective plane are non-orientable closed surfaces, it follows that they cannot be realized in 3-space.

E11: The Möbius strip is a non-orientable surface with boundary, and it can be realized in 3-space.

E12: Whereas the Jordan curve theorem asserts that every closed curve on the sphere separates the sphere, the Schönfliess theorem asserts the stronger result that every closed curve on a sphere bounds a disk.

E13: The pinched torus can be obtained by contracting ("pinching") a non-separating closed curve on a torus to a point.

7.1.2 Polygonal Complexes

DEFINITIONS

D30: A polygon is ***oriented*** if one of the two possible directions of traversal (i.e., clockwise or counterclockwise) of its boundary has been designated as preferred.

D31: Two topological spaces X and Y can be ***pasted together*** along homeomorphic subspaces by identifying the points of those subspaces under a homeomorphism.

D32: A ***polygonal complex*** is a structure obtained from a set of oriented polygons by pasting some of these polygons to each other and to themselves along their sides (which also results in the identification of corners). Within a polygonal complex,

- each polygon is called a ***face*** or a ***2-cell***;
- the image of arbitrarily many polygon sides that have been pasted together is called an ***edge*** or a ***1-cell***;
- the image of arbitrarily many polygon corners that have been pasted together is called a ***vertex*** or a ***0-cell***.

D33: The ***1-skeleton*** of a polygonal complex is the graph that is formed by its vertices and edges.

D34: Each edge e of a polygonal complex is given a preferred direction of traversal, and a traversal of that edge in the ***reverse direction*** within a walk in the 1-skeleton is denoted e^{-1}.

D35: A polygonal complex is ***consistently oriented at edge*** e if within the union of the oriented boundary walks, it is not traversed twice in the same direction. (Thus, a complex is consistently oriented at edge e if that edge results from a polygon side that was not pasted to another side, or if that edge results from pasting two sides together so that the traversal directions are *opposite*.)

D36: A polygonal complex is ***oriented*** if it is consistently oriented at every edge.

D37: The ***underlying topological space*** of a polygonal complex is the quotient space for the union of all the polygons after all the identifications.

D38: A polygonal complex is said to ***realize*** any topological space homeomorphic to its underlying space.

D39: Occurrences of an edge e or its inverse within a walk are called ***signed edges***.

D40: The ***oriented boundary walk*** of a face of a polygonal complex is the closed walk in the 1-skeleton that results from traversing the face boundary in the direction of orientation. (This walk is unique up to the choice of a starting/stopping vertex.)

D41: The ***signed boundary walk*** of a face of a polygonal complex is the list of the signed edges that occur on an oriented boundary walk of that face.

D42: The ***boundary-walk specification*** of a polygonal complex is a list of the signed boundary walks of the faces.

D43: The ***vertex variant*** of the boundary specification of a polygonal complex whose 1-skeleton is a simple graph gives the boundary walks as cyclic lists of vertices.

D44: A ***fundamental polygon*** for a closed surface is a polygon whose edges are pairwise identified and pasted so that the resulting polygonal complex has only one face and so that it realizes that surface.

D45: A ***specification of a fundamental polygon*** with $2n$ sides is its signed boundary walk.

D46: The ***standard fundamental polygon for the orientable surface*** S_g is specified as $a_1b_1a_1^{-1}b_1^{-1}a_2b_2a_2^{-1}b_2^{-1}...a_gb_ga_g^{-1}b_g^{-1}$.

D47: The ***standard fundamental polygon for the nonorientable surface*** N_k is specified as $a_1a_1a_2a_2...a_ka_k$.

FACTS

F14: A polygonal complex can be described combinatorially as the set of its signed boundary walks.

F15: A polygonal complex can realize any compact surface or pseudosurface.

F16: A polygonal complex realizes a pseudosurface or 2-manifold if and only if each side of each polygon is glued to exactly one other side; it realizes a 2-manifold if, in addition, every vertex has a topological neighborhood that is homeomorphic to a disk (this additional restriction serves to eliminate pinched disks).

EXAMPLES

E14: A ***book with*** n ***leaves,*** $n \geq 3$ (or an $n-$***book***) is a polygonal complex obtained by choosing a side in each of n polygons, often squares, and pasting all the chosen sides. The edge corresponding to the common side is called the ***spine*** of the book.

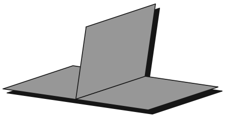


Figure 7.1.7 The 3-book is a polygonal complex that is not a surface.

E15: The 3-book with spine a can be specified as $\{ab_1c_1d_1, ab_2c_2d_2, ab_3c_3d_3\}$.

E16: The polygonal complex $\{abc, aeh^{-1}g^{-1}, bfi^{-1}e^{-1}, cgj^{-1}f^{-1}, hij\}$ is orientable but is not oriented. Reversing the orientation of the first polygon to $c^{-1}b^{-1}a^{-1}$ would make the complex oriented.

E17: The Möbius band can be specified as $\{abcd, efgb^{-1}, fid^{-1}h\}$, in which case the 1-skeleton is $K_{3,3}$.

E18: If we add a hexagon $aeh^{-1}c^{-1}g^{-1}i$ to the Möbius band specification in Example 17, the resulting polygonal complex realizes a projective plane.

E19: The polygonal complex $\{abb^{-1}a^{-1}bb^{-1}\}$ realizes the pinched torus with b as its pinch point.

E20: The ***standard*** fundamental polygon for the sphere S_0 has the form $\{a_1a_1^{-1}\}$.

7.1.3 Imbeddings

DEFINITIONS

D48: A ***topological realization of a graph*** G is obtained by first assigning to each of its edges a closed interval and then identifying endpoints of intervals according to the coincidences of the corresponding endpoints of edges of the graph.

D49: An ***immersion*** of a topological space is a continuous mapping that is locally one-to-one; that is, each point of the domain has a neighborhood that is mapped homeomorphically into the codomain.

D50: An ***imbedding*** is an immersion that is globally one-to-one.

D51: An ***imbedding of a graph*** G means an imbedding of a topological realization of G.

D52: A ***face*** of the imbedding is a connected component of the complement of the image.

D53: A ***cellular imbedding*** or ***2-cell imbedding*** of a graph into a surface is an imbedding such that the interior of each face is an open disk; thus, the complement of the image of the imbedding is a union of open disks.

D54: A ***strongly cellular imbedding*** is an imbedding such that the closure of each face is a closed disk; that is, no two points on the boundary of any face are identified.

D55: The ***minimum genus*** of a graph G (or sometimes, simply ***genus***) is the minimum of the set of integers g such that G is imbeddable in the orientable surface S_g. It is denoted by $\gamma_{\min}(G)$ or by $\gamma(G)$.

D56: A ***minimum genus imbedding*** of a graph G (or sometimes, simply ***genus imbedding***) is an imbedding of G into a closed surface of minimum genus.

D57: The ***maximum genus*** of a graph G is the maximum of the set of integers g such that G has a cellular imbedding in the orientable surface S_g. It is denoted by $\gamma_{\max}(G)$.

D58: A ***maximum genus imbedding*** of a graph is an imbedding into a closed surface of maximum genus.

D59: The ***minimum crosscap number*** of a graph G (or sometimes, simply ***crosscap number***) is the minimum of the set of integers k such that G is imbeddable in the nonorientable surface N_k. It is denoted by $\tilde{\gamma}_{\min}(G)$ or by $\tilde{\gamma}(G)$.

D60: A ***minimum crosscap imbedding*** is an imbedding into a closed nonorientable surface of minimum crosscap number.

D61: The ***maximum crosscap number*** of a graph G is the maximum of the set of integers k such that G has a cellular imbedding in the nonorientable surface N_k. It is denoted by $\tilde{\gamma}_{\max}(G)$.

D62: A ***maximum crosscap imbedding*** of a graph is 2-cell imbedding into a closed nonorientable surface of maximum crosscap number.

FACTS

F17: Every finite graph has a topological realization that can be imbedded in Euclidean 3-space.

F18: A disconnected graph has no cellular imbedding.

F19: [Yo63] Every connected graph has a minimum genus imbedding that is cellular.

F20: [PPPV87] If a connected graph is not a tree, then it has a minimum crosscap imbedding that is cellular.

F21: [Du66] Let $g' \le g \le g''$. If a graph admits a 2-cell imbedding in the surfaces $S_{g'}$ and $S_{g''}$ then it also admits a 2-cell imbedding in S_g.

F22: Let $k' \le k \le k''$. If a graph admits a 2-cell imbedding in the surfaces $N_{k'}$ and $N_{k''}$ then it also admits a 2-cell imbedding in S_k.

F23: [Wh33] Each planar 3-connected graph admits an essentially unique imbedding in the sphere. This is not generally true for imbeddings into other surfaces, not even for genus imbeddings.

F24: [Th89] The problem of determining the minimum genus of a graph is NP-hard.

F25: [Mo99] For a given graph and a fixed surface there exists a linear-time algorithm that either finds an imbedding of the graph in that surface or finds an obstruction for such an imbedding. The algorithm is not good for practical purposes since it subsumes the knowledge of all forbidden graphs for a given surface. The collection of such graphs may be quite large for a surface of moderate size genus.

F26: **Euler polyhedral equation**: Each cellular imbedding of a graph with v vertices, e edges and f faces into a surface S satisfies the relation

$$v - e + f = \chi(S)$$

F27: [Au63] For any graph G, $\bar{\gamma}(G) \le 2\gamma(G) + 1$, however, the gap may be arbitrarily large.

F28: [At68] Any graph can be imbedded in a 3-book.

F29: Every simple graph can be immersed in the plane by spacing the vertices evenly around the unit circle and joining adjacent vertices with line segments.

EXAMPLES

E21: Figure 7.1.8 shows two imbeddings of the complete graph K_4 on the torus, one noncellular and the other cellular.

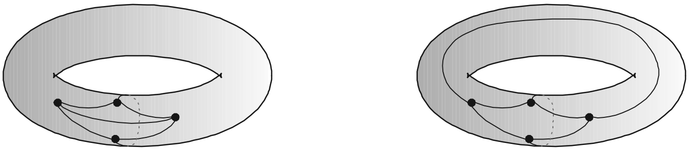


Figure 7.1.8 **Two toroidal imbeddings of K_4.**

E22: The vertex-variant specification (1234)(5678)(1265)(2376)(3487)(4158) for the cube graph Q_3 corresponds to the following imbedding:

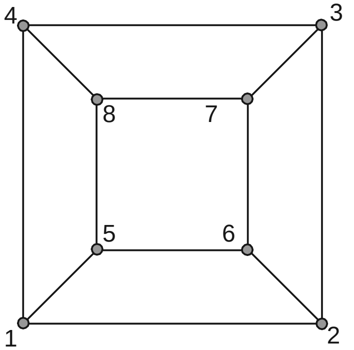


Figure 7.1.9 **A standard imbedding of the cube graph Q_3 in the sphere.**

E23: Two nonequivalent imbeddings of the cube graph Q_3 in the torus given by the following vertex-variant specifications

$$(123765)(341587)(234876)(126584)$$
$$(148762)(123785)(326584)(567341)$$

are shown in Figure 7.1.10.

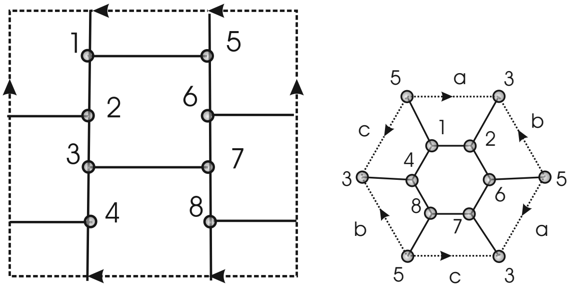


Figure 7.1.10 **Two hexagonal imbeddings of Q_3 in the torus.**

7.1.4 Combinatorial Descriptions of Maps

DEFINITIONS

D63: A ***graph*** can be defined alternatively as a combinatorial structure $\langle V, E\rangle$ with ground set S, as follows:

- The elements of the set S are called ***half-edges***.
- E is a partition of S into cells of size two, such that each half-edge is paired with what amounts to the other half of the same edge. This partition is often represented as the set of orbits of an involution τ.
- V is a partition of the half-edges according to the vertex at which they are incident.

D64: A ***rotation at a vertex*** is a cyclic permutation on the set of half-edges at that vertex.

D65: The ***surface rotation at a vertex*** v of a graph imbedding is the cyclic ordering of the half-edges at v on the surface. If the surface is orientable, this ordering is taken to be consistent with the orientation.

D66: A ***(global) rotation*** on a graph is an assignment of a rotation at each vertex. This corresponds to a permutation ρ on the set of half-edges whose orbits are the rotations at the vertices.

D67: The ***(global) surface rotation on an imbedded graph*** is the set of surface rotations at all the vertices.

D68: The ***induced imbedding*** of a global rotation ρ on a graph is an imbedding of that graph whose global surface rotation is ρ. (The face tracing algorithm below serves as proof that such an imbedding exists. It is obviously unique.)

D69: A ***face tracing*** for a global rotation on a graph is a list of the boundary walks of the faces of an induced imbedding.

D70: The ***signature*** of a graph $G = (V, E)$ is a subset $\Lambda \subseteq E_G$, whose edges are called ***switches***. They represent the edges whose traversal switches the sense of orientation in an imbedding.

D71: A ***generalized rotation*** is a pair (ρ, Λ) composed of a global rotation and a signature.

ALGORITHM

We suppose that a global rotation ρ and an involution τ on the set of half-edges of a graph G are given as input. We want to do a face tracing. To make this easily understood, we use notation of the form e and e^{-1} for two half-edges paired by the involution τ, i.e., for the two different ends of the same edge. To each cycle of the rotation ρ, we visualize a vertex at which the half-edges within that cycle are simultaneously incident in the graph G.

Algorithm 7.1.1: Face-Tracing Algorithm

```
Input: half-edge list $E^{\pm}$, involution $\tau$, rotation $\rho$
Output: list of all face-boundaries of the induced imbedding
    {Initialize} Mark all half-edges unused
    While any unused half-edges remain
        Choose next (lex order) unused half-edge $y$ from $E^{\pm}$
        Start new cycle by writing left paren    "("
        $x := y$
        Repeat
            Write $x$ next in current cycle
            $x := \rho(\tau(x))$    (next half-edge)
            Until $x = y$
        Close current cycle by writing right paren    ")"
        Continue with next iteration of while-loop
```

The algorithm for a generalized rotation is slightly more complicated, since it involves reversal of cycles. See, for example, Chapter 4 of [GrTu87].

EXAMPLES

E24: A convenient way to apply the Face-Tracing Algorithm uses a table that lists the half-edges incident at each vertex, in the cyclic order of the rotation there. For instance, this table presents an imbedding of the graph K_4 in the sphere S_0.

$v_1.$	a^+	b^+	c^+
$v_2.$	a^-	e^+	d^+
$v_2.$	c^-	f^-	e^-
$v_2.$	b^-	d^-	f^+

$$\rho = (a^+, b^+, c^+)(a^-, e^+, d^+)(c^-, f^-, e^-)(b^-, d^-, f^+)$$

$$\tau = (a^+a^-)(b^+b^-)(c^+c^-)(d^+d^-)(e^+e^-)(f^+f^-)$$

The composition permutation $\rho\tau$ has a disjoint cycle representation with four 3-cycles, which correspond to the boundary walks of the four triangular faces. Using notation that clearly associates corresponding half-edges avoids the need to write the involution.

E25: The following table presents an imbedding of K_4 in the torus S_1.

$v_1.$	a^+	b^+	c^+
$v_2.$	a^-	d^+	e^+
$v_2.$	c^-	f^-	e^-
$v_2.$	b^-	f^+	d^-

The imbedding has one 4-sided face and one 8-sided face.

E26: At the left of Figure 7.1.11 is an imbedding of the dipole D_3 on the sphere. At the right are shown the three polygons of that imbedding, prior to pasting.

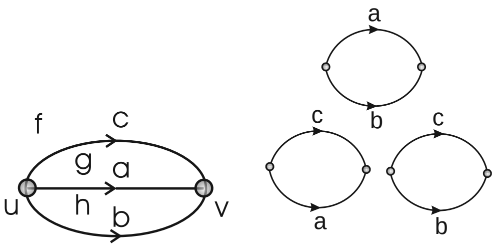


Figure 7.1.11 **A spherical imbedding of the dipole D_3.**

Since D_3 is not a simple graph, the specification of that imbedding as a set of boundary walks

$$f = (b, c^{-1}), \quad g = (c, a^{-1}), \quad h = (a, b^{-1})$$

uses edges, not vertices, as does the specification by global rotation

$$\rho = u : (abc) \quad v : (a^{-1}c^{-1}b^{-1})$$

E27: At the left of Figure 7.1.12 is an imbedding of the dipole D_3 on the torus. At the right is shown the one polygon of that imbedding, prior to pasting its sides.

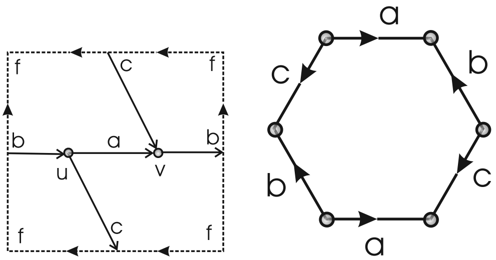


Figure 7.1.12 **Toroidal imbedding of the dipole D_3.**

As a set of boundary walks, it has the specification

$$f = (a, c^{-1}, b, a^{-1}, c, b^{-1})$$

and by global rotation the specification

$$\rho = u : (abc) \quad v : (a^{-1}b^{-1}c^{-1})$$

E28: At the left of Figure 7.1.13 is an imbedding of the dipole D_3 on the Klein bottle. At the right is shown the one polygon of that imbedding, prior to pasting its sides.

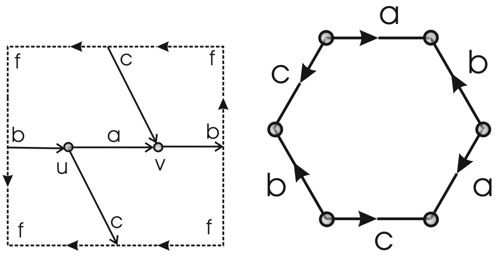


Figure 7.1.13 **Klein bottle imbedding of the dipole D_3.**

As a set of boundary walks, it has the specification

$$f = (a, c^{-1}, b, c^{-1}, a, b^{-1})$$

and by global generalized rotation the specification

$$\rho = u : (abc) \quad v : (a^{-1}b^{-1}c^{-1}) \quad \Lambda = \{b\}$$

FACTS

F30: The set of global rotations on a graph is in bijective correspondence with the set of oriented, cellular imbeddings of that graph.

F31: Generalized rotations correspond to cellular imbeddings into arbitrary closed surfaces.

F32: The imbedding of a graph specified by a generalized rotation is nonorientable if and only if there is a cycle in G containing an odd number of switches.

References

[Ar96] D. Archdeacon, Topological graph theory; a survey, Surveys in graph theory (San Francisco, CA, 1995) *Congr. Numer.* **115** (1996), 5–54.

[At68] G. Atneosen, *On the Embeddability of Compacta in n-Books: Intrinsic and Extrinsic Properties*, Ph. D. Thesis, Michigan State University, 1968.

[BiWh79] N. L. Biggs, A. T. White, *Permutation Groups and Combinatorial Structures*, LMSLNS 33, Cambridge University Press, 1979.

[Du66] R. A. Duke, The genus, regional number, and Betti number of a graph, *Canad. J. Math.* **18** (1966), 817–822.

[Gr00] J. L. Gross, Topological Graph Theory, *Handbook of Discrete and Combinatorial Mathematics*, ed. by K. H. Rosen, CRC Press, 2000, 574–579.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover, 2001. (First Edition, Wiley-Interscience, 1987.)

[Mo99] B. Mohar, A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discrete Math.* **12** (1999), 6–26.

[PPPV87] T. D. Parsons, G. Pica, T. Pisanski, A. G. S. Ventre, Orientably simple graphs, *Math. Slovaca* **37** (1987), 391–394.

[Wh01] A. T. White, *Graphs of Groups on Surfaces*, North-Holland Mathematical Studies 188, Elsevier, 2001.

[Wh33] H. Whitney, 2-isomorphic graphs, *Amer. J. Math.* **55** (1933), 73–84.

[Yo63] J. W. T. Youngs, Minimal imbeddings and the genus of a graph, *J. Math. Mech.* **12** (1963), 303–315.

7.2 MINIMUM AND MAXIMUM IMBEDDINGS

Jianer Chen, Texas A&M University

7.2.1 Fundamentals
7.2.2 Upper Bounds: Planarity and Upper-Imbeddability
7.2.3 Lower Bounds
7.2.4 Kuratowski-Type Theorems
7.2.5 Algorithmic Issues
References

Introduction

The study of minimum graph imbeddings has always been a dominant concern of topological graph theory. Maximum graph imbeddings were among the first of the new topics to emerge at the onset of the modern era. The theoretical importance of these topics has been enhanced by impressive connections to such areas as VLSI design, computer algorithms and complexity, and computer graphics.

The graphs in our discussion may have multiple adjacencies or self-adjacencies. All graphs in our discussion are assumed implicitly to be connected.

7.2.1 Fundamentals

DEFINITIONS

D1: An ***imbedding*** of a graph G in an orientable surface S is a continuous one-to-one function $\rho : G \to S$ from a topological representation of the graph G into the surface S.

D2: The ***image of an imbedding*** $\rho : G \to S$ is the subspace $\rho(G)$ of the surface S. Sometimes, one refers to $\rho(G)$ either as "the graph" or as "the imbedding".

D3: Each connected component of $S - \rho(G)$ is called a ***face*** of the imbedding $\rho(G)$.

D4: The imbedding is ***cellular*** if the interior of each face of the imbedding is homeomorphic to a 2-dimensional open disk. (Our discussion will be restricted to cellular graph imbeddings.)

D5: The ***genus of the imbedding*** $\rho : G \to S_g$ is the genus g of the imbedding surface.

D6: The ***crosscap number of the imbedding*** $\rho : G \to N_k$ is the crosscap number k of the imbedding surface. This is also called the ***non-orientable genus***.

D7: The ***minimum genus*** $\gamma_{\min}(G)$ (or simply the ***genus*** $\gamma(G)$) of a graph G is the minimum integer g such that there exists an imbedding of G into the orientable surface S_g of genus g.

D8: The ***minimum crosscap number*** $\tilde{\gamma}_{\min}(G)$ (or simply the ***crosscap number*** $\tilde{\gamma}(G)$) of a graph G is the minimum integer k such that there exists an imbedding of G into the surface N_k of crosscap number k. For a planar graph, $\tilde{\gamma}_{min}(G) = 0$.

D9: The ***maximum genus*** $\gamma_{\max}(G)$ of a graph G is the maximum integer g such that there exists a (cellular) imbedding of G into the orientable surface of genus g.

D10: The ***maximum crosscap number*** $\gamma_{\max}(G)$ of a graph G is the maximum integer k such that there exists a (cellular) imbedding of G into the non-orientable surface of crosscap number k.

D11: The number $|E|-|V|+1$ is called the ***cycle rank*** (or the ***Betti number***) of the graph G, denoted $\beta(G)$. This is best regarded conceptually as the number of non-tree edges relative to a spanning tree for G.

EXAMPLES

E1: For a tree, the minimum and the maximum genus are 0. Likewise, the minimum and the maximum crosscap number are 0.

E2: For the bouquet B_n, the minimum genus and crosscap number are 0. The maximum genus and crosscap number are $\lfloor n/2 \rfloor$ and n, respectively.

E3: For the dipole D_n, the minimum genus and crosscap number are 0. The maximum genus and crosscap number are $\lfloor (n-1)/2 \rfloor$ and $n-1$, respectively.

E4: The minimum genus of the complete graph K_4 is 0, and the maximum genus of K_4 is equal to 1.

FACTS

F1: [Br32] Any orientable surface is homeomorphic to one of the surfaces S_g obtained by adding g handles to the sphere, and any non-orientable surface is homeomorphic to one of the surfaces N_k obtained by adding k crosscaps to the sphere.

F2: The genus of any imbedding of a graph G is an integer between 0 and $\lfloor \beta(G)/2 \rfloor$, where $\beta(G)$ is the cycle rank of the graph G. The crosscap number is an integer between 0 and $\beta(G)$.

F3: **Euler Polyhedral Equation** (e.g., see [GrTu87]): An orientable imbedding $\rho : G \to S$ of a graph G with vertex set V, edge set E, face set F, and genus g satisfies the relation

$$|V| - |E| + |F| = 2 - 2g$$

F4: [BHKY62] (Additivity of Minimum Genus) Let $\{B_1, B_2, \cdots, B_k\}$ be the collection of 2-connected components of a graph G. Then

$$\gamma_{\min}(G) = \sum_{i=1}^{k} \gamma_{\min}(B_i)$$

F5: [NoStWh71] (Additivity of Maximum Genus) Let $\{B_1, B_2, \cdots, B_k\}$ be the collection of 2-edge connected components of a graph G. Then

$$\gamma_{\max}(G) = \sum_{i=1}^{k} \gamma_{\max}(B_i)$$

Whitney Synthesis of 2-Edge-Connected Graphs

According to Facts 4 and 5, in most cases, we may concentrate on the minimum genus of 2-connected graphs and on the maximum genus of 2-edge-connected graphs. Whitney's description how to synthesis any 2-edge-connected graph from a cycle facilitates an inductive approach to proving theorems about such graphs.

DEFINITIONS

D12: An ***ear decomposition*** $D = [P_1, P_2, \cdots, P_r]$ of a graph G is a partition of the edge set of G into an ordered collection P_1, P_2, $\cdots$, P_r such that P_1 is a simple cycle and P_i, $i \geq 2$, is a path with only its endpoints in common with $P_1 + \cdots + P_{i-1}$. Each path P_i is called an ***ear***. (In some drawings of this construction, the paths are shaped like the outer edge of a human ear.)

D13: ***Inserting a new edge*** e ***into an imbedding*** $\rho : G \to S$ means that the two ends of e are inserted into face corners in $\rho(G)$ to make an imbedding for the graph $G + e$.

D14: An inserted edge e ***splits a face*** f of an imbedding if the two ends of an edge e are inserted into the corners of the same face f in $\rho(G)$. Then f splits into two faces and the imbedding genus is unchanged. In this case, the two sides of the new edge e belong to two different faces in the resulting imbedding for $G + e$. See Figure 7.2.1(a) for illustration.

D15: If the two ends of e are inserted into the corners of two different faces f_1 and f_2 in an imbedding $\rho(G)$, then the edge e ***merges the faces*** f_1 and f_2 into a single larger face and increases the imbedding genus by 1. In this case, the two sides of the new edge e belong to the same face (i.e., the new larger face) in the resulting imbedding for $G + e$. See Figure 7.2.1(b) for illustration.

Topologically, the operation of merging by edge inserting can be implemented as follows: first we cut along the boundaries of the two faces f_1 and f_2 and leave two holes on the surface. Then we add a handle to the surface by pasting the two ends of a cylinder with two open ends to the boundaries of the two holes on the surface, respectively. The new edge e now runs along the new handle.

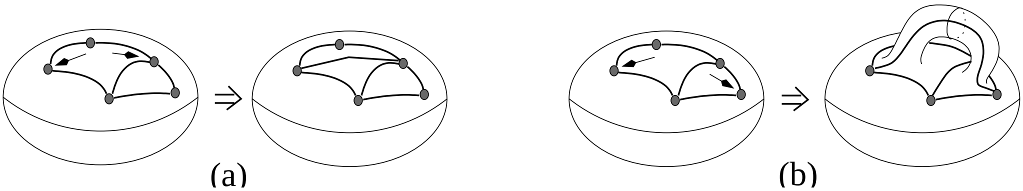


Figure 7.2.1 Inserting a new edge into an imbedding.

D16: ***Deleting an edge*** e ***from an imbedding*** $\rho : G \to S$ results naturally in an imbedding of the graph $G - e$, as indicated by the two cases described in the following two definitions.

D17: If the two sides of a deleted edge e belong to two different faces f_1 and f_2 in an imbedding $\rho(G')$, then deleting e from $\rho(G')$ ***merges the faces*** without changing the imbedding genus.

D18: If the two sides of an edge e, not a cutedge, belong to the same face in an imbedding $\rho(G')$, then there is a closed curve C in that (closed) face that intersects graph G' only in a single point, which lies in the interior of edge e. We say that deleting e from the imbedding $\rho(G')$ ***splits the face*** into two faces and decreases the imbedding genus by 1. This is accomplished by cutting the surface open on curve C (after deleting edge e), and then capping the two holes with disks.

FACTS

F6: [Wh32] A graph G has an ear decomposition if and only if G is 2-edge-connected.

F7: Inserting an edge into a graph imbedding can never decrease the imbedding genus, and deleting an edge can never increase the imbedding genus.

7.2.2 Upper Bounds: Planarity and Upper-Imbeddability

There has been extensive research in the study on graphs of minimum genus 0 and on graphs G of maximum genus $\lfloor \beta(G)/2 \rfloor$. Maximum crosscap number is equal to cycle rank, for all graphs.

DEFINITIONS

D19: A ***planar graph*** is a graph of minimum genus 0.

D20: An ***upper-imbeddable graph*** is a graph G of maximum genus $\lfloor \beta(G)/2 \rfloor$. (This is the theoretical upper bound given in Fact 2.)

D21: A ***minor*** of a graph G is a graph H that can be obtained from G by a sequence of edge-deletions and contractions.

D22: An ear decomposition $D = [P_1, P_2, \cdots, P_r]$ of a graph G is ***3-connected*** if $r \geq 3$, $P_1 + P_2 + P_3$ is homeomorphic to the complete graph K_4, and every subgraph $P_1 + \cdots P_i$ of G, $3 \leq i \leq r$, is homeomorphic to a 3-connected graph.

The study of 3-connected planar graphs has played an important role in the study of planar graphs. A number of important properties of planar imbeddings of 3-connected planar graphs can be derived based on a special ear-decomposition of 3-connected graphs.

FACTS

F8: [Ku30] (Kuratowski theorem) A graph is planar if and only if it contains no subgraph homeomorphic to either K_5 or $K_{3,3}$.

F9: [Wa37] A graph is planar if and only if it has neither K_5 nor $K_{3,3}$ as a minor.

F10: [Ch90] Every 3-connected graph has a 3-connected ear decomposition. (Using induction, we can derive a number of well-known results on 3-connected planar graphs.)

F11: [Wh33] There is essentially only one way to imbed a 3-connected planar graph in the plane.

F12: [Tu60] Every 3-connected planar graph has a planar imbedding in the geometric plane in which every face, except the outer face, is a convex polygon.

F13: [Fa48] Every planar graph has a planar imbedding in the geometric plane in which every edge is a straight line segment.

F14: [NoStWh71] A graph G is upper-imbeddable if and only if G has a spanning tree T such that the co-tree $G - T$ has at most one odd component.

F15: [Ku74] Every 4-edge-connected graph contains two edge-disjoint spanning trees.

F16: Every 4-edge-connected graph is upper-imbeddable. (This follows immediately from the two preceding facts and is commonly ascribed to [Ku74].)

F17: [Ed65] The maximum crosscap number of a graph G is $\beta(G)$. (See [GrTu87].)

REMARKS

R1: Because of Fact 16, the study of upper-imbeddability has been focused on graphs that are not 4-edge connected.

R2: See [ChArGr96] for a general construction of 3-edge connected graphs that are not upper-imbeddable.

EXAMPLE

E5: The complete bipartite graph $K_{3,n}$ is upper-imbeddable. (It is easy to construct a spanning tree whose complement is connected.)

Deficiency

Early study on graph upper-imbeddability was focused on derivation of the upper-imbeddability of special graph classes. It became clear later that most of these results could be obtained from effective characterizations of graph maximum genus. There have been several productive characterizations of graph maximum genus.

DEFINITIONS

D23: The ***co-tree*** for a spanning tree T of a graph G is the edge complement $G - T$. (The co-tree $G - T$ need not be connected.)

D24: A connected component H of the co-tree $G - T$ is called an ***even component*** (resp. ***odd component***) if the number of edges in H is even (resp. odd).

D25: The ***deficiency*** $\xi(G, T)$ of a spanning tree T is defined to be the number of odd components of the co-tree $G - T$.

D26: [Xu79a] The ***deficiency*** $\xi(G)$ ***of the graph*** G is defined to be the minimum of $\xi(G, T)$ over all spanning trees T of the graph G.

D27: A spanning tree T of G is called a ***Xuong tree*** if its deficiency $\xi(G, T)$ of T is equal to the deficiency $\xi(G)$ of the graph G.

D28: Two edges are ***adjacent*** if they share a common end.

D29: [Ne81a] For any subset A of edges of a graph G, define $C_o(G-A)$ and $C_e(G-A)$ to be the numbers of connected components in $G-A$ with odd cycle rank and with even cycle rank, respectively, and let

$$\nu(G,A) = C_e(G-A) + 2C_o(G-A) - |A| - 1.$$

The ***Nebeský nu-invariant*** $\nu(G)$ is defined to be the maximum of $\nu(G,A)$ over all edge subsets A of the graph G.

FACTS

F18: Inserting two adjacent edges e_1 and e_2 to an imbedding $\rho(G)$ of a graph G can always increase the imbedding genus.

F19: [ChKa99] A graph G has a spanning tree such that the co-tree $G-T$ contains at least $\gamma_{\max}(G)$ pairs of adjacent edges.

F20: [Xu79a] The maximum genus $\gamma_{\max}(G)$ of a graph G is equal to $(\beta(G)-\xi(G))/2$.

F21: [Ne81a] $\nu(G) = \xi(G)$. The maximum genus $\gamma_{\max}(G)$ of the graph G is equal to $(\beta(G)-\nu(G))/2$.

F22: [ChKa99] Maximum genus can also be characterized in terms of ear decompositions of the graph.

F23: A graph G is upper-imbeddable if and only if $C_e(G-A)+2C_o(G-A)-2 \leq |A|$, for all edge subsets A of G, where $C_o(G-A)$ and $C_e(G-A)$ are the numbers of connected components in $G-A$ with odd cycle rank and with even cycle rank, respectively.

F24: [Ne81b] Locally connected graphs are upper-imbeddable.

F25: [PaXu79] Cyclically 4-edge connected graphs are upper-imbeddable.

F26: [SkNe89] k-regular vertex-transitive graphs of girth g such that $k \geq 4$ or $g \geq 4$ are upper-imbeddable.

F27: [Skov91] Loopless graphs of diameter 2 are upper-imbeddable.

F28: [HuLi00a] $(4k+2)$-regular graphs and $(2k)$-regular bipartite graphs are upper-imbeddable.

EXAMPLES

E6: The complete graphs K_n are upper-imbeddable for all $n \geq 1$.

E7: The complete bipartite graphs $K_{n,m}$ are upper-imbeddable for all $n,m \geq 1$.

7.2.3 Lower Bounds

The *Euler Polyhedral Equation* $|V|-|E|+|F| = 2-2g$ implies that a minimum genus imbedding of a graph G is an imbedding with the largest number of faces, and that a maximum genus imbedding of G is one with the minimum number of faces.

Lower Bounds for Minimum Genus

DEFINITIONS

D30: The ***size of a face*** f, denoted $size(f)$ is the number of edge-steps in its boundary walk. (If f is an n-gon, then $size(f) = n$; that is, each edge in the boundary walk of f that occurs twice is counted twice.)

D31: The ***girth*** of a graph G is the length of a shortest cycle in G. It is undefined for a tree.

D32: The ***Heawood number*** of a surface S with Euler characteristic c and chromatic number $chr(S)$ is

$$H(S) = \left\lfloor \frac{7 + \sqrt{49 - 24c}}{2} \right\rfloor$$

FACTS

F29: *Edge-Face Equality*: For any imbedded graph $G = \langle V, E \rangle$ with face set F,

$$2|E| = \sum_{f \in F} size(f)$$

This is because each edge of G is counted twice on the right size.

F30: The girth of a simple graph is at least 3.

F31: The girth of a simple bipartite graph is at least 4.

F32: For any imbedded graph $G = \langle V, E \rangle$ with face set F, and for any face $f \in F$,

$$girth(G) \leq size(f)$$

F33: *Edge-Face Inequality*: For any imbedded graph $G = \langle V, E \rangle$ with face set F,

$$2|E| \geq girth(G)|F|$$

This follows from the Edge-Face Equality and Fact 32.

F34: For any graph $G = \langle V, E \rangle$,

$$\gamma_{\min}(G) \geq \frac{(girth(G) - 2)|E|}{2girth(G)} - \frac{|V|}{2} + 1 \quad \text{and} \quad \tilde{\gamma}_{\min}(G) \geq \frac{(girth(G) - 2)|E|}{girth(G)} - |V| + 2$$

F35: Let G be a simple graph. Then

$$\gamma_{\min}(G) \geq \frac{|E|}{6} - \frac{|V|}{2} + 1 \quad \text{and} \quad \tilde{\gamma}_{\min}(G) \geq \frac{|E|}{3} - |V| + 2$$

F36: Let G be a simple bipartite graph. Then

$$\gamma_{\min}(G) \geq \frac{|E|}{4} - \frac{|V|}{2} + 1 \quad \text{and} \quad \tilde{\gamma}_{\min}(G) \geq \frac{|E|}{2} - |V| + 2$$

F37: If a simple graph $G = \langle V, E \rangle$ has a triangulated orientable imbedding, then the imbedding is a minimum genus imbedding and

$$\gamma_{\min}(G) = \frac{|E|}{6} - \frac{|V|}{2} + 1$$

F38: If a simple graph $G = \langle V, E \rangle$ has a triangulated non-orientable imbedding, then the imbedding is a minimum crosscap imbedding and

$$\tilde{\gamma}_{\min}(G) = \frac{|E|}{3} - |V| + 2$$

F39: If a simple bipartite graph $G = \langle V, E \rangle$ has a quadrangulated orientable imbedding, then the imbedding is a minimum genus imbedding and

$$\gamma_{\min}(G) = \frac{|E|}{4} - \frac{|V|}{2} + 1$$

F40: If a simple bipartite graph $G = \langle V, E \rangle$ has a quadrangulated non-orientable imbedding, then the imbedding is a minimum crosscap imbedding and

$$\tilde{\gamma}_{\min}(G) = \frac{|E|}{2} - |V| + 2$$

REMARK

R3: A classical approach to computing the minimum genus of a simple (non-bipartite) graph is to try to construct a triangulated imbedding of the graph, or for a simple bipartite graph, a quadrangulated imbedding. This approach has been very successful in deriving minimum genus of well-known graph classes. Voltage graphs are presently the main tool for constructing triangulations and quadrangulations.

FACTS

Proofs (or sketches of proofs) of most of the following facts appear in [GrTu87].

F41: [RiYo68] For the complete graph K_n of n vertices, with $n \geq 3$,

$$\gamma_{\min}(K_n) = \left\lceil \frac{(n-3)(n-4)}{12} \right\rceil \quad \text{and} \quad \tilde{\gamma}_{\min}(K_n) = \left\lceil \frac{(n-3)(n-4)}{6} \right\rceil$$

except that ([Fr34]) $\tilde{\gamma}_{min}(K_7) = 3$.

F42: [RiYo68] (Formerly called the Heawood Conjecture) The chromatic number of every surface except the Klein bottle N_2 is equal to its Heawood number. This is a corollary to Fact 41.

F43: Let G be a graph of chromatic number c, then

$$\gamma_{\min}(G) \geq \frac{(c^2 - 7c + 12)}{12}$$

F44: [Ri65] For the complete bipartite graph $K_{m,n}$ with $m, n \geq 2$,

$$\gamma_{\min}(K_{m,n}) = \left\lceil \frac{(n-2)(n-2)}{4} \right\rceil$$

F45: [Ri55] For the cube graph Q_n of n vertices, with $n \geq 3$,

$$\gamma_{\min}(Q_n) = 1 + 2^{n-3}(n-4)$$

F46: [Ju78a] For the cube graph Q_n of n vertices, with $n \geq 3$,

$$\tilde{\gamma}_{\min}(Q_n) = 2\gamma_{\min}(Q_n)$$

except that for $n = 4, 5$, $\tilde{\gamma}_{\min}(Q_n) = 2\gamma_{\min}(Q_n) + 1$.

F47: [Wh70] Let $G = C_{n_1} \times \cdots \times C_{n_r}$, where $r > 1$ and $n_j > 3$ for all j. If every n_j is even, then

$$\gamma_{\min}(G) = 1 + \frac{|V|(r-2)}{4}$$

and if n_1 and n_2 are even and n_3 is odd, then

$$\tilde{\gamma}_{\min}(G) = 2 + \frac{|V|(r-2)}{2}$$

F48: [Pi80] Let G and H be 1-factorable, r-regular graphs of girth at least 4. If G and H are both bipartite, then

$$\gamma_{\min}(G \times H) = 1 + \frac{|V_G| \cdot ||V_H| \cdot (r-2)}{4}$$

and if G and H are not both bipartite, then

$$\tilde{\gamma}_{\min}(G \times H) = 2 + \frac{|V_G| \cdot ||V_H| \cdot (r-2)}{2}$$

Lower Bounds on Maximum Genus

By the additivity theorem for maximum genus, we may confine our interest to graphs that are 2-connected. Subdivision and smoothing do not change the maximum genus. The maximum genus of a 4-edge connected graph G is $\lfloor \beta(G)/2 \rfloor$. This leads us to focus on lower bounds for the maximum genus of graphs of minimum degree at least 3 and are not 4-edge-connected.

DEFINITION

D33: [GrKlRi93] A ***necklace of type*** (r, s) is a graph obtained from an $(r+s)$-cycle by doubling r disjoint edges and then attaching a self-loop at each of the s vertices that is not an endpoint of a doubled edge.

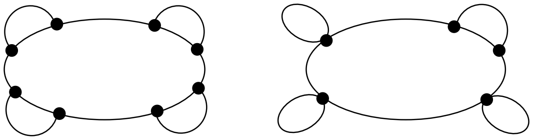


Figure 7.2.2 Left: a type $(4, 0)$ necklace; Right: a type $(1, 3)$ necklace.

FACTS

F49: The cycle rank of a necklace of type (r, s) is $r + s + 1$.

F50: [GrKlRi93] The maximum genus of every necklace is 1.

F51: [ChGr95] The necklace construction is essentially the only way to construct graphs of large cycle rank and small maximum genus.

F52: [ChKaGr96] Let G be a simple graph of minimum degree at least 3. Then $\gamma_{\max}(G) \geq \lceil \beta(G)/4 \rceil$. This bound is tight, since that there are infinitely many simple graphs G of minimum degree 3 whose maximum genus is arbitrarily close to $\beta(G)/4$.

F53: Let G be a 2-edge connected simple graph of minimum degree at least 3. Then $\gamma_{\max}(G) \geq \lceil \beta(G)/3 \rceil$. (See [Ar+02].) Fact 54 indicates that this result cannot be improved for 3-edge-connected graphs.

F54: [ChKaGr96] There exists an infinite class of 3-edge-connected simple graphs G whose maximum genus is equal to $\lceil \beta(G)/3 \rceil$.

F55: [Ar+02] Let G be a 3-edge connected graph. Then $\gamma_{\max}(G) \geq \lceil \beta(G)/3 \rceil$. This bound is tight, by Fact 54.

F56: [ChArGr96] Fact 16 and Facts 53–55 for k-edge connected graphs can be translated without much trouble to k-vertex connected graphs.

F57: Finally, we mention that lower bounds on graph maximum genus have also been derived in terms of graph connectivity, independent number, girth, and chromatic number [HuLi00b, LiLi00].

7.2.4 Kuratowski-Type Theorems

The Kuratowski theorem is equivalent to the assertion that a graph G is planar if and only if G has neither K_5 nor $K_{3,3}$ as a minor. From this point of view, Kuratowski theorem gives a prototype of characterization of planar graphs in terms of a finite number of forbidden minors, which turns out to be very important not only in the study of graph imbeddings, but also in other subareas in graph theory.

Complete Forbidden Sets for Minimum Genus

Erdős and König [Koni36] raised the question whether there is a Kuratowski-type theorem for the class of graphs that are imbeddable (not necessarily 2-cellular imbeddable) in a fixed surface S.

DEFINITIONS

D34: A class $\mathcal{F}$ of graphs is ***closed under minors*** if for each graph G in $\mathcal{F}$, all minors of G are also in $\mathcal{F}$.

D35: Let $\mathcal{F}$ be a graph class closed under minors. A graph G is a ***minimal forbidden minor*** for $\mathcal{F}$ if G is not in $\mathcal{F}$ but every proper minor of G is in $\mathcal{F}$.

D36: Let $\mathcal{F}$ be a graph class that is closed under minors. A set $\mathcal{M}$ of minimal forbidden minors is a ***complete set of forbidden minors*** for $\mathcal{F}$ if for every graph G that is not in $\mathcal{F}$, there exists a graph in $\mathcal{M}$ that is a minor of G.

FACTS

F58: The graphs K_5 and $K_{3,3}$ make a complete set of forbidden minors for the class of planar graphs. (This is an alternative version of the Kuratowski Theorem.)

F59: [GlHu78] There exists a finite complete set of forbidden minors for the class of graphs that can be imbedded in the projective plane N_1. A complete list can be found in [GlHuWa79] or [Ar81].

F60: [ArHu89] For every non-orientable surface N, there is a finite set $\mathcal{F}_N$ of forbidden minors for the class of graphs that are imbeddable in N.

F61: [RoSe88] (Formerly known as Wagner's Conjecture) Any class of graphs closed under minors has a finite complete set of forbidden minors. An extraordinary series of papers ([RoSe85, RoSe88, RoSe90a, RoSe90b, RoSe95]) led to this result.

F62: For every integer $g \geq 0$, the class of graphs whose minimum genus is bounded by g is closed under minors.

F63: [RoSe90b] For every integer $g \geq 0$, there is a finite complete set of minimal forbidden minors for the class of graphs of minimum genus bounded by g.

F64: [Se93] The size of a minimum complete set of minimal forbidden minors for graphs of minimum genus bounded by g is at most $2^{2^{(6g+9)^9}}$.

REMARKS

R4: Contracting an edge e of G on a planar imbedding $\rho(G)$ can be accomplished by continuously "shrinking" the edge e on the plane until its two ends are identified. This yields a planar imbedding of the contracted graph G/e. Moreover, edge deletion does not increase imbedding genus. Thus, the class of planar graphs is closed under minors. Using a similar argument, one can show that the minimum genus of a minor of a graph G can never be larger than $\gamma_{\min}(G)$.

R5: A constructive proof for Fact 63 was developed by Mohar [Mo99]. There has been further effort to simplify the proof [Th97b]. On the other hand, it has remained as a challenge, even for very small g such as $g = 1$, to give a good estimation on the number of graphs or the size of the graphs in the set of minimal forbidden minors in Fact 63.

Complete Forbidden Sets for Maximum Genus

Classes of graphs of given maximum genus are not closed under minors. For example, the bouquet B_2 of two self-loops (i.e., the graph with a single vertex and two self-loops) is a minor of the dumbbell graph D (i.e., the graph consisting of an edge $[u, v]$ plus two self-loops on u and v, respectively). However, it is easy to verify that $\gamma_{\max}(B_2) = 1$ while $\gamma_{\max}(D) = 0$. Thus, Kuratowski-type theorems for maximum genus cannot be derived from Fact 61.

DEFINITIONS

D37: Let G be a graph and let v be a degree-2 vertex with two neighbors u and w in G (u and w could be the same vertex). We say that a graph G' is obtained from G by ***smoothing*** the vertex v if G' is constructed from G by removing the vertex v then adding a new edge connecting the vertices u and w.

D38: Two graphs G_1 and G_2 are ***homeomorphic*** if they become isomorphic after smoothing all degree-2 vertices. It is easy to see that two homeomorphic graphs have the same minimum genus and the same maximum genus.

FACTS

F65: A 2-edge-connected graph G has maximum genus 0 if and only if G is a cycle.

F66: [NoStWh71] A graph G has maximum genus 0 if and only if no vertex is contained in two different cycles in G. Such a graph has been called a ***cactus***.

F67: [ChGr93] A 2-edge connected graph G has maximum genus 1 if and only if G is homeomorphic to either a necklace or one of the graphs in Figure 7.2.3.

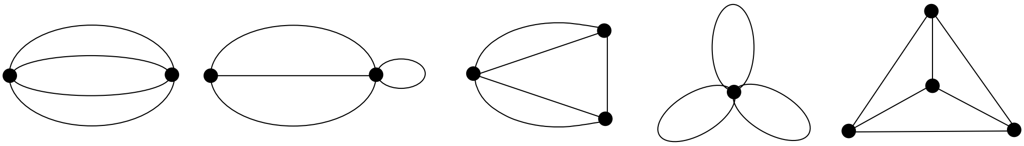


Figure 7.2.3 Graphs of maximum genus 1 that are not necklaces.

F68: A graph G has maximum genus 1 if and only if all but one of its 2-edge connected components are either a cycle or a single vertex, and the exceptional 2-edge connected component of G is homeomorphic either to a necklace or to one of the graphs in Figure 7.2.3.

7.2.5 Algorithmic Issues

Fellows and Langston [FeLa88] indicated that the graph minimum genus problem of graphs of bounded minimum genus can be solved in polynomial time based on Robertson and Seymour's results in graph minor theory. In fact, they showed a much stronger result that for any graph class $\mathcal{C}$ closed under minors, there is a polynomial time algorithm that tests the membership for the class $\mathcal{C}$.

Minimum Genus Testing

DEFINITION

D39: A graph G is an ***apex graph*** if it contains a vertex v such that $G - v$ is planar. Note that it is easy to test whether a given graph is apex in polynomial time.

FACTS

F69: [HoTa74] There is a linear-time algorithm that either constructs a planar imbedding for a given graph or reports that the graph is not planar.

F70: [HoWo74] There is a linear time algorithm that tests the isomorphism of planar graphs.

F71: [FiMiRe79] There is an algorithm to decide whether a graph has genus at most g with time complexity $O(|V|^{O(g)})$.

F72: [RoSe95] Let H be a fixed graph. There is a polynomial time algorithm that for a given graph G decides whether H is a minor of G.

F73: For any closed under minors graph class $\mathcal{C}$, there is a polynomial time algorithm that tests the membership for the class $\mathcal{C}$.

F74: [Mo99] For each fixed integer g, there is a linear-time algorithm that, for a given graph G, either constructs an imbedding of genus bounded by g for G or reports no such an imbedding exists. (This constructive result of Mohar significantly improves the corollary to Fact 73 that a polynomial-time algorithm exists.)

F75: [Th89] The following problem is NP-complete: given a graph G and an integer k, decide whether $\gamma_{\min}(G) \leq k$.

F76: [Th97a] The problem of deciding whether a graph of maximum degree 3 has its minimum genus bounded by a given integer k is NP-complete.

F77: [Mo98] The problem of deciding whether an apex graph has its minimum genus bounded by a given integer k is NP-complete.

Maximum Genus Testing

FACTS

F78: [FiGrMc88] There is a polynomial time algorithm that constructs a maximum genus imbedding for a given graph. (Direct calculation from the Xuong and Nebeský characterizations requires exponential time. The polynomial-time algorithm is based on a reduction to the linear matroid parity problem, which is solvable in polynomial time [GaSt85].)

F79: [Ch94] For any fixed integer g, there is a linear time algorithm that decides whether a given graph has maximum genus g, and if so, the algorithm constructs a maximum genus imbedding for the graph.

F80: [Ch94] For any fixed integer g, there is a linear time isomorphism algorithm for graphs of maximum genus bounded by g.

F81: [GrRi91] Starting from any imbedding of a graph, there is a sequence of edge deletion-then-reinsertion operations that never decreases the imbedding genus and eventually leads to a maximum genus imbedding of the graph. Thus, there are no graph imbeddings that are “strictly locally maximal” but not globally maximum for imbedding genus.

F82: [GrRi91] There exist “strictly locally minimal” graph imbeddings that are not minimum genus imbeddings, that are traps of arbitrarily great depth, serving as obstructions.

REMARK

R6: Facts 81 and 82 together explain the difference in complexity of the calculations of minimum genus and maximum genus.

References

[ApHa76] K. Appel and W. Haken, Every planar map is four-colorable, *Bull. Amer. Math. Soc.* 82 (1976), 711–712.

[Ar81] D. Archdeacon, A Kuratowski theorem for the projective plane, *J. Graph Theory* 5 (1981), 243–246.

[Ar+02] D. Archdeacon, J. Chen, Y. Huang, S. P. Kanchi, D. Li, Y. Liu, R. Nedela, and M. Škoviera, Maximum genus, connectivity, and Nebeský's theorem, *Discrete Mathematics*, to appear.

[ArHu89] D. Archdeacon and P. Huneke, A Kuratowski theorem for nonorientable surfaces, *J. Combin. Theory Ser. B* 46 (1989), 173–231.

[BHKY62] J. Battle, F. Harary, Y. Kodama, J. W. Y. Youngs, Additivity of the genus of a graph, *Bull. Amer. Math. Soc.* 68 (1962), 565–568.

[Br32] H. R. Brahana, Systems of circuits of two-dimensional manifolds, *Ann. of Math.* 30 (1923), 234–243.

[Ch90] J. Chen, *The distribution of graph imbeddings on topological surfaces*, Ph.D. thesis, Department of Mathematics, Columbia University, (1990).

[Ch94] J. Chen, A linear time algorithm for isomorphism of graphs of bounded average genus, *SIAM J. Discrete Mathematics* 7 (1994), 614–631.

[ChArGr96] J. Chen, D. Archdeacon, and J. L. Gross, Maximum genus and connectivity, *Discrete Mathematics* 149 (1996), 19–29.

[ChGr93] J. Chen and J. L. Gross, Kuratowski-type theorem for average genus, *J. Combin. Theory Ser. B* 57 (1993), 100–121.

[ChGr95] J. Chen and J. L. Gross, No lower limit points for average genus, *Graph Theory, Combinatorics, and Applications*, ed. by Y. Alavi and A. Schwenk, Wiley-Interscience (1995), 183–194.

[ChKa99] J. Chen and S. P. Kanchi, Graph ear decompositions and graph embeddings, *SIAM J. Discrete Math.* 12(2) (1999), 229–242.

[ChKaGr96] J. Chen, S. P. Kanchi, and J. L. Gross, A tight lower bound on the maximum genus of a simplicial graph, *Discrete Mathematics* 156 (1996), 83–102.

[Fa48] I. Fary, On straight line representation of planar graphs, *Acta Sci. Math. (Szeged)* 11 (1948), 229–233.

[FeLa88] M. Fellows and M. Langston, Nonconstructive tools for proving polynomial-time decidability, *J. Assoc. Comput. Mach.* 35(3) (1988), 727–739.

[FiMiRe79] L. Filotti, G. Miller, and J. Reif, On determining the genus of a graph in $O(v^{O(g)})$ steps, *Proc. 11th Annu. ACM Symp. on Theory of Computing* (1979), 27–37.

[FuGrMc88] M. Furst, J. L. Gross, and L. A. McGeoch, Finding a maximum-genus graph imbedding, *J. Assoc. Comput. Mach.* 35(3) (1988), 523–534.

[GaSt85] H. N. Gabow and M. Stallmann, Efficient algorithms for graphic matroid intersection and parity, *Lecture Notes in Computer Science* 194 (1985), 210–220.

[GlHu78] H. Glover, and J. P. Huneke, The set of irreducible graphs for the projective plane is finite, *Discrete Math.* 22 (1978), 243–256.

[GlHuWa79] H. Glover, J. P. Huneke, and C.-S. Wang, 103 graphs that are irreducible for the projective plane, *J. Combin. Theory Ser. B* 27 (1979), 332–370.

[GKR93] J. L. Gross, E. W. Klein, and R. G. Rieper, On the average genus of a graph, *Graphs and Combinatorics* 9 (1993), 153–162.

[GrRi91] J. L. Gross and R. G. Rieper, Local extrema in genus-stratified graphs, *J. Graph Theory* 15 (1991), 159–171.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover, 2001. First Edition, Wiley-Interscience, 1987.

[Heaw90] P. J. Heawood, Map-colour theorem, *Quart. J. Math.* 24 (1890), 332–338.

[HoTa74] J. Hopcroft and R. Tarjan, Efficient planarity testing, *J. Assoc. Comput. Mach.* 21 (1974), 549–568.

[HoWo74] J. Hopcroft and J. Wong, Linear time algorithm for isomorphism of planar graphs, *Proc. 6th Annual ACM Symp. on Theory of Computing* (1974), 172–184.

[HuLi00a] Y. Huang and Y. Liu, The classes of upper embeddable graphs with the same value of degree of vertex under modulo, *Acta Math. Sci.* 20 (2000), 251–255.

[HuLi00b] Y. Huang and Y. Liu, Maximum genus, independent number and girth, *Chinese Annals of Mathematics* 21, (2000), 77–82.

[Ju78] M. Jungerman, A characterization of upper-embeddable graphs, *Trans. Amer. Math. Soc.* 241 (1978), 401–406.

[Ko36] D. König, Theorie der endlichen und unendlichen Graphen, *Akademische Verlagsgesellschaft* (1936).

[Ku74] S. Kundu, Bounds on the number of disjoint spanning trees, *J. Combin. Theory Ser. B* 17 (1974), 199–203.

[Ku30] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fund. Math.* 15 (1930), 271–283.

[LiLi00] D. Li and Y. Liu, Maximum genus, girth and connectivity, *European J. Combin.* 21 (2000), 651–657.

[Mo98] B. Mohar, On the orientable genus of graphs with bounded nonorientable genus, *Discrete Math.* 182 (1998), 245–253.

[Mo99] B. Mohar, A linear time algorithm for embedding graphs in an arbitrary surface, *SIAM J. Discrete Math.* 12 (1999), 6–26.

[Ne81a] L. Nebeský, A new characterization of the maximum genus of a graph, *Czechoslovak Math. J.* 31 (1981), 604–613.

[Ne81b] L. Nebeský, Every connected, locally connected graph is upper embeddable, *J. Graph Theory* 5 (1981), 205–207.

[NoStWh71] E. Nordhaus, B. Stewart, and A. White, On the maximum genus of a graph, *J. Combin. Theory Ser. B* 11 (1971), 258–267.

[PaXu79] C. Payan and N. H. Xuong, Upper embeddability and connectivity of graphs, *Discrete Mathematics* 27 (1979), 71–80.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Nat. Acad. Sci. U.S.A.* 60 (1968), 438–445.

[RoSe85] N. Robertson and P. D. Seymour, Graph minors — a survey, pp. 153–171 in *Surveys in Combinatorics 1985*, ed. by I. Anderson, Cambridge Univ. Press, 1985.

[RoSe88] N. Robertson and P. D. Seymour, Graph minors XX. Wagner's conjecture, *preprint*, (1988).

[RoSe90a] N. Robertson and P. D. Seymour, Graph minors IV. Tree-width and well-quasi-ordering, *J. Combin. Theory Ser. B* 48 (1990), 227–254.

[RoSe90b] N. Robertson and P. D. Seymour, Graph minors VIII. A Kuratowski theorem for general surfaces, *J. Combin. Theory Ser. B* 48 (1990), 255–288.

[RoSe95] N. Robertson and P. D. Seymour, Graph minors XIII. The disjoint paths problem, *J. Combin. Theory Ser. B* 63 (1995), 65–110.

[Se93] P. D. Seymour, A bound on the excluded minors for a surface, unpublished manuscript, 1993.

[Sk91] M. Škoviera, The maximum genus of graphs of diameter two, *Discrete Mathematics* 87 (1991), 175–180.

[SkNe89] M. Škoviera and R. Nedela, The maximum genus of vertex-transitive graphs, *Discrete Mathematics* 78 (1989), 179–186.

[Th89] C. Thomassen, The graph genus problem is NP-complete, *J. Algorithms* (10) (1989), 568–576.

[Th97a] C. Thomassen, The genus problem for cubic graphs, *J. Combin. Theory Ser. B* 69 (1997), 52–58.

[Th97b] C. Thomassen, A simpler proof of the excluded minor theorem for higher surfaces, *J. Combin. Theory Ser. B* 70 (1997), 306–311.

[Tu60] W. Tutte, Convex representation of graphs, *Proc. London Math. Soc.* 10 (1960), 474–483.

[Wa37] K. Wagner, Über eine Eigenschaft der ebenen Komplexe, *Math. Ann.* 114 (1937), 570–590.

[Wh32] H. Whitney, Non-separable and planar graphs, *Trans. Amer. Math. Soc.* 34 (1932), 339–362.

[Wh33] H. Whitney, A set of topological invariants for graphs, *Amer. J. Math.* 55 (1933), 231–235.

[Xu79a] N. H. Xuong, How to determine the maximum genus of a graph, *J. Combin. Theory Ser. B* 26 (1979), 217–225.

[Xu79b] N. H. Xuong, Upper-embeddable graphs and related topics, *J. Combin. Theory Ser. B* 26 (1979), 226–232.

7.3 GENUS DISTRIBUTIONS

Jonathan L. Gross, Columbia University

7.3.1 Ranges and Distributions of Imbeddings
7.3.2 Counting Noncellular Imbeddings
7.3.3 Genus Distribution Formulas for Special Classes
7.3.4 Other Imbedding Distribution Calculations
7.3.5 The Unimodality Problem
7.3.6 Average Genus
7.3.7 Stratification of Imbeddings
References

Introduction

This chapter explores the natural problem of constructing surface-by-surface inventories of the imbeddings of a fixed graph, which was introduced by Gross and Furst [GrFu87]. The present scope includes several interesting extensions of that problem.

An imbedding is taken to be cellular, unless it is clear from context that a noncellular imbedding is under consideration. We shall regard two cellular imbeddings as "the same" if they have equivalent rotation systems. Moreover, a graph is taken to be connected unless the context implies otherwise.

7.3.1 Ranges and Distributions of Imbeddings

DEFINITIONS

D1: The ***minimum genus of a graph*** G is the smallest positive integer g such that the graph G has an imbedding in the orientable surface S_g. It is denoted $\gamma_{min}(G)$.

D2: The ***maximum genus of a graph*** G is the largest integer g such that the graph G has a cellular imbedding in the orientable surface S_g. It is denoted $\gamma_{max}(G)$.

D3: The ***genus range of a graph*** G, is the integer interval $[\gamma_{min}(G), \gamma_{max}(G)]$.

D4: The $j^{\underline{th}}$ ***orientable imbedding number of a graph*** G, denoted $\gamma_j(G)$, is the number of equivalence classes of orientable imbeddings of G into the orientable surface S_j, or equivalently (see §7.1), the number of rotation systems for graph G that induce an imbedding in S_j.

D5: The ***genus distribution sequence*** of a graph G is the sequence whose $j^{\underline{th}}$ entry is $\gamma_j(G)$, starting with a (possibly empty) subsequence of zeroes, followed by the subsequence of the orientable imbedding numbers, and then an infinite sequence of zeroes.

D6: The ***genus distribution polynomial*** is

$$I_G(x) = \sum_{j=0}^{\infty} \gamma_j(G)x^j$$

D7: The ***minimum crosscap number of a graph*** G, also known as the ***minimum nonorientable genus*** is the smallest integer k such that the graph G has an imbedding in the nonorientable surface N_k. It is denoted $\overline{\gamma}_{min}(G)$.

D8: The ***maximum crosscap number of a graph*** G, also known as the ***maximum nonorientable genus*** is the largest integer k such that the graph G has a cellular imbedding in the nonorientable surface N_k. It is denoted $\overline{\gamma}_{max}(G)$.

D9: The ***crosscap range of a graph*** G, is the integer interval $[\overline{\gamma}_{min}(G), \overline{\gamma}_{max}(G)]$.

D10: The $j^{\underline{th}}$ ***crosscap imbedding number of a graph*** G, denoted $\overline{\gamma}_j(G)$, is the number of equivalence classes of nonorientable imbeddings of G into the nonorientable surface N_j.

D11: The ***crosscap distribution sequence*** of a graph G is the sequence whose $j^{\underline{th}}$ entry is $\overline{\gamma}_j(G)$, starting with a (possibly empty) subsequence of zeroes, followed by the subsequence of the crosscap imbedding numbers, and then an infinite sequence of zeroes.

D12: The ***crosscap distribution polynomial*** is

$$\overline{I}_G(y) = \sum_{j=1}^{\infty} \overline{\gamma}_j(G)y^j$$

D13: A ***bar-amalgamation*** of two disjoint graphs G and H is obtained by running a new edge e between a vertex u of G and a vertex v of H. Notation: $G_u *_e H_v$.

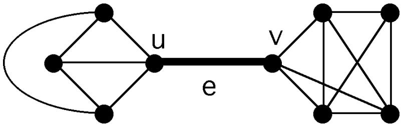


Figure 7.3.1 A bar-amalgamation of K_4 and $K_5 - e$.

D14: A ***vertex-amalgamation*** of two disjoint graphs G and H is obtained by identifying a vertex u of G and a vertex v of H. Notation: $G_u * H_v$.

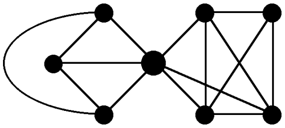


Figure 7.3.2 A vertex-amalgamation of K_4 and $K_5 - e$.

D15: The ***convolution*** of the sequences $\langle a_i \rangle$ and $\langle b_i \rangle$ is the sequence whose k^{th} term is $\sum_{i=0}^{k} a_i b_{k-i}$.

FACTS

F1: [Du66] For every integer j within the genus range of a graph G, i.e., whenever $\gamma_{min}(G) \leq j \leq \gamma_{max}(G)$, the number $\gamma_j(G)$ of orientable imbeddings of G is positive.

F2: Let G be a graph. Then the total number of equivalence classes of orientable imbeddings equals

$$\sum_{j=0}^{\infty} \gamma_j(G) = \prod_{v \in V(G)} [\deg(v) - 1]!$$

since the sum on the left and the product on the right both count every imbedding of G exactly once. Moreover, the polynomial evaluation $I_G(1)$ gives this same number.

F3: [St78] For every integer j within the crosscap range of a graph G, i.e., whenever $\overline{\gamma}_{min}(G) \leq j \leq \overline{\gamma}_{max}(G)$, the number $\gamma_j(G)$ of nonorientable imbeddings of G is positive.

F4: Let G be a graph. Then the total number of equivalence classes of imbeddings (orientable and nonorientable) equals

$$\sum_{j=0}^{\infty} \gamma_j(G) + \sum_{j=1}^{\infty} \overline{\gamma}_j(G) \quad = \quad 2^{\beta(G)} \prod_{v \in V(G)} [\deg(v) - 1]!$$

since the sum on the left and the product on the right both count every imbedding of G exactly once. The factor of $2^{\beta(G)}$ on the right accounts for the possible choices of orientation on every edge not in a designated spanning tree for G.

F5: [GrFu87] The genus distribution of a bar-amalgamation of two graphs is representable as the convolution of their respective genus distributions, multiplied by a scalar equal to the product of the degrees of the vertices at the ends of the bar.

F6: The minimum genus of a vertex-amalgamation $G_u * H_v$ of two graphs is always equal to the minimum genus of the bar-amalgamation $G_u *_e H_v$. That is,

$$\gamma_{min}(G_u * H_v) \ = \ \gamma_{min}(G_u *_e H_v)$$

F7: The maximum genus of a vertex-amalgamation $G_u * H_v$ of two graphs is either equal to the maximum genus of the bar-amalgamation $G_u *_e H_v$ or larger by one. That is,

$$\gamma_{max}(G_u *_e H_v) \ \leq \ \gamma_{max}(G_u * H_v) \ \leq \ \gamma_{max}(G_u *_e H_v) + 1$$

F8: Every term of the genus distribution sequence of a vertex-amalgamation $G_u * H_v$ of two graphs is at least as large as the corresponding term of the bar-amalgamation $G_u *_e H_v$.

EXAMPLES

All of the following examples can be calculated by considering the rotation systems (see §7.1). Consideration of symmetries expedites the calculations.

E1: The graph K_4 has the following genus distribution sequence:

$$2, 14, 0, 0, \ldots$$

E2: The bouquet B_2 has the following genus distribution sequence:

$$4, 2, 0, 0, \ldots$$

E3: The dipole D_3 has the following genus distribution sequence:

$$2, 2, 0, 0, \ldots$$

E4: The complete bipartite graph $K_{3,3}$ has the following genus distribution sequence:

$$0, 40, 24, 0, 0, \ldots$$

E5: The bar-amalgamation $K_4 *_e K_{3,3}$ has the following genus distribution sequence:

$$12(0, 80, 608, 336, 0, 0, \ldots)$$

E6: The bar-amalgamation $K_3 *_e K_3$ has the genus distribution sequence

$$(4, 0, 0, 0, \ldots)$$

and the vertex-amalgamation $K_3 * K_3$ has the genus distribution sequence

$$(4, 2, 0, 0, \ldots)$$

RESEARCH PROBLEM

Research Problem: Construct a general method for calculating the distribution of the vertex amalgamation of two graphs.

REMARK

R1: Complementary to the problems of counting imbeddings of a given graph, over all surfaces, are the problems of counting maps on a given surface, over all possible imbedded graphs. See §7.6.

7.3.2 Counting Noncellular Imbeddings

This section describes how the problem of calculating distributions of noncellular imbeddings reduces to counting cellular imbeddings. These methods seem to be overlooked in the literature.

DEFINITIONS

D16: A ***semicellular graph imbedding*** is an imbedding $G \to S$ whose regions are planar, but which may have more than one boundary component.

D17: A graph imbedding $G \to S$ is ***strongly noncellular*** if any of its regions is nonplanar.

D18: A closed curve that separates a region of a noncellular graph imbedding $G \to S$ is ***boundary-separating*** if there is at least one boundary component of the region on each side of the separation.

D19: A closed curve in a region of a noncellular graph imbedding $G \to S$ is ***strongly noncontractible*** if it is noncontractible in every surface in which that region can be imbedded. Equivalently, cutting it open reduces the genus of the region.

D20: Given a semicellular graph imbedding $G \to S$, the ***underlying cellular imbedding*** is obtained by cutting each non-cell region open along a maximal family of boundary-separating closed curves and capping the holes with disks.

D21: Given a noncellular graph imbedding $G \to S$, a ***planarizing curve*** for a nonplanar region is a separating closed curve such that all of the boundary components lie to one side of the separation and all of the genus lies to the other.

D22: Given a strongly noncellular graph imbedding $G \to S$, the ***underlying semicellular imbedding*** is obtained by cutting each non-cell region open along a maximal family of boundary-separating closed curves and capping the holes with disks.

FACTS

F9: Every semicellular graph imbedding has an underlying cellular imbedding that is unique up to homeomorphism.

F10: The semicellular orientable imbeddings of a graph are in bijective correspondence with partitions of the regions of the underlying cellular imbedding.

F11: Every nonplanar region of a noncellular graph imbedding has a planarizing curve.

F12: Every strongly noncontractible imbedding has an underlying semicellular imbedding that is unique up to homeomorphism.

F13: The strongly noncontractible imbeddings of a graph are in bijective correspondence with the set of functions from the regions of the underlying semicellular imbedding to the nonnegative integers.

F14: A strongly noncellular imbedding $G \to S_{n+k}$ of a graph in a surface can be obtained from a semicellular imbedding into $G \to S_n$ by partitioning the number k, and next selecting one face of the imbedding into S_n for each of the parts of the partition, and then increasing the genus of each selected face by the value of the associated part of the partition.

EXAMPLES

E7: The graph $K_2 \times C_3$ has six vertices, each of degree 3. Thus, the total number of orientable cellular imbeddings is $64 = 2^6$. The cellular genus distribution sequence is $2, 38, 24, 0, 0, \ldots$.

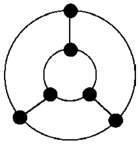


Figure 7.3.3 The graph $K_2 \times C_3$.

E8: Each of the two imbeddings of $K_2 \times C_3$ in S_0 has five faces. Five faces can be partitioned into four nonempty parts in five ways. (In general, one can use *Stirling subset numbers* for these partition-number calculations.) Thus, each imbedding in S_0 yields five different possible semicellular imbeddings in S_1. Thus, there are ten semicellular noncellular imbeddings of $K_2 \times C_3$ in S_1, plus 38 cellular imbeddings, as menetioned in Example 7, for a total of 38 semicellular imbeddings in S_1.

E9: Each semicellular imbedding of $K_2 \times C_3$ in the surface S_2 corresponds to a partition into three parts of the five faces of a cellular imbedding in S_0 or to a partition into two parts of the three faces of a cellular imbedding into S_1. Using the Stirling numbers $\left\{ {5 \atop 3} \right\} = 25$ and $\left\{ {3 \atop 2} \right\} = 3$, and the cellular genus distribution sequence from Example 7, we calculate that the number of semicellular (but noncellular) imbeddings into S_2 equals $2 \cdot 25 + 38 \cdot 3 = 164$. Adding in the 24 cellular imbeddings in S_2, we obtain a total of 188.

7.3.3 Genus Distribution Formulas for Special Classes

Even at the outset of the program to provide explicit calculations of imbedding distributions, it was clear that a variety of techniques would be needed. Different topological and combinatorial methods seem to be needed for every class of graphs.

DEFINITIONS

D23: The ***n*-rung closed-end ladder** L_n is the graph obtained from the cartesian product $P_n \times K_2$ by doubling the edges $v_1 \times K_2$ and $v_n \times K_2$ at both ends of the path, as illustrated in Figure 7.3.4.

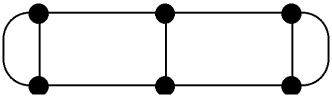


Figure 7.3.4 **The 3-rung closed-end ladder L_3.**

D24: The ***cobblestone path*** J_n is the graph obtained by doubling every edge of the n-vertex path P_n, as illustrated in Figure 7.3.5.

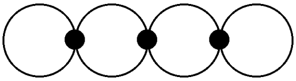


Figure 7.3.5 **The cobblestone path J_3.**

D25: The ***n*-bouquet** B_n is the graph with one vertex and n self-loops, as illustrated in Figure 7.3.6.

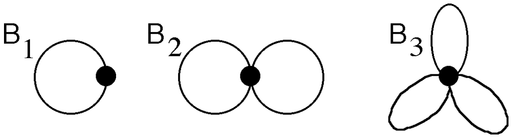


Figure 7.3.6 Some bouquets.

FACTS

F15: [FuGrSt89] The closed-end ladders have the following formula for their orientable imbedding numbers:

$$\gamma_i\left(L_n\right) = \begin{cases} 2^{n-1+i}\dbinom{n+1-i}{i}\frac{2n+2-3i}{n+1-i} & \text{for } i \leq \lfloor \frac{n+1}{2} \rfloor \\ 0 \quad \text{otherwise} & \end{cases}$$

The following table shows the genus distributions for some of the smaller ladders.

	γ_0	γ_1	γ_2	γ_3	γ_4	total
L_1	2	2	0	0	0	4
L_2	4	12	0	0	0	16
L_3	8	40	16	0	0	64
L_4	16	112	128	0	0	256
L_5	32	288	576	128	0	1024

F16: [FuGrSt89] The cobblestone paths have the following formula for their orientable imbedding numbers:

$$\gamma_i\left(J_n\right) = 3^i \cdot 4^{n-1} \cdot \binom{n-i}{i} + 2 \cdot 3^{i-1} \cdot 4^{n-1} \cdot \binom{n-i}{i-1} \qquad \text{for } i \geq 0 \text{ and } n \geq 1$$

The following table shows the genus distribution for some of the smaller cobblestone paths.

	γ_0	γ_1	γ_2	total
J_1	4	2	0	6
J_2	16	20	0	36
J_3	64	128	24	216
J_4	256	704	336	1296

F17: [GrRoTu89] The bouquets have the following formula for their orientable imbedding numbers:

$$\gamma_g\left(B_n\right) = (n-1)! \cdot 2^{n-1} \cdot e_{n-2g+1}\left(n\right)$$

where the numbers

$$e_k\left(n\right) = \#\left\{ \pi \in \Sigma_{2n} \;\middle|\; \begin{array}{l} \text{permutation } \pi \text{ has } k \text{ cycles, and} \\ (\exists \text{ full involution } \beta)\left[\pi = \rho_0 \circ \beta\right] \end{array} \right.$$

where ρ_0 is an arbitrary fixed cycle of length $2n$

are given by the formula of Jackson [Ja87]. The closed formula above for $\gamma_g\,(B_n)$ leads to the following recursion.

Initial conditions:

$$\gamma_g(B_g) = 0 \text{ for } g < 0 \text{ or } n < 0$$
$$\gamma_g(B_0) = \gamma_g(B_1) = 0 \text{ for } g > 0$$
$$\gamma_g(B_2) = \begin{cases} 4 & \text{for } g = 0 \\ 2 & \text{for } g = 1 \\ 0 & \text{for } g \geq 2 \end{cases}$$

Recursion for $n > 2$:

$$\begin{aligned}(n+1)\gamma_g(B_n) = {} & 4(2n-1)(2n-3)(n-1)^2(n-2)\gamma_g(B_{n-2}) \\ & + 4(2n-1)(n-1)\gamma_g(B_{n-1})\end{aligned}$$

This recursion enables us to calculate numerical values.

$\gamma_g\,(B_n)$	$g = 0$	1	2	Total
$n = 0$	1			1
1	1			$1! = 1$
2	4	2		$3! = 6$
3	40	80		$5! = 120$
4	672	3360	1008	$7! = 5040$
5	16128	161280	185472	$9! = 362880$

REMARKS

R2: Ladder-like graphs played a crucial role in the solution of the Heawood map-coloring problem. (See [Ri74].) McGeoch [McG87] calculated the genus distribution of "circular ladders" and of "Möbius ladders". Tesar [Te00] calculated the genus distribution of "Ringel ladders".

R3: The computations of imbedding distributions of ladders and cobblestone paths were subsequently generalized by Stahl [St91] to "linear families".

R4: Rieper [Ri90] elaborated upon the use of group characters in his analysis of the genus distribution of "dipoles", which are graphs with two vertices and no self-loops. Andrews, Jackson, and Visentin [AnJaVi94] took a map-theoretic approach to dipole imbeddings.

R5: Stahl [St91a] calculated genus distributions for small-diameter graphs.

R6: Kwak, Kim, and Lee [KwKiLe96] took a distributional approach in studying a class of branched coverings of surfaces.

R7: Riskin [Ri95] took a distributional approach in studying a class of polyhedral imbeddings.

R8: Stahl [St97] studied the zeroes of a class of genus polynomials.

R9: Among the important properties of bouquets to topological graph theory is that every regular graph can be derived by assigning voltages (possibly permutation voltages) to a bouquet. (See [GrTu77] or [GrTu87].)

7.3.4 Other Imbedding Distribution Calculations

Calculations of total imbedding distributions, including nonorientable surfaces require some additional theory, partly because the possible twisting of edges complicates the recurrences one might derive. Yet another enumerative aspect of graph imbeddings regards as equivalent any two imbeddings that "look alike" when vertex and edge labels are removed.

DEFINITIONS

D26: The ***total imbedding distribution*** of a graph G is the bivariate polynomial

$$\ddot{I}_G(x,y) = I_G(x) + \overline{I}_G(y) = \sum_{j=0}^{\infty} \gamma_j(G)x^j + \sum_{j=1}^{\infty} \overline{\gamma}_j(G)y^j$$

D27: Given a general rotation system ρ for a graph G and a spanning tree T, the entries of the ***overlap matrix*** $M_{\rho,T} = [m_{i,j}]$ are given for all pairs of edges e_i, e_j of the cotree $G - T$ by

$$m_{i,j} = \begin{cases} 1 & \text{if } i \neq j \text{ and } pure\,(\rho)\,|_{T+e_i+e_j} \text{ is nonplanar} \\ 1 & \text{if } i = j \text{ and edge } i \text{ is twisted} \\ 0 & \text{otherwise} \end{cases}$$

The notation $pure(\rho)|_{T+e_i+e_j}$ means the restriction of the underlying pure part of the rotation system ρ to the subgraph $T + e_i + e_j$.

D28: The imbeddings $\iota_1 : G \to S$ and $\iota_2 : G \to S$ are ***congruent*** if there exist a graph automorphism $\alpha : G \to G$ and a surface homeomorphism $h : S \to S$ such that the diagram in Figure 7.3.7 is commutative. We write $\iota_1 \simeq \iota_2$.

$$\begin{tikzcd} G \arrow[r, "\iota_1"] \arrow[d, "\alpha"'] & S \arrow[d, "h"] \\ G \arrow[r, "\iota_2"'] & S \end{tikzcd}$$

Figure 7.3.7 **Commutativity condition for imbedding congruence $\iota_1 \simeq \iota_2$.**

FACTS

F18: When non-orientable imbeddings of a graph G are also to be considered, the total number of imbeddings increases by a factor of $2^{\beta(G)-1}$, since each of the $\beta(G)$ edges in the complement of a spanning tree may be twisted or untwisted.

$$\ddot{I}_G(1,1) \quad = \quad 2^{\beta(G)} \prod_{v \in V(G)} [\deg(v) - 1]!$$

F19: [Mo89] Let G be a graph, T a spanning tree of G, and ρ a general rotation system of G. Then

$$rank\,(M_{\rho,T}) = \begin{cases} 2g & \text{if } S\,(\rho) \cong S_g \text{ (induced surface orientable)} \\ k & \text{if } S\,(\rho) \cong N_k \text{ (induced surface non-orientable)} \end{cases}$$

Thus, the genus of the surface induced by a rotation system can be calculated without doing face-tracing.

F20: [ChGrRi94] Calculating the surface type by face-tracing requires $O(n)$ time for a graph with n edges, and calculating the rank of the overlap matrix deteriorates to $O(n^2)$ time. However, regrouping the total set of imbeddings according to rank of the overlap matrix sometimes facilitates calculation of the total imbedding distribution.

F21: [ChGrRi94] Closed-end ladders have the following total imbedding distribution polynomials.

$$\ddot{I}_{L_n}(x,y) = 2^n \sum_{i_1,\cdots,i_r>0}^{i_1+\cdots+i_r=n+1} y^{n+1-r} \prod_{h=1}^{r} \left[round\left(\frac{2^{i_h}}{3}\right) + round\left(\frac{2^{i_h+1}}{3}\right) y \right]$$

$$- I_{L_n}\left(y^2\right) + I_{L_n}(x)$$

F22: [ChGrRi94] Cobblestone paths have the following total imbedding distribution polynomials.

$$\ddot{I}_{J_n}(x,y) = 2^n \sum_{i_1,\cdots,i_r>0}^{i_1+\cdots+i_r=n+1} 2^{n+r-1} y^{n+1-r} \prod_{h=1}^{r} \left[round\left(\frac{2^{i_h}}{3}\right) + round\left(\frac{2^{i_h+1}}{3}\right) y \right]$$

$$- I_{J_n}\left(y^2\right) + I_{J_n}(x)$$

REMARKS

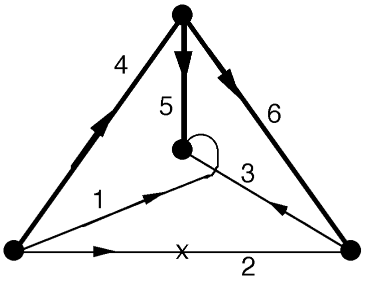
R10: [MuRiWh88] counted congruence classes of imbeddings of K_n into oriented surfaces. The key to counting congruence classes was to convert the cycle index of $Aut(G)$ acting on V_G into the cycle index for the induced action on the rotation systems.

R11: [KwLe94] counted congruence classes of imbeddings into non-orientable surfaces. One of their underlying ideas is to regard an edge-twist as the voltage 1 (mod 2) and to construct the orientable double cover. Then the graph automorphisms act on the induced rotation systems.

R12: [KwSh02] developed a formula for the total imbedding distributions of bouquets.

EXAMPLES

E10: In the illustrative calculation of Figure 7.3.8, the spanning tree has edges 4, 5, and 6. Thus, the rows and columns correspond to cotree edges 1, 2, and 3. Since the rank of the matrix is 3 and the imbedding is non-orientable, the imbedding surface must be N_3 (by Fact 19).

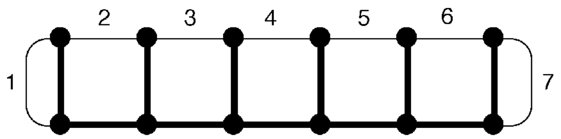


$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

overlap matrix

Figure 7.3.8 **Sample calculation of the overlap matrix.**

E11: In deriving Fact 21, [ChGrRi94] chose a tree T in the ladder graph with a path as a cotree, as in Figure 7.3.9. This yielded a "tridiagonal" overlap matrix, which is a convenient property in rank calculations.



$$\begin{matrix} \mathrm{x} & \mathrm{x} & 0 & 0 & 0 & 0 & 0 \\ \mathrm{x} & \mathrm{x} & \mathrm{x} & 0 & 0 & 0 & 0 \\ 0 & \mathrm{x} & \mathrm{x} & \mathrm{x} & 0 & 0 & 0 \\ 0 & 0 & \mathrm{x} & \mathrm{x} & \mathrm{x} & 0 & 0 \\ 0 & 0 & 0 & \mathrm{x} & \mathrm{x} & \mathrm{x} & 0 \\ 0 & 0 & 0 & 0 & \mathrm{x} & \mathrm{x} & \mathrm{x} \\ 0 & 0 & 0 & 0 & 0 & \mathrm{x} & \mathrm{x} \end{matrix}$$

Figure 7.3.9 **Ladder L_6, spanning tree, and tridiagonal overlap matrix.**

E12: In deriving Fact 22, [ChGrRi94] chose a tree T in the cobblestone path again with a path as a cotree, as in Figure 7.3.10. This again yielded a "tridiagonal" overlap matrix.

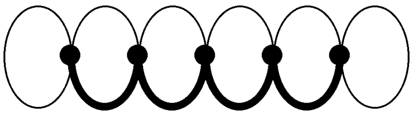
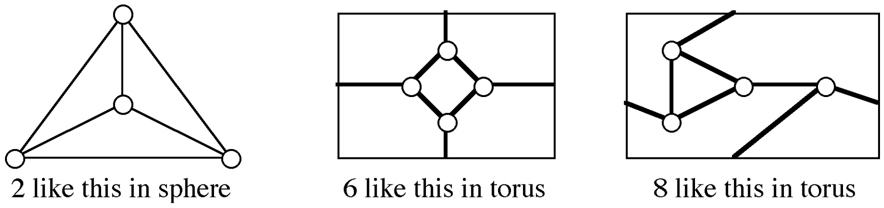


Figure 7.3.10 **Cobblestone path J_5 and spanning tree.**

E13: In regard to Remark 7, Figure 7.3.11 shows how the 16 different orientable imbeddings of the complete graph K_4 are partitioned into congruence classes.


2 like this in sphere

6 like this in torus

8 like this in torus

Figure 7.3.11 **Partitioning the 16 imbeddings of K_4 into congruence classes.**

Burnside's Lemma is used to count congruence classes. Each automorphism on a graph G induces a permutation on the rotation systems of G that preserves the congruence class, but does not necessarily preserve the (oriented homeomorphism of pairs)-equivalence class, as illustrated in Figure 7.3.12.

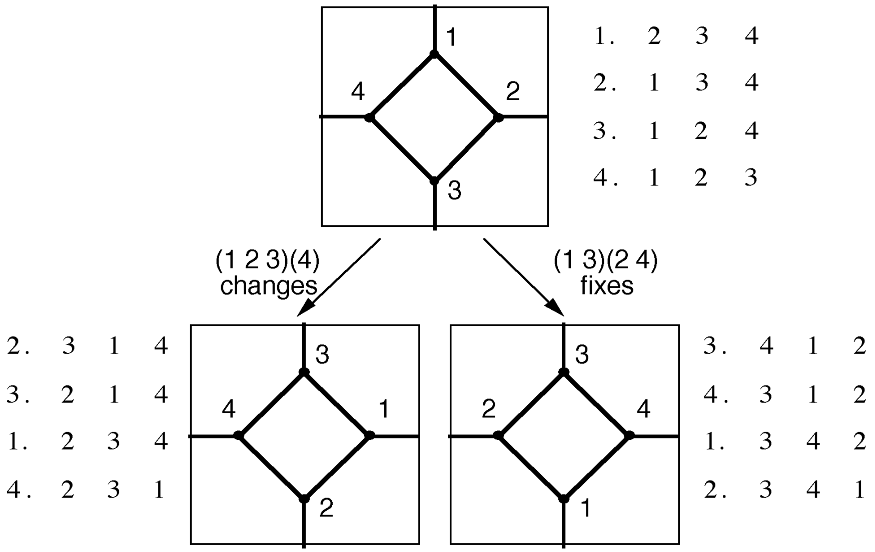


Figure 7.3.12 The induced action of a permutation on rotation systems.

7.3.5 The Unimodality Problem

DEFINITIONS

D29: A sequence $\{a_m\}$ is ***unimodal*** if there exists at least one integer M such that

$$\begin{aligned} a_{m-1} &\le a_m \quad \text{for all } m \le M \text{ and} \\ a_m &\ge a_{m+1} \quad \text{for all } m \ge M \end{aligned}$$

D30: A sequence $\{a_m\}$ is ***strongly unimodal*** if its convolution with any unimodal sequence yields a unimodal sequence.

FACTS

F23: A typical unimodal sequence first rises and then falls, as illustrated in Figure 7.3.13.

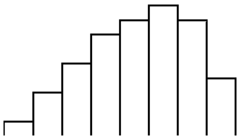


Figure 7.3.13 A unimodal sequence has no false maxima.

F24: [KeGe71] An equivalent criterion for unimodality is that

$$a_m^2 \ge a_{m+1}a_{m-1} \quad \text{for all } m$$

F25: [FuGrSt89] The genus distribution of every closed-end ladder graph is strongly unimodal.

F26: [FuGrSt89] The genus distribution of every cobblestone path is strongly unimodal.

F27: [GrRoTu89] The genus distribution of every bouquet is strongly unimodal.

REMARKS

R13: We observe that an imbedding of the bouquet B_n has $n+1$ faces if in the sphere S_0, $n-1$ faces if in the torus S_1, $n-3$ faces if in the surface S_2, and so on. Intuitively, this suggests that the genus distribution of the bouquet B_n might resemble the sequence of Stirling cycle numbers

$$\begin{bmatrix} 2n \\ n+1 \end{bmatrix}, \begin{bmatrix} 2n \\ n-1 \end{bmatrix}, \begin{bmatrix} 2n \\ n-3 \end{bmatrix}, \ldots$$

which is a strongly unimodal sequence.

R14: [St91a] The resemblence to Stirling numbers holds also for various graphs of small diameter, including partial suspensions of trees and of cycles.

R15: [St90] The genus distribution of the bouquet B_n is asymptotically proportional to this sequence. The proof uses group character theory.

RESEARCH PROBLEM

Research Problem: Decide whether the genus distribution of every graph is strongly unimodal.

7.3.6 Average Genus

DEFINITIONS

D31: The ***average genus*** of a graph G, denoted $\gamma_{\mathrm{avg}}(G)$, is the average value of the genus of the imbedding surface, taken over all orientable imbeddings.

D32: The ***cycle rank*** of a connected graph G, denoted $\beta(G)$, is the number $|E_G| - |V_G| + 1$; this is conceptually best understood as the number of edges in the cotree of a spanning tree for G.

D33: [GrKlRi93] A ***necklace*** of type (r,s) is obtained from a $2r+s$-cycle by doubling r disjoint edges and then adding a self-loop at each of the s vertices that is not an endpoint of a doubled edge.

D34: Let e be an edge of a graph. We say that we ***attach an open ear*** to the interior of edge e if we insert two new vertices u and v and then double the edge between them. The two new vertices are called the ***ends*** of that open ear.

D35: We ***attach a closed ear*** to the interior of edge e if we insert one new vertex w in its interior and then attach a self-loop at w. The vertex w is called the ***end*** of that closed ear.

D36: We say that r open ears and s closed ears are ***attached serially*** to the edge e if the ends of the ears are all distinct, and if no ear has an end between the two ends of an open ear.

GENERAL FACTS ABOUT AVERAGE GENUS

F28: [GrKlRi93] The average genus of a graph with nontrivial genus range can lie arbitrarily close to the maximum genus.

F29: [GrKlRi93] The average genus of a graph is at least as large as the average genus of any of its subgraphs.

F30: [ChGrRi95] For any 3-regular graph G,

$$\gamma_{\text{avg}}(G) \geq \frac{1}{2}\gamma_{\max}(G)$$

F31: [ChGrRi95] For any 2-connected simple graph G other than a cycle,

$$\gamma_{\text{avg}}(G) \geq \frac{1}{16}\beta(G)$$

F32: [Ch94] Isomorphism testing of graphs of bounded average genus can be achieved in linear time.

F33: [GrFu87] The average genus of the bar-amalgamation of two graphs G and H equals $\gamma_{\text{avg}}(G) + \gamma_{\text{avg}}(H)$.

F34: [ChGr92b] Let G be a 2-connected graph, and let G_+ be a graph obtained by serially attaching ears to an edge of G. Then

$$\gamma_{\text{avg}}(G) \;\leq\; \gamma_{\text{avg}}(G_+) \;\leq\; \gamma_{\text{avg}}(G) + 1$$

FACTS ABOUT SMALL VALUES OF AVERAGE GENUS

F35: A graph has average genus 0 if and only if at most one cycle passes through any vertex. This follows from [NoRiStWh72].

F36: The maximum genus of a necklace is 1. This follows from [Xu79].

F37: [GrKlRi93] The average genus of any necklace of type (r, s) is

$$1 - \left(\frac{1}{2}\right)^r \left(\frac{2}{3}\right)^s$$

F38: [GrKlRi93] Each of the six smallest possible values of average genus is realizable by a necklace. Figure 7.3.14 indicates these values and shows a graph realizing each of them.

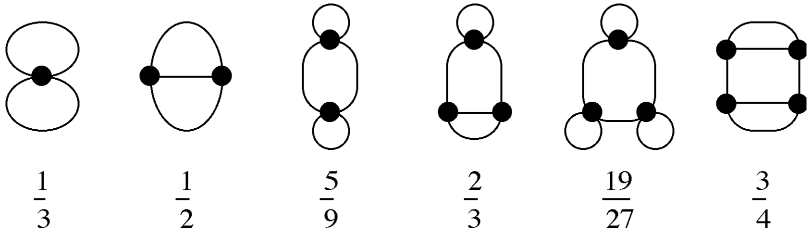


Figure 7.3.14 **Realizations of the six smallest positive values of average genus.**

F39: [ChGr93] Except for necklaces, there are exactly eight 2-connected graphs of average genus less than one. The bouquet B_3, the dipole D_4, and the complete graph K_4 have average genus

$$\frac{2}{3}, \quad \frac{5}{6}, \quad \frac{7}{8}$$

respectively. Figure 7.3.15 shows the other five such graphs and their average genus.

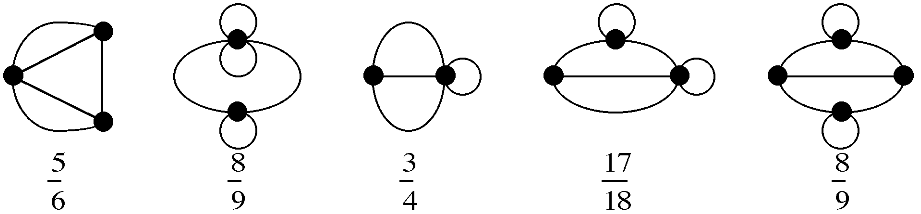


Figure 7.3.15 **Five sporadic 2-connected graphs with average genus less than one.**

F40: Facts 33 and 39 together yield a complete classification of all graphs of average genus less than one.

F41: [ChGr93] There are exactly three 2-connected graphs with average genus equal to 1.

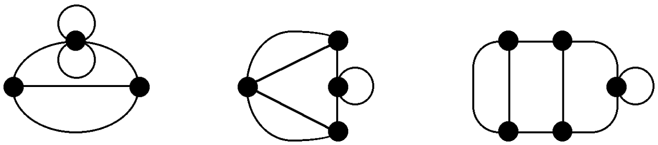


Figure 7.3.16 **The three 2-connected graphs with average genus equal to one.**

FACTS ABOUT CLOSE VALUES OF AVERAGE GENUS

F42: [GrKlRi93] Arbitrarily many mutually nonhomeomorphic 2-connected graphs can have the same average genus.

F43: [ChGr92a] For each real number r, only finitely many 3-connected graphs have average genus less than r.

F44: [ChGr92a] For each real number r, only finitely many 2-connected simple graphs have average genus less than r.

FACTS ABOUT LIMIT POINTS OF AVERAGE GENUS

F45: [GrKlRi93] The number 1 is an upper limit point of the set of possible values of average genus.

F46: [ChGr92a] The set of possible values of average genus for 3-connected graphs has no limit points.

F47: [ChGr92a] The set of possible values of average genus for 2-connected simple graphs has no limit points.

F48: [ChGr95] Lower limit points of average genus do not exist.

REMARKS

R16: Fact 34 provides a means for constructing upper limit points. In fact, all limit points arise from this construction.

R17: Additional results on average genus are given by [MaSt96], [Sc99], [St95a], and [St95b].

7.3.7 Stratification of Imbeddings

Superimposing an adjacency structure on the distribution of orientable imbeddings appears to offer some insight into the problem of deciding whether two given graphs are isomorphic.

DEFINITIONS

D37: Two orientable imbeddings ι_1 and ι_2 of the same graph G are ***V*-adjacent** if there is a vertex v of G such that moving a single edge-end at v is sufficient to transform a rotation system representing ι_1 into a rotation system representing ι_2.

D38: Two orientable graph imbeddings of the same graph G are ***E*-adjacent** if there is an edge e of G such that moving both edge-ends of e can transform a rotation system representing ι_1 into a rotation system representing ι_2.

D39: For any graph G, the ***stratified graph*** SG has as its points the orientable imbeddings of G. Its lines are the V-adjacencies and the E-adjacencies.

D40: The induced subgraph of SG on the set of imbeddings into the surface S_j is called the j^{th} ***stratum*** of SG and is denoted S_jG.

D41: A ***complete isomorphism invariant*** for a graph is a graph invariant that has a different value on each isomorphism type of graph.

FACTS

F49: The cardinality of S_jG is $\gamma_j(G)$.

F50: [GrTu79] There may be false minima in the stratified graph, that is, local minima that are not global minima.

F51: [GrRi91] The false minima may have arbitrarily great depth.

F52: [GrRi91] No false maxima exist, so that it is possible to ascend from any imbedding to a maximum imbedding, even though strict ascent might not always be possible.

F53: [GrTu95] For every vertex of the stratified graph SG, the induced subgraph on its set of neighbors in SG is a complete isomorphism invariant of the graph.

REMARKS

R18: This is consistent with [Th89], which proves that the minimum genus problem is NP-complete.

R19: This is consistent with [FuGrMcG88], which establishes a polynomial-time algorithm for maximum genus.

R20: [GrTu95] also demonstrated how two graphs with similar genus distributions may have markedly different imbedding strata. These findings support the plausibility of a probabilistic approach to graph isomorphism testing, based on the sampling of higher-order imbedding distribution data.

References

[AnJaVi94] G. E. Andrews, D. M. Jackson, and T. I. Visentin, A hypergeometric analysis of the genus series for a class of 2-cell embeddings in orientable surfaces, *SIAM J. Math. Anal.* 25 (1994), 243–255.

[Ch94] J. Chen, A linear-time algorithm for isomorphism of graphs of bounded average genus, *SIAM J. of Discrete Math.* 7 (1994), 614–631.

[ChGr92a] J. Chen and J. L. Gross, Limit points for average genus (I): 3-connected and 2-connected simplicial graphs, *J. Combin. Theory Ser. B* 55 (1992), 83–103.

[ChGr92b] J. Chen and J. L. Gross, Limit points for average genus (II): 2-connected non-simplicial graphs (with J. Chen), *J. Combin. Theory Ser. B* 56 (1992), 108–129.

[ChGr93] J. Chen and J. L. Gross, Kuratowski-type theorems for average genus, *J. Combin. Theory Ser. B* 57 (1993), 100–121.

[ChGrRi94] J. Chen, J. L. Gross, and R. G. Rieper, Overlap matrices and total imbedding distributions, *Discrete Math.* 128 (1994), 73–94.

[ChGr95] J. Chen and J. L. Gross, No lower limit points for average genus, *Graph Theory, Combinatorics, and Algorithms, Vol. 1 (Kalamazoo, MI, 1992)*, 183–194, Wiley-Interscience, New York, 1995.

[ChGrRi95] J. Chen, J. L. Gross, and R. G. Rieper, Lower bounds for the average genus, *J. Graph Theory* 19 (1995), 281–296.

[FuGrSt89] M. L. Furst, J. L. Gross, and R. Statman, Genus distribution for two classes of graphs, *J. Combin. Theory Ser. B* 46 (1989), 22–36.

[GrFu87] J. L. Gross and M. L. Furst, Hierarchy for imbedding-distribution invariants of a graph, *J. Graph Theory* 11 (1987), 205–220.

[GrRi91] J. L. Gross and R. G. Rieper, Local extrema in genus-stratified graphs, *J. Graph Theory* 15 (1991), 159–171.

[GrKlRi93] J. L. Gross, E. W. Klein and R. G. Rieper, On the average genus of a graph, *Graphs and Combinatorics* 9 (1993), 153–162.

[GrRoTu89] J. L. Gross, D. P. Robbins, and T. W. Tucker, Genus distributions for bouquets of circles, *J. Combin. Theory Ser. B* 47 (1989), 292–306.

[GrTu79] J. L. Gross and T. W. Tucker, Local maxima in graded graphs of imbeddings, *Ann. NY Acad. Sci.* 319 (1979), 254–257.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover Publications, 2001. First Edition, Wiley-Insterscience, 1987.

[GrTu95] J. L. Gross and T. W. Tucker, Stratified graphs for imbedding systems, *Discrete Math.* 143 (1995), 71–86.

[Ja87] D. M. Jackson, Counting cycles in permutations by group characters, with an application to a topological problem, *Trans. Amer. Math. Soc.* 299 (1987), 785–801.

[KeGe71] J. Keilson and H. Gerber, Some results for discrete unimodality, *J. Amer. Statist. Assoc.* 66 (1971), 386–389.

[KwKiLe96] J. H. Kwak, S. G. Kim, and J. Lee, Distributions of regular branched prime-fold coverings of surfaces, *Discrete Math.* 156 (1996), 141–170.

[KwLe94] J. H. Kwak and J. Lee, Enumeration of graph embeddings, *Discrete Math.* 135 (1994), 129–151.

[KwSh02] J. H. Kwak and S. H. Shim, Total embedding distributions for bouquets of circles, *Discrete Math.* 248 (2002), 93–108.

[MaSt96] C. Mauk and S. Stahl, Cubic graphs whose average number of regions is small, *Discrete Math.* 159 (1996), 285–290.

[McG87] L. A. McGeoch, Ph.D. Thesis, Carnegie-Mellon University, 1987.

[Mo89] B. Mohar, An obstruction to embedding graphs in surfaces, *Discrete Math.* 78 (1989), 135–142.

[MuRiWh88] B. G. Mull, R. G. Rieper, and A. T. White, Enumerating 2-cell imbeddings of complete graphs, *Proc. Amer. Math. Soc.* 103 (1988), 321–330.

[Ri90] R. G. Rieper, Ph.D. Thesis, Western Michigan University, 1990.

[Ri74] G. Ringel, *Map Color Theorem*, Springer-Verlag, 1974.

[Ri95] A. Riskin, On the enumeration of polyhedral embeddings of Cartesian products of cycles, *Ars Combinatoria* 41 (1995), 193–198.

[Sc99] M. Schultz, Random Cayley maps for groups generated by involutions, *J. Combin. Theory Ser. B* 76 (1999), 247–261.

[St90] S. Stahl, Region distributions of graph embeddings and Stirling numbers, *Discrete Math.* 82 (1990), 57–78.

[St91] S. Stahl, Permutation-partition pairs III: Embedding distributions of linear families of graphs, *J. Combin. Theory Ser. B* 52 (1991), 191–218.

[St91a] S. Stahl, Region distributions of some small diameter graphs, *Discrete Math.* 89 (1991), 281–299.

[St92] S. Stahl, On the number of maximum genus embeddings of almost all graphs, *Eur. J. Combin.* 13 (1992), 119–126.

[St95a] S. Stahl, Bounds for the average genus of the vertex-amalgamation of graphs, *Discrete Math.* 142 (1995), 235–245.

[St95b] S. Stahl, On the average genus of the random graph, *J. Graph Theory* 20 (1995), 1–18.

[St97] S. Stahl, On the zeros of some genus polynomials, *Canad. J. Math.* 49 (1997), 617–640.

[Te00] E. H. Tesar, Genus distribution of Ringel ladders, *Discrete Math.* 216 (2000), 235–252.

[Th89] C. Thomassen, The graph genus problem is NP-complete, *J. Algorithms* 10 (1989), 568–576.

[Xu79] N. H. Xuong, How to determine the maximum genus of a graph, *J. Combin. Theory Ser. B* 26 (1979), 217–225.

7.4 VOLTAGE GRAPHS

Jonathan L. Gross, Columbia University

7.4.1 Regular Voltage Graphs
7.4.2 Net Voltages, Local Group, and Natural Automorphisms
7.4.3 Permutation Voltage Graphs
7.4.4 Representing Coverings with Voltage Graphs
7.4.5 The Kirchhoff Voltage Law
7.4.6 Imbedded Voltage Graphs
7.4.7 Topological Current Graphs
7.4.8 Lifting Voltage Graph Mappings
7.4.9 Applications of Voltage Graphs
References

Introduction

In the voltage graph construction, a small graph with algebraic labels (called "voltages") on its edges specifies a large graph with global symmetries. A Cayley graph for a group can be specified by assigning group elements to the self-loops of a one-vertex graph (a "bouquet"). In this sense, voltage graphs are a generalization of Cayley graphs.

7.4.1 Regular Voltage Graphs

The usual purpose of a voltage graph is to specify an undirected graph. Accordingly, even though the voltage graph construction formally employs directions on the edges as a formal convenience, the terminology adopted concentrates on the undirected object.

DEFINITIONS

The regular voltage graph construction now described was introduced in [Gr74]. The definition of a Cayley graph here is as in §7.5. See §6.2 for an algebraic perspective.

D1: Let $G = (V, E)$ be a digraph and $\mathcal{B}$ a group. A ***regular voltage assignment*** for G in $\mathcal{B}$ is a function $\alpha : E \to \mathcal{B}$ that labels each edge e with a value $\alpha(e)$.

- The pair $\langle G, \alpha : E \to \mathcal{B}\rangle$ is called a ***regular voltage graph***.
- Graph G is called the ***base graph*** and group $\mathcal{B}$ is called the ***voltage group***.
- The label $\alpha(e)$ is called the ***voltage*** on edge e.

D2: The ***derived digraph*** G^α associated with a given regular voltage graph $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$ is defined as follows:

- $V(G^\alpha) = V^\alpha = V \times \mathcal{B}$, the cartesian product.
- $E(G^\alpha) = E^\alpha = E \times \mathcal{B}$.
- If the edge e is directed from vertex u to vertex v in G, then the edge $e_b = (e, b)$ in G^α is from the vertex $u_b = (u, b)$ to the vertex $v_{b\alpha(e)} = (v, b\alpha(e))$.

NOTATION: Vertices and edges of the derived graph are usually specified in the subscript notation, rather than in cartesian product notation. The only standard exception to this convention is to avoid double subscripting.

TERMINOLOGY NOTE: The digraph G^α is usually called, simply, the ***derived graph***. Moreover, its underlying (undirected) graph is also denoted G^α and is also called the ***derived graph***. Such shared terminology avoids excessively formalistic prose. In context, no ambiguity results.

D3: The ***Cayley graph*** $C(\mathcal{A}, X)$ for a group $\mathcal{A}$ with generating set X has the elements of $\mathcal{A}$ as vertices and has edges directed from a to ax for every $a \in \mathcal{A}$ and $x \in X$. We will assume that vertices are labeled by elements of $\mathcal{A}$ and that edges are labeled by elements of X. Although an involution x (i.e., an element of order 2 in the group $\mathcal{A}$) gives rise to a directed edge from a to ax and also one from ax to a, for all a, sometimes we will choose to identify these pair of edges to a single undirected edge labeled x. In §6.2, such a pair is always represented by a single edge.

EXAMPLES

E1: Figure 7.4.1 shows how a Cayley graph for the cyclic group $\mathbb{Z}_5$ is specified by assigning the elements 1 mod 5 and 2 mod 5 to the two self-loops.

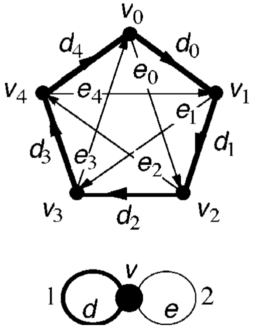


Figure 7.4.1 **A voltage assignment in $\mathbb{Z}_5$ for the Cayley graph K_5.**

E2: Figure 7.4.2(i) is a regular voltage graph $\langle G, \alpha : E \to \mathbb{Z}_3 \rangle$, and Figure 7.4.2(ii) is the corresponding derived graph.

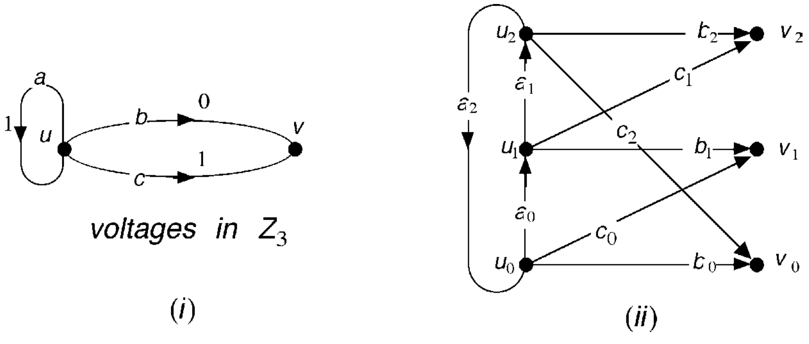


Figure 7.4.2 **A regular voltage assignment and the derived graph.**

Each a-edge of the derived graph G^α joins two u-vertices in G^α, because edge a of the base graph is a self-loop at vertex u. Since the voltage on edge a is 1 mod 3, each subscript increments by 1 in a traversal of an a-edge from tail to head. Since edge b of the base graph G goes from vertex u to vertex v, each of the b-edges in the derived graph G^α crosses from a u-vertex to a v-vertex. The subscripts on tail and on head are equal on all the b-edges, because edge b carries voltage 0.

TERMINOLOGY

Involutions in the voltage group. Assigning an involution x as the voltage to a self-loop e at a vertex v in the base graph causes the e-edges in the derived digraph to be paired. That is, the directed edge e_b from vertex v_b to vertex v_{bx} is paired with the directed edge e_{b+x} from v_{b+x} to v_b.

TERMINOLOGY NOTE: The term ***derived graph*** also refers to the undirected graph that is obtained by identifying these pairs of directed edges to a single edge, as one may do with (topological) Cayley graphs.

REMARK

R1: The earliest application of voltage graphs was to construct imbeddings of large graphs on surfaces, often in connection with minimum genus or symmetric maps.

Fibers

DEFINITIONS

D4: Let G^α be the derived graph for a regular voltage graph $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$.

- The vertex subset $\{v\} \times \mathcal{B} = \{v_b : b \in \mathcal{B}\}$ is called the ***(vertex) fiber*** over v.
- Similarly, the edge subset $\{e\} \times \mathcal{B} = \{e_b : b \in \mathcal{B}\}$ is called the ***(edge) fiber*** over e.

D5: Let $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$ be a regular voltage graph. The graph mapping from the derived graph G^α to the voltage graph G given by the vertex function and edge function

$$v_b \mapsto v \quad e_b \mapsto e$$

respectively, is called the ***natural projection***. (Thus, the natural projection is given by "erasure of subscripts".)

EXAMPLE

E2, continued: In Figure 7.4.2, the subset $\{u_0, u_1, u_2\}$ of $V(G^\alpha)$ (what we were calling the "u-vertices") is the vertex fiber over u. The subset $\{b_0, b_1, b_2\}$ of $E(G^\alpha)$ (what we were calling the "b-edges") is the edge fiber over b.

FACT

F1: It is clear from the definition of the derived graph that the vertex-set of the derived graph is partitioned into $|V|$ fibers, each with $|\mathcal{B}|$ vertices. Similarly, the edge-set of the derived graph is partitioned into $|E|$ fibers, each with $|\mathcal{B}|$ vertices.

Bouquets and Dipoles

For economy of description, it is helpful to use a base graph with as few vertices as possible.

DEFINITIONS

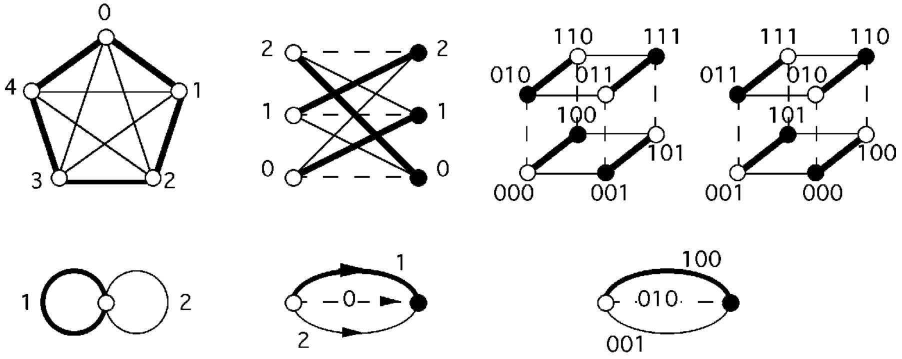
D6: The ***bouquet*** B_n is the one-vertex graph with n self-loops.

D7: The ***dipole*** D_n is the two-vertex graph with n edges joining the two vertices.

EXAMPLE

Voltage graph theory is intuitively spatial. Instead of cluttering the drawings with cumbersome labels, one uses graphic features to represent the partitions into fibers.

E3: Figure 7.4.3 illustrates how graphic features are used. For instance, in each derived graph, a particular fiber and its corresponding voltage assignment are displayed in bold.



(*i*) *voltages in* Z_5 (*ii*) *voltages in* Z_3 (*iii*) *voltages in* Z_2^3

Figure 7.4.3 **Three regular voltage assignments and their derived graphs.**

Figure 7.4.3(i) derives K_5 with $\mathbb{Z}_5$-voltages on B_2, as in Figure 7.4.1. This time, we have "suppressed the mainscript" on the vertex fiber in the derived graph, and shown only the subscripts. Moreover, we have suppressed the directions in the derived graph.

Figure 7.4.3(ii) derives $K_{3,3}$ with $\mathbb{Z}_3$-voltages on D_3. Using hollow and solid vertex graphics enables us to label all the vertices in the derived graph by their subscripts, without loss of information. Similarly, the edge graphics enable us to verify readily that the edges in each edge fiber join vertices whose labels differ by the correct amount.

Figure 7.4.3(iii) derives the union of two isomorphic copies of the cube graph Q_3 with $\mathbb{Z}_2^3$-voltages on D_3. This phenomenon is examined in §7.4.2.

Additional elementary examples appear in [GrTu87] and in [GrYe99].

FACTS

F2: The Cayley graph of a group with generating set $\{x_1, \ldots, x_k\}$ is (naturally) isomorphic to the derived graph specified by the bouquet B_k with voltages $\{x_1, \ldots, x_k\}$ on its respective self-loops.

F3: The complete graph K_{2n+1} can be derived by assigning the voltages $1, 2, \ldots, n$ from the cyclic group $\mathbb{Z}_{2n+1}$ to the edges of the bouquet B_n. (This is a special instance of Fact 2.)

F4: The complete graph K_{2n} can be derived by assigning the voltages $1, 2, \ldots, n$ from the cyclic group $\mathbb{Z}_{2n}$ to the edges of the bouquet B_n, if one compresses the pairs of edges that arise from the involution n mod $2n$.

F5: The symmetric complete bipartite graph $K_{n,n}$ can be derived by assigning voltages $0, 1, \ldots, n-1$ in the cyclic group $\mathbb{Z}_n$ to the n edges of the dipole D_n.

F6: The d-dimensional cube graph Q_d can be derived by assigning the d elementary vectors in $\mathbb{Z}_2^d$ to the edges of the dipole D_d.

7.4.2 Net Voltages, Local Group, and Natural Automorphisms

Net Voltages

DEFINITIONS

D8: A ***walk in a voltage graph*** is any walk, as if the voltage graph were undirected. This means that some of its edge-steps may proceed in the opposite direction from the direction on the edge it traverses.

D9: The ***voltage sequence*** on a walk $W = v_0, e_1, v_1, e_2, ..., e_n, v_n$ is the sequence of voltages $a_1, ..., a_n$ encountered, where $a_j = \alpha(e_j)$ or $\alpha(e_j)^{-1}$, depending on whether edge e_j is traversed in the forward or backward direction, respectively.

D10: The ***net voltage on a walk*** in a voltage graph is the product of the algebraic elements in its voltage sequence.

EXAMPLE

E2, continued: We return to the voltage graph of Figure 7.4.2, reproduced here for convenience.

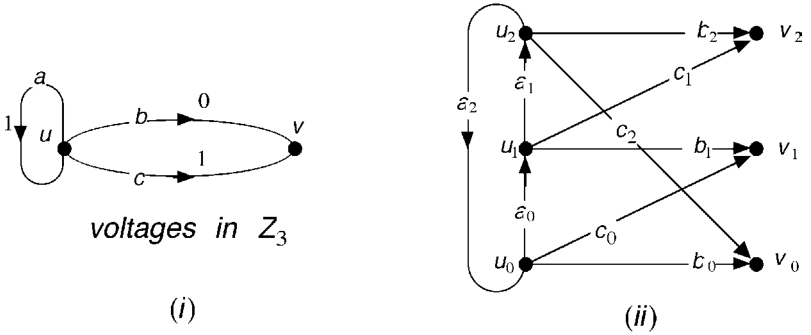


Figure 7.4.4 **A regular voltage assignment and the derived graph.**

The walk $W = u, c, v, b^-, u, c, u$ has net voltage $1 + 0 + 1 = 2 \bmod 3$. We observe that in the walk $u_0, c_0, v_1, b_1^-, u_1, c_1, v_2$ in the derived graph, which follows the same c, b^-, c edge pattern as walk W, the subscript increases by 2 from initial vertex to final vertex. This phenomenon is examined in §7.4.5.

The Local Group

DEFINITION

D11: The ***local group*** at a vertex v of a voltage graph $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$ is the subgroup of all elements of $\mathcal{B}$ that occur as the net voltage on a closed walk that starts and ends at vertex v. It is denoted Loc_v.

FACTS

F7: [AlGr76] If the voltage group is abelian, then the local group is the same at every vertex. If it is non-abelian, then the local group at a vertex is conjugate to the local group at any other vertex.

F8: [AlGr76] For any regular voltage graph $\langle G, \alpha : E \to \mathcal{B} \rangle$, the number of components of the derived graph equals the index $[\mathcal{B} : \mathrm{Loc}_v]$ of the local group in the voltage group.

F9: [AlGr76] The components of the derived graph are mutually isomorphic.

EXAMPLE

E3, continued: The local group for Figure 7.4.3(iii) is the subgroup of 3-tuples with evenly many 1's. This subgroup has index two in $\mathbb{Z}_2^3$. Thus, there are two components to the derived graph.

REMARKS

R2: If we select a root vertex r in the base graph, then the component of the derived graph containing the vertex r_0 (here, zero denotes the group identity, even for a non-abelian group) serves as a preferred component of the derived graph.

R3: The theory of the local group and multiple components was developed by [AlGr76] in the terminology of topological current groups and general algebra.

Natural Automorphisms

The natural action of the group on any of its Cayley graphs generalizes to a natural action of a voltage graph on the derived graph.

DEFINITION

D12: Let $\langle G, \alpha : E \to \mathcal{B} \rangle$ be a voltage graph, and let $x \in \mathcal{B}$. The ***natural automorphism*** $\varphi_x : G^\alpha \to G^\alpha$ is given by the rules

$$u_b \mapsto u_{xb} \quad \text{and} \quad e_b \mapsto e_{xb}$$

Thus, if edge e runs from vertex u to vertex v of the base graph, then edge e_b runs from vertex u_b to vertex $v_{xb\alpha(e)}$ in the derived graph.

FACTS

See [GrTu87] for details.

F10: The group of natural transformations is fiber preserving. That is, each vertex and edge of the derived graph is mapped to another vertex or edge, respectively, within the same fiber. (This is immediate from the definition of a natural transformation.)

F11: The group of natural transformations acts transitively on the vertices within each vertex fiber and transitively on the edges within each edge fiber.

F12: Let $\langle G, \alpha : E \to \mathcal{B} \rangle$ be a voltage graph. A component of the derived graph G^α that contains a vertex in the fiber over v is mapped to itself by the natural automorphism φ_x if and only if $x \in \mathrm{Loc}_v$.

EXAMPLE

E3, continued: The natural automorphism φ_{xyz} for Figure 7.4.3(iii) maps a component of the derived graph to itself if and only if xyz has evenly many 1's.

7.4.3 Permutation Voltage Graphs

The permutation voltage graph construction of [GrTu77] uses the objects permuted by a permutation group as the subscripts, rather than using the group elements as in the regular voltage graph construction. This leads to increased generality (Fact 15).

DEFINITIONS

D13: Let $G = (V, E)$ be a digraph. A ***Σ_n-permutation voltage assignment*** for G is a function $\alpha : E \to \Sigma_n$ that labels each edge with a permutation in the symmetric group.

- The pair $\langle G, \alpha \rangle$ is called a ***Σ_n-permutation voltage graph***.
- Graph G is called the ***base graph*** and group Σ_n is called the ***permutation voltage group***.
- The permutation label $\alpha(e)$ is called the ***voltage*** on edge e.

D14: The ***(Σ_n-permutation) derived digraph*** G^α associated with a permutation voltage graph $\langle G = (V, E), \alpha : E \to \Sigma_n \rangle$ is defined as follows:

- $V(G^\alpha) = V^\alpha = V \times \{1, ..., n\}$, the cartesian product.
- $E(G^\alpha) = E^\alpha = E \times \{1, ..., n\}$.
- If the edge e is from vertex u to vertex v in G then the edge $e_j = (e, j)$ is from the vertex $u_j = (u, j)$ to the vertex $v_{\alpha(j)} = (v, \alpha(j))$.

EXAMPLE

E4: Figure 7.4.5 shows a Σ_3-permutation voltage graph and the corresponding derived graph.

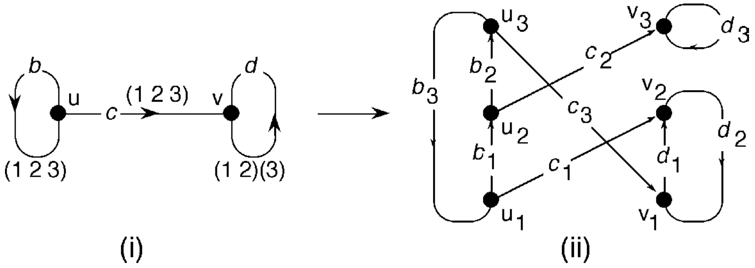


Figure 7.4.5 **(i) A Σ_3-voltage graph; (ii) the derived digraph.**

The edge fiber over self-loop b at vertex u of the base graph forms the single cycle $(u_1, b_1, u_2, b_2, u_3, b_3)$, because b has voltage (123). The edge fiber over self-loop d forms two disjoint cycles $(v_1, d_1 v_2, d_2)(v_3, d_3)$, because the voltage on d is $(12)(3)$. Since edge c goes from vertex u to vertex v in the base graph, each edge in the fiber over c crosses from the vertex fiber over u to the vertex fiber over v, and the vertex subscripts are permuted in accordance with the voltage (123) on edge c.

FACTS

F13: [Gr77] Every regular graph of even degree $2k$ is specifiable by assigning permutation voltages to the bouquet B_k. (See also [SiSk85].)

F14: From Bäbler's theorem [Bä38] (see Fact 6 of §5.4), it follows that every $2k$-edge-connected $(2k+1)$-regular graph is specifiable by assigning permutation voltages to the bouquet B_k, if one permits a 1-factor to represent the fiber resulting from an involution.

F15: Any graph derivable by regular voltages is also derivable by permutation voltages. This follows from the fact that the right regular representation of any group can be imbedded in a symmetric permutation group.

REMARKS

R4: Analogous to the regular case, the vertex-set and edge-set of a Σ_n-permutation derived graph are partitioned respectively into $|V|$ vertex-fibers, each with n vertices, and into $|E|$ edge-fibers, each with n edges. Analogously, natural projection is by erasure of subscripts.

R5: Further elementary examples of permutation voltage graphs are given in [GrTu87] and in [GrYe99].

R6: In labeling a voltage graph drawing with permutations, one must specify whether the voltages are to be regarded as permutation voltages or as regular voltages. For permutation voltages, there are n vertices or edges in each fiber. For regular voltages, there are $n!$ vertices or edges.

R7: In particular, it is possible to label the bouquet B_2 with permutations in the wreath product $\mathbb{Z}_n \otimes_{\mathrm{wr}} \mathbb{Z}_2$ (with a cycle shift and a deBruijn permutation) so that the permutation derived graph is the n-dimensional deBruijn graph and the regular derived graph is the *wrapped butterfly* graph.

7.4.4 Representing Coverings with Voltage Graphs

Covering spaces are a topological abstraction of Riemann surfaces. In fact, every covering space of a graph can be specified by assigning voltages. The advantage of specifying a covering graph by voltages, rather than by the classical abstract descriptions, is that the derived graph over a voltage graph has every vertex and edge labeled according to its fiber, in a manner that lends itself to topological and combinatorial intuition.

Coverings and Branched Coverings of Surfaces

DEFINITIONS

D15: Let S and $\tilde{S}$ be surfaces, and let $p : \tilde{S} \to S$ be a continuous function, such that the following condition holds:

- Every point of S has an open neighborhood U such that each component of $p^{-1}(U)$ is mapped homeomorphically by p onto U.

Then $p : \tilde{S} \to S$ is called a ***covering projection*** and the surface $\tilde{S}$ is called a ***covering space*** of S.

D16: Let $p : \tilde{S} \to S$ be a covering projection. For each point $x \in S$, the set $p^{-1}(x)$ is called the ***fiber*** over x.

D17: Let S and $\tilde{S}$ be closed surfaces, and let $\tilde{B}$ be a finite set of points in $\tilde{S}$ such that the restriction of the mapping $p : \tilde{S} \to S$ to $\tilde{S} - \cup\tilde{B}$ is a covering projection. Then

- The mapping $p : \tilde{S} \to S$ is called a ***branched covering***.
- The space $\tilde{S}$ is called a ***branched covering space*** of S.
- The set $\tilde{B}$ is called the ***branch set***.
- The images of points in the branch set are called ***branch points***.

EXAMPLES

E5: The complex function e^{3ix} is a covering projection of the unit circle in the complex plane onto itself. The fiber over a point e^{ix} is the set $\{e^{ix}, e^{ix+2\pi/3}, e^{ix+4\pi/3}\}$. Moreover, the function e^{3ix} is a branched covering of the unit disk in the complex plane onto itself, in which $\{0\}$ is the only branch point.

E6: The classical Riemann surfaces are branched coverings of the complex plane.

E7: [Al20] Every closed orientable surface is a branched covering of the sphere.

E8: Consider the unit sphere $S_0 = \{(x, y, z) \in R^3 : x^2+y^2+z^2 = 0\}$ and the antipodal mapping $(x, y, z) \mapsto (-x, -y, -z)$. The quotient mapping induced by the antipodal homeomorphism is a covering projection of S_0 onto the projective plane. Moreover, the restriction of this covering projection to the annular region of S_0 between the "Tropic of Cancer" and the "Tropic of Capricorn" is a covering projection of this annular region onto a Möbius band.

REMARK

R8: The branch set in any covering of manifolds has codimension 2. Thus unless a graph is imbedded in a surface, there is no branching.

Using Voltage Graph Constructions

A few basic facts serve as a guide to the use of voltage graph constructions.

FACTS

Let $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$ be a regular voltage graph. Then the following statements hold.

F16: $|V(G^\alpha)| = |V(G)| \cdot |\mathcal{B}|$ and $|E(G^\alpha)| = |E(G)| \cdot |\mathcal{B}|$.

F17: In the fiber over a vertex $v \in V(G)$, every vertex v_b has the same degree as v.

F18: A proper coloring of the base graph can be *lifted* to a proper coloring of the voltage graph, in the following sense: every vertex in the fiber over a vertex $v \in V(G)$ is assigned the same color as v. (A graph with self-loops is considered to have no proper colorings.)

F19: [GrTu79] Let T be a spanning tree of a graph G. If a graph $\tilde{G}$ can be constructed by assigning $\mathcal{B}$-voltages to G, then it is possible to do so by completing an assignment of arbitrary voltages from $\mathcal{B}$ to the edges of T.

EXAMPLES

E9: To represent the Petersen graph as a regular covering space of the dumbbell graph, we observe that the Petersen graph has 10 vertices and 15 edges. The only nontrivial common divisor of 10 and 15 is 5, so using Fact 16, we seek a base graph with 2 vertices and 3 edges. There are four such connected graphs. In accordance with Fact 17, the base graph must be 3-valent regular, which narrows the possibilities to two graphs. One of these two, the dipole D_3, is 2-colorable. By Fact 18, it cannot be a base graph for the Petersen graph. This leaves the *dumbbell graph*, shown in Figure 7.4.6, as the only possible base graph.

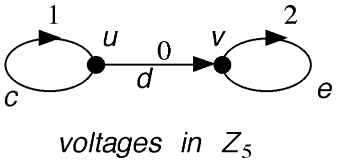


Figure 7.4.6 A regular voltage assignment for the Petersen graph.

We seek $\mathbb{Z}_5$-voltages on the dumbbell graph, since $\mathbb{Z}_5$ is the only group of order 5. By Fact 19, we may start by assigning the voltage 0 to the edge d. Figure 7.4.6 shows the completed assignment.

E10: By similar considerations, we can demonstrate that the complete graph K_4 is not nontrivially derivable as a voltage graph. It has 4 vertices and 6 edges, so the only nontrivial common divisor is 2. By the exact same progression of steps as in Example 9, we narrow the possible candidates down to the dumbbell graph, and we narrow the possible voltage groups to $\mathbb{Z}_2$.

Assigning 0 to a self-loop in the base leads to an edge fiber of self-loops. Assigning the involution 1 to both self-loops leads to a 4-vertex 6-edge graph with two double adjacencies, whose elimination yields a 4-cycle, not K_4.

Action of the Group of Covering Transformations

The subsection is confined to exploring the sense in which voltage graphs provide all possible covering graphs. Prior acquaintance with algebraic topology is helpful.

DEFINITIONS

D18: Let $p : \tilde{S} \to S$ be a covering projection. A homeomorphism h on $\tilde{S}$ is called a ***covering transformation*** if $ph = p$. One sometimes says that such a homeomorphism h is ***fiber preserving***, since its restriction to any fiber is a permutation of that fiber.

D19: A group H of covering transformations on a covering projection $p : \tilde{S} \to S$ is said to ***act freely*** if no transformation in H except the identity has a fixed point in $\tilde{S}$.

D20: A group H of covering transformations on a covering projection $p : \tilde{S} \to S$ is said to ***act transitively*** if its restriction to each fiber acts transitively.

D21: A ***regular covering projection*** is a covering projection $p : \tilde{S} \to S$ such that there exists a group of freely acting covering transformations. In this case, the domain $\tilde{S}$ is called a ***regular covering space*** of S.

TERMINOLOGY: The phrase "covering space of a graph G" is used to describe a covering space of a topological realization of the graph G, e.g., in 3-space. It also refers to any graph $\tilde{G}$ such that there is a graph map $\tilde{G} \to G$ whose topological realization is a covering projection.

EXAMPLE

E10, continued: The three functions $e^{ix} \mapsto e^{ix}, e^{ix} \mapsto e^{ix+2\pi/3}$, and $e^{ix} \mapsto e^{ix+4\pi/3}$ form a group of covering transformations. This group acts freely, and it acts transitively on the fibers over every point of the unit circle. Thus, the complex function e^{3ix} is a regular covering projection of the unit circle onto itself.

FACTS

F20: [Gr74] Let $\langle G, \alpha \rangle$ be a regular voltage graph. Then the derived graph G^{α} is a regular covering graph of G.

F21: [GrTu77] Let X be a topological realization of a graph G be any graph, and let $\tilde{X}$ be a regular covering space of X. Then X is homeomorphic to the topological realization of the derived graph corresponding to a regular voltage assignment on G. ("Every regular covering space of a graph is realizable by a regular voltage assignment.")

F22: [GrTu77] Let $\langle G, \alpha \rangle$ be a permutation voltage graph. Then the derived graph G^{α} is a covering graph of G.

F23: [GrTu77] Let X be a topological realization of a graph G be any graph, and let $\tilde{X}$ be a covering space of X. Then X is homeomorphic to the topological realization of the derived graph corresponding to a permutation voltage assignment on G. ("Every covering space of a graph is realizable by a permutation voltage assignment.")

F24: Let $\langle G, \alpha \rangle$ be a regular voltage graph, whose voltage group $\mathcal{B}$ has order n. Then the corresponding derived graph is isomorphic to the derived graph of the Σ_n-permutation voltage graph $\langle G, \hat{\alpha} \rangle$, where $b \mapsto \hat{b}$ is the right regular representation of the group $\mathcal{B}$ embedded in Σ_n. ("Every regular voltage assignment can be represented as a permutation voltage assignment.")

F25: Let $\langle G, \alpha \rangle$ be any regular or permutation voltage graph, and let e be a directed edge of graph G. If the direction of e is reversed, and if the voltage $\alpha(e)$ is replaced by its algebraic inverse $\alpha(e)^{-1}$, then the resulting derived graph is isomorphic to the derived graph G^{α}.

F26: Let $p : C \to C$ be a covering projection of the unit circle onto itself, such that each point in the image is covered k times. Then $p : C \to C$ extends to a branched covering of the unit disk itself, in which 0 is the only branch point in the codomain and $\{0\}$ is the branch set in the domain.

7.4.5 The Kirchhoff Voltage Law

We recall that the Kirchhoff voltage law for electrical circuits asserts that the net voltage change around a circuit is zero. In graph theory, voltages are generalized to elements of an arbitrary group, and zero is generalized to the group identity.

DEFINITIONS

D22: Let $W = v_0, e_1, v_1, e_2, ..., e_n, v_n$ be a walk in a regular voltage graph $\langle G, \alpha : E \to \mathcal{B}\rangle$, and let $a_1, \ldots, a_n$ be its voltage sequence. Let $b \in \mathcal{B}$. Then the walk

$$W_b = (v_0, b), (e_1, ba_1), (v_1, ba_1), (e_2, ba_1a_2), \ldots, (e_n, ba_1a_2 \cdots a_n), (v_n, ba_1a_2 \cdots a_n)$$

is called a ***lift*** of the walk W.

D23: Let $W = v_0, e_1, v_1, e_2, ..., e_k, v_k$ be a walk in a permutation voltage graph $\langle G, \alpha : E \to \Sigma_n\rangle$, and let $\eta_1, \ldots, \eta_k$ be its voltage sequence. Let $j \in \{1, \ldots, n\}$. Then the walk

$$\begin{gathered}(v_0, j), (e_1, \eta_1(j)), (v_1, \eta_1(j)), (e_2, \eta_2(\eta_1(j))), \\ \ldots, (e_k, \eta_k(\cdots(\eta_1(j)))), (v_k, \eta_k(\cdots(\eta_1(j))))\end{gathered}$$

is called a ***lift*** of the walk W.

D24: Let W be a closed walk in a voltage graph. If the net voltage on W is the identity of the voltage group, then we say that ***the Kirchhoff voltage law (KVL)*** holds on W.

FACTS

F27: [Gr74, GrTu77] Let W be a closed walk in a voltage graph. If the Kirchhoff voltage law holds on W, then every lift W_b of W is a closed walk in the derived graph.

F28: [Gr74] Let W be a closed walk in a regular voltage graph $\langle G, \alpha : E \to \mathcal{B}\rangle$, with net voltage c. Let c have order k in the voltage group $\mathcal{B}$. Then the concatenation of the sequence of lifts $W_b, W_{bc}, W_{bc^2}, \ldots, W_{bc^{k-1}}$ is a closed walk in the derived graph G^α.

NOTATION: Under the hypotheses of Fact 28, the set of lifts of the walk W is conceptualized as partitioned into $\frac{|\mathcal{B}|}{k}$ sequences of lifts, as in the conclusion, whose concatenations are closed walks in the derived graph. This set of closed walks formed by such concatenation is denoted W^*.

F29: [GrTu77] Let W be a closed walk in a permutation voltage graph $\langle G, \alpha : E \to \Sigma_n\rangle$, with net voltage η. Let η have order k in the voltage group Σ_n. Then the concatenation of the sequence of lifts $W_j, W_{\eta(j)}, W_{\eta^2(j)}, \ldots, W_{\eta^{k-1}(j)}$ is a closed walk in the derived graph G^α.

NOTATION: Under the hypotheses of Fact 29, the set of lifts of the walk W is conceptualized as partitioned into $\frac{n}{k}$ sequences of lifts, as in the conclusion, whose concatenations are closed walks in the derived graph. This set of closed walks formed by such concatenation is denoted W^*, as in the regular case.

7.4.6 Imbedded Voltage Graphs

Imbedded voltage graphs and their duals, called current graphs, are used to specify the imbeddings of graphs on surfaces. Imbedded voltage graphs are used extensively in calculations of maximum and minimum genus of a graph (see §7.2), in calculating the minimum genus of a group (see §7.5), and in constructing regular maps on surfaces (see §7.6).

DEFINITIONS

D25: Let $\langle G, \alpha\rangle$ be a voltage graph such that the graph G is (cellularly) imbedded in a closed surface S. Then the pair $\langle G \to S, \alpha\rangle$ is called an ***imbedded voltage graph***; also, $\langle G \to S, \alpha\rangle$ is called the base imbedding, and S is called the ***base surface***.

D26: Let Ω be the set of closed walks of the faces of an imbedded voltage graph $\langle G \to S, \alpha\rangle$. Then the union $\tilde{\Omega}$ of the sets W^*, where $W \in \Omega$, is called the ***set of lifted boundary walks***.

D27: Let $\tilde{\Omega}$ be the set of lifted boundary walks in the derived graph G^α for an imbedded voltage graph $\langle G \to S, \alpha\rangle$. The cellular 2-complex S^α that results from fitting to each closed walk in $\tilde{\Omega}$ a polygonal region (whose number of sides equals the length of that closed walk) is called the ***derived surface***. The imbedding $G \to S^\alpha$ is called the ***derived imbedding***.

D28: Let $\langle G \to S, \alpha\rangle$ be an imbedded voltage graph. To extend the natural projection $p : G^\alpha \to G$ to the surfaces, the natural projection p is extended from the set of lifted boundary walks in the imbedding $G^\alpha \to S^\alpha$ to the regions they bound (with branching as needed), in accordance with Fact 28. The resulting extended function is called the ***natural projection***.

D29: A ***monogon*** is a face whose boundary walk has length equal to 1.

D30: A ***digon*** is a face whose boundary walk has length equal to 2.

FACTS

F30: Let $\langle G \to S, \alpha\rangle$ be an imbedded voltage graph. Then the derived surface is a closed surface, and the derived imbedding is a cellular imbedding; moreover, if the base surface S is orientable, then so is the derived surface.

F31: [GrAl74, Gr74] Let $\langle G \to S, \alpha\rangle$ be an imbedded voltage graph. If the Kirchhoff voltage law holds on the boundary walk of every face of the base imbedding, then the natural projection $p : \tilde{S} \to S$ is a covering projection. If KVL does not hold, then the natural projection is a branched covering.

EXAMPLES

E11: Figure 7.4.7 shows an imbedded ordinary voltage graph in which the base graph is the bouquet B_2, the base surface is the torus S_1, and the voltages are in the cyclic group Z_5. The derived graph is the complete graph K_5 and the derived surface is the torus S_1. There is only one base face, and KVL holds on its boundary walk. Thus, each of the derived faces has the same number of sides as the base face, and the natural projection is a covering projection.

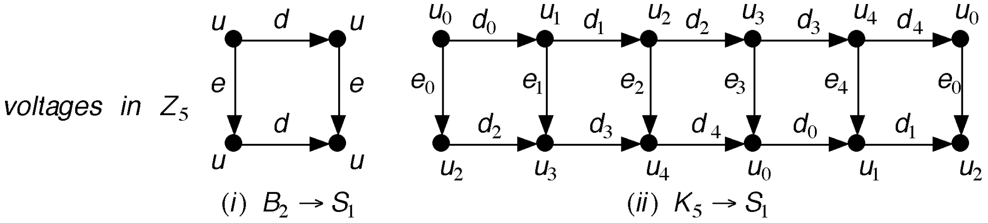

voltages in Z_5
u
d
u
e
e
d
u
u
(i) $B_2 \to S_1$
u_0 d_0 u_1 d_1 u_2 d_2 u_3 d_3 u_4 d_4 u_0
e_0 e_1 e_2 e_3 e_4 e_0
d_2 d_3 d_4 d_0 d_1
u_2 u_3 u_4 u_0 u_1 u_2
(ii) $K_5 \to S_1$

Figure 7.4.7 (i) an imbedded voltage graph; (ii) its derived imbedding.

E12: Figure 7.4.8 shows another imbedded ordinary voltage graph in which the base graph is the bouquet B_2 and the voltages are in the cyclic group Z_5, but the base surface is the sphere S_0. The derived graph is the complete graph K_5.

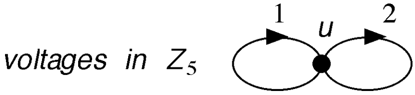


Figure 7.4.8 An imbedded voltage graph $B_2 \to S_0$.

There are three base faces, i.e., two monogons and one digon, and KVL does not hold on any of their boundary walks; indeed, the net voltage on each boundary walk has order 5 in the group Z_5. Thus, each of the derived faces has 5 times as many sides as the base face, so there are two 5-gons and one 10-gon; the natural projection is a branched covering projection, with a branch point in each base face. Since the Euler characteristic of the derived surface is $-2 = 5 - 10 + 3$, it follows that the derived surface is S_2.

E13: [Gr74] Every Z-metacyclic group with presentation

$$\left\langle s, t \mid s^m = t^n = e, t^{-1}st = s^{-1} \right\rangle$$

such that m is odd and n is even is a toroidal group. Various such results on the genus of a group have been derived with the aid of voltage graphs.

7.4.7 Topological Current Graphs

The origin of voltage graphs was in the pursuit of a unified explanation (see [GrAl74] and [GrTu74]) of the 300-page Ringel-Youngs solution [RiYo68] (see also [Ri74]) to the *Heawood map-coloring problem* [He:1890], which is to calculate the chromatic number of every closed surface except the sphere. Several extensions of voltage graph theory have augmented its utility.

DEFINITIONS

D31: Let G be a digraph with vertex-set V and edge-set E, imbedded in a surface S. A ***regular current assignment*** for G in a group $\mathcal{B}$ is a function α from E to $\mathcal{B}$. The function value $\alpha(e)$ is called the ***current*** on edge e. The pair $\langle G \to S, \alpha \rangle$ is called a ***regular current graph***, and $\mathcal{B}$ is called the ***current group***.

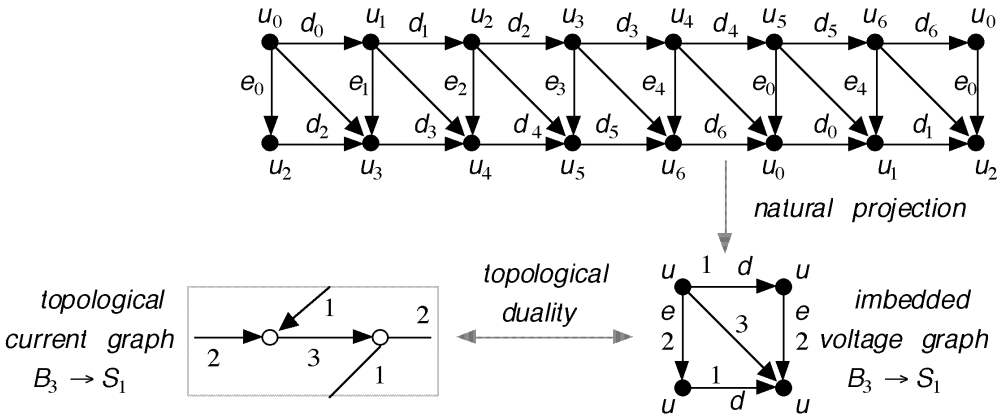
D32: Let $\langle G = (V, E) \to S, \alpha : E \to \mathcal{B} \rangle$ be a regular current graph. Its ***dual*** is the imbedded voltage graph whose base imbedding is $G^* \to S$, the dual of the imbedding $G \to S$ (which involves reversed orientation from the primal imbedding surface, if S is orientable). For each primal directed edge $e \in E$, we define $\alpha^*(e^*) = \alpha(e)$ to be its voltage.

D33: The ***derived imbedding of a current graph*** $\langle G \to S, \alpha : E \to \mathcal{B} \rangle$ be a current graph is the derived imbedding of its dual, that is, of the imbedded voltage graph $\langle G^* \to S, \alpha^* \rangle$.

D34: Let v be a vertex in a current graph. If the net current at v is the identity of the current group, then we say that ***the Kirchhoff current law (KCL)*** holds at v. (In an abelian group, the net current is the sum of the inflowing currents. In a non-abelian group, one calculates the product in the cyclic order of the rotation at v.)

EXAMPLE

E14: In Figure 7.4.9, all three drawings shown are on tori, with the left side of the rectangle pasted to the right, and the top pasted to the bottom. The derived imbedding is pasted with a $\frac{2}{7}$ twist, so that like labels match. We observe that KVL holds on both faces of the imbedded voltage graph. In accordance with duality, KCL holds at both vertices of the corresponding current graph.



u_0 d_0 u_1 d_1 u_2 d_2 u_3 d_3 u_4 d_4 u_5 d_5 u_6 d_6 u_0
e_0 e_1 e_2 e_3 e_4 e_0 e_4 e_0
d_2 d_3 d_4 d_5 d_6 d_0 d_1
u_2 u_3 u_4 u_5 u_6 u_0 u_1 u_2
natural projection
topological current graph
$B_3 \to S_1$
1
2
2
3
1
topological duality
u 1 d u
e e
2 3 2
1
u d u
imbedded voltage graph
$B_3 \to S_1$

Figure 7.4.9 **Deriving an imbedding from a topological current graph.**

REMARKS

R9: In Gustin's original conception [Gu63], a current graph was a 3-regular graph whose vertices were marked with instructions for traversing a small family of closed walks that doubly covered its edges, and whose edges were marked with algebraic labels. Ringel and Youngs augmented Gustin's "nomograms" (Youngs's terminology) into numerous varieties of combinatorial current graph, each with a distinct set of defining rules, whatever helped in their endeavors to solve the Heawood problem. The common feature of all varieties was that recording the algebraic elements traversed along those closed walks yielded generating rows for rotations systems (see §7.1) that specified minimum-genus imbeddings for complete graphs.

R10: The theory of topological current graphs [GrAl74] generalized the various combinatorial current graphs referred to in Remark 9 into a single unified construction, applicable not just to complete graphs, but to a wide variety of graphs with symmetries, and it identified the underlying construction as a branched covering. The theory of regular voltage graphs [Gr74] separated the base graph from its imbedding, which facilitated a divide-and-conquer approach to constructing imbeddings, as in Example 9. The theory of permutation voltage graphs [GrTu77] expanded the construction.

R11: When one dualizes the natural projection of the derived imbedding onto an imbedded voltage graph, there is a natural projection of the dual of the derived graph onto the current graph, which is a folded covering in the sense of [Tu36]. The relationship of the dual derived graph to the current graph was studied by [PaPiJa80] and [GrJaPaPi82]. A different perspective on simultaneous consideration of voltages and currents is given by [Ar92].

7.4.8 Lifting Voltage Graph Mappings

In porting distributed algorithms between parallel architectures, a theoretical problem that arises is the construction of a mapping between two large symmetric graphs that would minimize the slowdown involved in emulating a computation designed for one computer architecture on a different architecture.

DEFINITIONS

D35: Let $f : G \to H$ be a graph map. The ***guest*** is the domain, and the ***host*** is the codomain. (A computation designed for the guest is to be emulated by the host.)

D36: Let $f : G \to H$ be a graph map. The ***load at a vertex*** v of the host is the cardinality $|f^{-1}(v)|$ of its preimage. (A processor at host vertex v is required to reproduce the computations of the processors at every guest vertex mapped to v. These must be done consecutively, so there is a delay at v proportional to the load.) The ***load of the mapping*** $f : G \to H$ is the maximum vertex load, taken over all vertices of the host.

D37: Let $f : G \to H$ be a graph map. The ***congestion at an edge*** e of the host is the cardinality $|f^{-1}(e)|$ of its preimage. (A link at host edge e is required to carry the messages of the links represented by every guest edge mapped to e. These must be done consecutively, so there is a delay at v proportional to the load.) The ***congestion of the mapping*** $f : G \to H$ is the maximum edge congestion, taken over all edges of the host.

FACTS

F32: [GrCh96] A graph mapping from a guest G to a host H can *sometimes* be constructed by lifting a graph mapping from a voltage graph for G to a voltage graph for H.

F33: [ArGvSi97], [ArRiSiSk97] A mapping of imbedded graphs can *sometimes* be constructed by lifting a graph mapping from an imbedded voltage graph for G to an imbedded voltage graph for H.

F34: [GrCh96] A graph mapping from a guest G to a host H that minimizes load or congestion can *sometimes* be constructed by lifting a graph mapping from a voltage graph for G to a voltage graph for H.

REMARK

R12: Other references about lifting mappings include [MaNeSk00], [MiSa00a], and [Su90].

7.4.9 Applications of Voltage Graphs

Beyond their initial uses in the construction of minimum imbeddings of graphs and maps with various forms of symmetry, voltage graphs have acquired several other uses.

REMARKS

R13: The use of voltage graphs in the construction of graphs with special prescribed properties or in the calculation of invariants appears in [AnGa81], [ArKwLeSo00], [FeKw02], [KrPrTe97], [KwHoLeSo00], [KwLeSo99], [Le82], [MiSa97], [MiSa00b], [MiSa02], and [MiSa03].

R14: For an extensive survey of counting covering spaces of a graph, see [KwLe04].

R15: Among the many papers that apply voltage graphs to counting covering spaces of a graph are [EwHo93], [FeKwKiLe98], [Ho91a], [Ho91b], [Ho95a], [Ho95b], [HoKw93], [HoKwLe96], [HoKwLe99], [KiKiLe99], [KwChLe98], [KwHoLeSo00], [KwLe92], [KwLe94], [KwLe98], and [KwKiLe96].

R16: For a study of random permutation voltages, see [AmLi02].

R17: For connections to the Vassiliev knot invariants, see [Pl01].

R18: For connections between voltage graphs and coding theory, see [LaPe81].

R19: For connections between voltage graphs and biological networks, see [MoRe02].

R20: For the applications of voltage graphs to the study of isomorphisms of graph coverings, see [Sa94] and [Sa01].

Applications of Imbedded Voltage Graphs and Topological Current Graphs

Imbedded voltage graphs and topological current graphs have often been used in connection with investigations in algebra and geometry.

REMARKS

R21: For results on lifting imbeddings with special properties, including symmetries and the realization of minimum imbeddings for a graph, see [Mo85], [Mo86], [Mo87b], [Mo88], [NeSk97], [Pa80], [Pi80], [Pi82], and [St79].

R22: For the use of voltage graphs in counting graph imbeddings, see [KwChLe98], [KwKiLe96], [LeKi01], [LeKi02], and [Mo87a].

R23: Applications of voltage graphs or topological current graphs to the genus of a group (see §7.5) appear in [Gr74], [GrLo80], [JuWh80], [Pi92], [PiTu89], [Pr77], [Pr81], [PiTu97], [PiTuWi92], [Tu80], [Tu84], and [Tu83].

R24: Voltage graphs have been used in the construction of block designs by [Al75], [BrHu87], [Ga79], and [Wh78], among others.

R25: The use of voltage graphs in the representation of finite geometries (see §7.9) has been pioneered by A. T. White. See especially [Wh01].

References

[AbPa83] M. Abu-Sbeih and T. D. Parsons, Embeddings of bipartite graphs, *J. Graph Theory* 7 (1983), 325–334.

[Al20] J. W. Alexander, Note on Riemann spaces, *Bull. Amer. Math. Soc.* 26 (1920), 370–372.

[Al75] S. R. Alpert, Twofold triple systems and graph imbeddings, *J. Combin. Theory, Ser. A* 18 (1975), 101–107.

[AlGr76] S. R. Alpert and J. L. Gross, Components of branched coverings of current graphs, *J. Combin. Theory, Ser. B* 20 (1976), 283–303.

[AmLi02] A. Amit and N. Linial, Random graph coverings I: General theory and graph connectivity, *Combinatorica* 22 (2002), 1–18.

[AnGa91] D. Angluin and A. Gardiner, Finite common coverings of pairs of regular graphs, *J. Combin. Theory, Ser. B* 30 (1981), 184–187.

[Ar92] D. Archdeacon, The medial graph and voltage-current duality, *Discrete Math.* 104 (1992), 111–141.

[ArGvSi97] D. Archdeacon, P. Gvozdjak, and J. Širaň, Constructing and forbidding automorphisms in lifted maps, *Math. Slovaca* 47 (1997), 113–129.

[ArKwLeSo00] D. Archdeacon, J. H. Kwak, J. Lee, and M. Y. Sohn, Bipartite covering graphs, *Discrete Math.* 214 (2000), 51–63.

[ArRiSiSk94] D. Archdeacon, R. B. Richter, J. Širaň, and M. Škoviera, Branched coverings of maps and lifts of map homomorphisms, *Australas. J. Combin.* 9 (1994), 109–121.

[Bä38] F. Bäbler, Über die Zerlegung regulärer Streckencomplexe ungerader Ordnung, *Comment. Math. Helv.* 10 (1938), 275–287.

[BrHu87] N. Brand and W. C. Huffman, Constructions and topological invariants of 2-(upsilon, 3, lambda) designs with group-actions, *Eur. J. Combin.* 8 (1987), 5–26.

[EwHo93] R. Ewen and M. Hofmeister, On coverings of the complete graph with 4 vertices, *Ars Combinatoria* 35 (1993), 87–96.

[FeKwKiLe98] R. Q. Feng, J. H. Kwak, J. Kim, and J. Lee, Isomorphism classes of concrete graph coverings, *SIAM J. Discrete Math.* 11 (1998), 265–272.

[FeKw02] Y. Q. Feng and J. H. Kwak, Constructing an infinite family of cubic 1-regular graphs, *Eur. J. Combin.* 23 (2002), 559–565.

[Ga79] B. L. Garman, Voltage graph embeddings and the associated block designs, *J. Graph Theory* 3 (1979), 53–67.

[Gr74] J. L. Gross, Voltage graphs, *Discrete Math.* 9 (1974), 239–246.

[Gr77] J. L. Gross, Every regular graph of even degree is a Schreier coset graph, *J. Combin. Theory, Ser. B* 22 (1974), 227–232.

[GrAl74] J. L. Gross and S. R. Alpert, The topological theory of current graphs, *J. Combin. Theory, Ser. B* 17 (1974), 218–233.

[GrCh96] J. L. Gross and J. Chen, Algebraic specification of interconnection networks by permutation voltage graph morphisms, *Mathematical Systems Theory* 29 (1996), 451–470.

[GrJaPaPi82] J. L. Gross, B. Jackson, T. Pisanski, and T. D. Parsons, Wrapped coverings and their realizations, *Congressus Numerantium* 35 (1982), 333–347.

[GrLo80] J. L. Gross and S. J. Lomonaco, A determination of the toroidal k-metacyclic groups, *J. Graph Theory* 4 (1980), 165–172.

[GrTu74] J. L. Gross and T. W. Tucker, Quotients of complete graphs: revisiting the Heawood map-coloring problem, *Pacific J Math.* 55 (1974), 391–402.

[GrTu77] J. L. Gross and T. W. Tucker, Generating all graph coverings by permutation voltage assignments, *Discrete Math. 18 (1977)*, 273–283.

[GrTu79] J. L. Gross and T. W. Tucker, Fast computations in voltage graph theory, *Ann. NY Acad. Sci.* 319 (1979), 254–257.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover, 2001. (First Edition, Wiley-Interscience, 1987.)

[GrYe99] J. L. Gross and J. Yellen, *Graph-Theory and Its Applications*, CRC Press, 1999.

[Gu63], W. Gustin, Orientable emedding of Cayley graphs, *Bull. Amer. Math. Soc.* 69 (1963), 272–275.

[He:1890] P. J. Heawood, Map-colour theorem, *Quart. J. Math.* 24 (1890), 332–338.

[Ho91a] M. Hofmeister, Concrete graph covering projections, *Ars Combinatoria* 32 (1991), 121–128.

[Ho91b] M. Hofmeister, Isomorphisms and automorphisms of graph coverings, *Discrete Math.* 98 (1991), 175–183.

[Ho95a] M. Hofmeister, Enumeration of concrete regular covering projections, *SIAM J. Discrete Math.* 8 (1995), 51–61.

[Ho95b] M. Hofmeister, Graph covering projections arising from finite vector-spaces over finite-fields, *Discrete Math.* 143 (1995), 87–97.

[HoKw93] S. P. Hong and J. H. Kwak, Regular fourfold coverings with respect to the identity automorphism, *J. Graph Theory* 17 (1993), 621–627.

[HoKwLe96] S. P. Hong, J. H. Kwak, and J. Lee, Regular graph coverings whose covering transformation groups have the isomorphism extension property, *Discrete Math.* 148 (1996), 85–105.

[HoKwLe99] S. P. Hong, J. H. Kwak, and J. Lee, Bipartite graph bundles with connected fibres, *Bull. Austral. Math. Soc.* 59 (1999), 153–161.

[JaPaPi81] B. Jackson, T. D. Parsons, and T. Pisanski, A duality theorem for graph embeddings, *J. Graph Theory* 5 (1981), 55–77.

[JuWh80] M. Jungerman and A. T. White, On the genus of finite abelian groups, *Europ. J. Combin.* 1 (1980), 243–251.

[KiKiLi99] H. K. Kim, J. H. Kim, and D. Lim, Natural isomorphism classes of cycle permutation graphs over a graph, *Graphs and Combinatorics* 15 (1999), 327–336.

[KrPrTe97] J. Kratochvil, A. Proskurowski, and J. A. Telle, Covering regular graphs, *J. Combin. Theory, Ser. B* 71 (1997).

[KwChLe98] J. H. Kwak, J. H. Chun, and J. Lee, Enumeration of regular graph coverings having finite abelian covering transformation groups, *SIAM J. Discrete Math.* 11 (1998), 273–285.

[KwHoLeSo00] J. H. Kwak, S. P. Hong, J. Lee, and M. Y. Sohn, Isoperimetric numbers and bisection widths of double coverings of a complete graph, *Ars Combinatoria* 57 (2000), 49–64.

[KwKiLe96] J. H. Kwak, S. G. Kim, and J. Lee, Distributions of regular branched prime-fold coverings of surfaces, *Discrete Math.* 156 (1996), 141–170.

[KwKw01] J. H. Kwak and Y. S. Kwon, Characteristic polynomials of graph bundles having voltages in a dihedral group, *Linear Algebra and Its Appl.* 336 (2001), 99–118.

[KwLe92] J. H. Kwak and J. Lee, Isomorphism-classes of cycle permutation graphs, *Discrete Math.* 105 (1992), 131–142.

[KwLe94] J. H. Kwak and J. Lee, Enumeration of graph embeddings, *Discrete Math.* 135 (1994), 129–151.

[KwLe98] J. H. Kwak and J. Lee, Distribution of branched D-p-coverings of surfaces, *Discrete Math.* 183 (1998), 193–212.

[KwLe04] J. H. Kwak and J. Lee, Enumeration of graph coverings, Chapter 10 of *Studies in Topological Graph Theory*, ed. by J. L. Gross and T. W. Tucker, Cambridge University Press, 2004.

[KwLeSo99] J. H. Kwak, J. Lee, and M. Y. Sohn, Isomorphic periodic links as coverings, *J. Knot Theory and Its Ramif.* 8 (1999), 215–240.

[LaPe81] G. Lallement and D. Perrin, A graph covering construction of all the finite complete biprefix codes, *Discrete Math.* 36 (1981), 261–271.

[LeKi01] J. Lee and J. W. Kim, Enumeration of the branched mZ(P)-coverings of closed surfaces, *Eur. J. Combin.* 22 (2001), 1125–1138.

[LeKi02] J. Lee and J. W. Kim, On abelian branched coverings of the sphere, *Graphs and Combinatorics* 18 (2002), 329–342.

[Le82] F. T. Leighton, Finite common coverings of graphs, *J. Combin. Theory, Ser. B* 33 (1982), 231–238.

[MaNeSk00] A. Malnic, R. Nedela, and M. Skoviera, Lifting graph automorphisms by voltage assignments, *Eur. J. Combin.* 21 (2000), 927–947.

[MiSa97] H. Mizuno and I. Sato, Characteristic polynomials of some covers of symmetric digraphs, *Ars Combinatoria* 45 (1997), 3–12.

[MiSa00a] H. Mizuno and I. Sato, Isomorphisms of cyclic abelian covers of symmetric digraphs, *Ars Combinatoria* 54 (2000), 51–64.

[MiSa00b] H. Mizuno and I. Sato, Zeta functions of graph coverings, *J. Combin. Theory, Ser. B* 80 (2000), 247–257.

[MiSa02] H. Mizuno and I. Sato, L-functions for images of graph coverings by some operations, *Discrete Math.* 256 (2002), 335–347.

[MiSa03] H. Mizuno and I. Sato, L-functions of regular coverings of graphs, *Eur. J. Combin.* 24 (2003), 321–329.

[Mo85] B. Mohar, Akempic triangulations with 4 odd vertices, *Discrete Math.* 54 (1985), 23–29.

[Mo86] B. Mohar, A common cover of graphs and 2-cell embeddings, *J. Combin. Theory, Ser. B* 40 (1986), 94–106.

[Mo87a] B. Mohar, The enumeration of akempic triangulations, *J. Combin. Theory, Ser. B* 42 (1987), 14–23.

[Mo87b] B. Mohar, Simplicial schemes, *J. Combin. Theory, Ser. B* 42 (1987), 68–86.

[Mo88] B. Mohar, Branched-coverings, *Discrete Comput. Geom.* 3 (1988), 339–348.

[MoRe02] H. S. Mortveit and C. M. Reidys, Towards a calculus of biological networks, *Z. Phys. Chem.* 216 (2002), 235–247.

[NeSk97] R. Nedela and M. Skoviera, Regular maps from voltage assignments and exponent groups, *Eur. J. Combin.* 18 (1997), 807–823.

[Pa80] T. D. Parsons, Circulant graph imbeddings, *J. Combin. Theory, Ser. B* 29 (1980), 310–320.

[PaPiJa80] T. D. Parsons, T. Pisanski, and B. Jackson, Dual imbeddings and wrapped quasi-coverings of graphs, *Discrete Math.* 18 (1980), 43–52.

[Pi80] T. Pisanski, Genus of cartesian products of regular bipartite graphs, *J. Graph Theory* 4 (1980), 31–42.

[Pi82] T. Pisanski, Nonorientable genus of cartesian products of regular graphs, *J. Graph Theory* 6 (1982), 391–402.

[Pi92] T. Pisanski, Orientable quadrilateral embeddings of products of graphs, *Discrete Math.* 109 (1992), 203–205.

[PiTu89] T. Pisanski and T. W. Tucker, The genus of a product of a group with an abelian group, *Europ. J. Combin.* 10 (1989), 469–475.

[PiTu92] T. Pisanski and T. W. Tucker, The genus of low rank Hamiltonian groups, *Discrete Math.* 78 (1989), 157–167.

[PiTuWi92] T. Pisanski, T. W. Tucker, and D. Witte, The nonorientable genus of some metacyclic groups, *Combinatorica* 12 (1), 77–87.

[Pl01] L. Plachta, Voltage graphs, weight systems and odd symmetry, *Discrete Math.* 236 (2001), 287–313.

[Pr77] V. K. Proulx, The classification of toroidal groups, PhD Thesis, Columbia University, 1977.

[Pr81] V. K. Proulx, On the genus of symmetric-groups, *Trans. Amer. Math. Soc.* 266 (1981), 531–538.

[Ri74] G. Ringel, *Map Color Theorem*, Springer-Verlag, 1974.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Natl. Acad. Sci. USA* 60 (1968), 438–445.

[Sa01] I. Sato, Isomorphisms of cyclic abelian covers of symmetric digraphs, III, *Ars Combinatoria* 61 (2001), 173–186.

[Sa94] I. Sato, Isomorphisms of some graph coverings, *Discrete Math.* 128 (1994), 317–326.

[SiSk85] J. Siran and M. Skoviera, Quotients of connected regular graphs of even degree, *J. Combin. Theory, Ser. B* 38 (1985). 214–225.

[St79] S. Stahl, Self-dual embeddings of cayley graphs, *J. Combin. Theory, Ser. B* 27 (1979), 92–107.

[StWh76] S. Stahl and A. T. White, Genus embeddings for some tripartite graphs, *Discrete Math.* 14 (1976), 279–296.

[Su90] D. B. Surowski, Lifting map automorphisms and Macbeath theorem, *J. Combin. Theory, Ser. B* 50 (1990), 135–149.

[Tu36] A. W. Tucker, Branched and folded coverings, *Bull. Amer. Math. Soc.* 42 (1936), 859–862.

[Tu80] T. W. Tucker, Number of groups of a given genus, *Trans. Amer. Math. Soc.* 258 (1980), 167–179.

[Tu83] T. W. Tucker, Finite-groups acting on surfaces and the genus of a group, *J. Combin. Theory, Ser. B* 34 (1983), 82–98.

[Tu84] T. W. Tucker, A refined Hurwitz theorem for imbeddings of irredundant Cayley-graphs, *J. Combin. Theory, Ser. B* 36 (1984), 244–268.

[Wh78] A. T. White, Block designs and graph imbeddings, *J. Combin. Theory, Ser. B* 25 (1978), 166–183.

[Wh01] A. T. White, *Graphs of Groups on Surfaces*, North-Holland, 2001.

7.5 THE GENUS OF A GROUP

Thomas W. Tucker, Colgate University

7.5.1 Symmetric Imbeddings of Cayley Graphs
7.5.2 The Riemann-Hurwitz Equation and Hurwitz's Theorem
7.5.3 Groups of Low Genus
7.5.4 Genus for Families of Groups
7.5.5 Nonorientable Surfaces
References

Introduction

When a Cayley graph $C(\mathcal{A}, X)$ for a finite group $\mathcal{A}$ is imbedded in a surface, the face boundaries, as cycles in the Cayley graph, give relations in the generating set X, that is, words in the generators and their inverses that represent the identity element of $\mathcal{A}$. Thus, the possible imbeddings of Cayley graphs for the group $\mathcal{A}$ are closely related to the possible presentations for that group in terms of generators and relations. The smallest genus g such that some Cayley graph for the group $\mathcal{A}$ can be imbedded in the surface of genus g is called the *genus of the group* $\mathcal{A}$. White [Wh72,Wh84] first introduced the term, although Burnside [Bu11] considers a closely related concept. Study of the genus of groups is closely related to questions about group actions on surfaces, regular branched coverings, and automorphisms of Riemann surfaces.

7.5.1 Symmetric Imbeddings of Cayley Graphs

If the Cayley graph $C(\mathcal{A}, X)$ is imbedded in an orientable surface S, one might ask whether the natural vertex-transitive symmetry of the Cayley graph is somehow reflected in the symmetry of its imbedding, especially for minimum genus imbeddings.

DEFINITIONS

D1: The ***Cayley graph*** $C(\mathcal{A}, X)$ for a group $\mathcal{A}$ with generating set X has the elements of $\mathcal{A}$ as vertices and has edges directed from a to ax for every $a \in \mathcal{A}$ and $x \in X$. We will assume that vertices are labeled by elements of $\mathcal{A}$ and that edges are labeled by elements of X.

We notice that an involution x gives rise to a directed edge from a to ax and also one from ax to a, for all a; sometimes we will choose to identify these pair of edges to a single undirected edge labeled x.

D2: The ***genus of a group*** $\mathcal{A}$, which is denoted $\gamma(\mathcal{A})$, is the smallest genus g such that some Cayley graph for the group $\mathcal{A}$ can be imbedded in the orientable surface S_g of genus g.

D3: The ***natural action*** of the group $\mathcal{A}$ on the Cayley graph $C(\mathcal{A}, X)$ is the group of automorphisms corresponding to left multiplication of the vertices of a Cayley graph $C(\mathcal{A}, X)$ by an element b of the group $\mathcal{A}$. (This respects the labeling and directing of the edges, since $(ba)x = b(ax)$.)

D4: The action of an automorphism group on a graph is ***vertex-transitive*** if it takes any vertex to any other vertex.

D5: The action of an automorphism group on a graph is ***free*** if each nonidentity element leaves no vertex fixed.

D6: The finite group $\mathcal{A}$ ***acts*** on the orientable surface S if $\mathcal{A}$ is isomorphic to a subgroup of the group of all homeomorphisms of S.

D7: The action of a group $\mathcal{A}$ on a surface S ***preserves orientation*** if every element of the corresponding subgroup of homeomorphisms on S preserves the orientation of S.

D8: An imbedding of a Cayley graph $C(\mathcal{A}, X)$ in the orientable surface S is ***symmetric*** if the natural action of $\mathcal{A}$ on $C(\mathcal{A}, X)$ extends to an action of $\mathcal{A}$ on S.

D9: An imbedding of a Cayley graph $C(\mathcal{A}, X)$ in the orientable surface S is ***strongly symmetric*** if the natural action of $\mathcal{A}$ on $C(\mathcal{A}, X)$ extends to an orientation-preserving action of $\mathcal{A}$ on S.

D10: The ***symmetric genus*** (respectively, ***strong symmetric genus***) of the group $\mathcal{A}$, denoted $\sigma(\mathcal{A})$ (respectively, $\sigma^o(\mathcal{A})$), is the smallest g such that some Cayley graph imbeds symmetrically (respectively, strongly symmetrically) in the surface of genus g.

TERMINOLOGY: A strongly symmetric imbedding of a Cayley graph for the group $\mathcal{A}$ is also called a ***Cayley map*** for the group $\mathcal{A}$ [BiWh79].

FACTS

F1: The definitions immediately imply that $\gamma(\mathcal{A}) \leq \sigma(\mathcal{A}) \leq \sigma^o(\mathcal{A})$.

F2: The natural action of the group $\mathcal{A}$ on the Cayley graph $C(\mathcal{A}, X)$ is vertex-transitive and free.

F3: [Sa58] A graph G is a Cayley graph for the group $\mathcal{A}$ if and only if there is a group of automorphisms isomorphic to $\mathcal{A}$ acting on G, such that the action is vertex-transitive and free.

F4: [GrTu87] Any orientation-preserving automorphism of a graph imbedding must respect rotations at vertices (the cyclic ordering of edges around each vertex given by the imbedding), and the natural action of a group on a Cayley graph respects labels. Thus, if an imbedding of a Cayley graph is strongly symmetric, then it must have the same cyclic ordering of generators and their inverses at every vertex, and conversely.

F5: It is a corollary to Fact 4 that to specify a symmetric imbedding of the Cayley graph $C(\mathcal{A}, X)$, all we need do is to give a cyclic ordering of the elements of X and their inverses.

F6: The derived graph of an imbedded voltage graph for a bouquet of circles, where the assigned voltage set X generates the voltage group $\mathcal{A}$, gives a strongly symmetric imbedding of the Cayley graph $C(\mathcal{A}, X)$. Every strong symmetric imbedding of a Cayley graph can be obtained this way.

F7: [GrTu87] Any orientation-reversing automorphism of a graph imbedding reverses the rotations at vertices. If the action of the group $\mathcal{A}$ on the orientable surface S does not preserve orientation, then the set of elements of $\mathcal{A}$ that do preserve orientation form an index-two subgroup $\mathcal{B}$ of $\mathcal{A}$. Thus if an imbedding of a Cayley graph for $\mathcal{A}$ is symmetric but not strongly symmetric, there is a subgroup $\mathcal{B}$ of index two in $\mathcal{A}$ such that all vertices in $\mathcal{B}$ have the same rotation and all vertices not in $\mathcal{B}$ have the opposite rotation, and conversely.

F8: Viewing symmetric imbeddings that are not strongly symmetric as derived graphs of a small (one or two vertices) voltage graph tends to be complicated. One possibility is to begin with an imbedded voltage graph of a bouquet of circles in a nonorientable surface and hope that the derived surface is orientable. Another is to begin with a two-vertex graph imbedded in a symmetric surface having an orientation-reversing involution f that interchanges the two vertices.

F9: [GrTu87] If the voltage group $\mathcal{A}$ has an index two subgroup $\mathcal{B}$, such that all loops are assigned voltages in $\mathcal{B}$, such that all edges between the two vertices are assigned voltages not in $\mathcal{B}$, and such that e and $f(e)$ get the same voltage, then the derived graph will be a symmetric, but not strongly symmetric imbedding of a Cayley graph for $\mathcal{A}$.

F10: [Tu83] Any action of the finite group $\mathcal{A}$ on the orientable surface S comes from a symmetric imbedding of a Cayley graph $C(\mathcal{A}, X)$ in S. If the action preserves orientation, then the imbedding is strongly symmetric. Thus $\sigma(\mathcal{A})$, respectively $\sigma^o(\mathcal{A})$, is the minimal g such that $\mathcal{A}$ acts, respectively, acts preserving orientation, on a surface of genus g.

F11: If $\mathcal{B}$ is a subgroup of $\mathcal{A}$, then

$$\sigma(\mathcal{B}) \le \sigma(\mathcal{A}) \quad \text{and} \quad \sigma^o(\mathcal{B}) \le \sigma^o(\mathcal{A})$$

since if $\mathcal{A}$ acts on a surface, then so does $\mathcal{B}$.

F12: [Ba77] If $\mathcal{B}$ is a subgroup of $\mathcal{A}$, then any Cayley graph for $\mathcal{A}$ edge-contracts to a Cayley graph for $\mathcal{B}$. In particular, $\gamma(\mathcal{B}) \le \gamma(\mathcal{A})$.

EXAMPLES

E1: View the standard cube as having a top and bottom face with four vertical faces. Let y denote the rotation by 90 degrees about the centers of the top and bottom faces. Let x denote the reflection that interchanges the top and bottom faces but takes each vertical face to itself. It is not hard to see that the action generated by the symmetries x and y is vertex-transitive and free, that y preserves orientation and x does not, and that $xy = yx$. Thus, the vertices and edges of the cube can be labeled to give a symmetric, but not strongly symmetric imbedding of a Cayley graph for the abelian group

$$\langle x, y : x^2 = y^4 = 1, xy = yx \rangle \cong \mathcal{Z}_2 \times \mathcal{Z}_4$$

E2: Let y be as in the previous example, but let z denote the rotation by 180 degrees about the midpoint of a vertical edge; then z interchanges not only the top and bottom faces, but also the vertical faces in pairs as well. It is again not hard to see that the action generated by the symmetries z and y is vertex-transitive and free, that it preserves orientation, and that $zyz = y^{-1}$. Thus, the vertices and edges of the cube can also be labeled to give a strongly symmetric imbedding of a Cayley graph for the group

$$\langle z, y : z^2 = y^4 = 1, zyz = y^{-1} \rangle$$

which is the dihedral group of order 8.

REMARKS

R1: In both examples above, the vertical edges correspond to an involution, with the resulting pairs of edges identified. In the first example, the involution reversed orientation; it is not hard to check that the vertical edges could not be replaced by a pair of directed edges and still have the involution x respect the directions. In the second example, the vertical edges could be replaced by a pair of directed edges and still have the directions respected by the involution z. In general, pairs of edges in a symmetric imbedding of a Cayley graph corresponding to an involution that reverses orientation must be identified but need not be otherwise.

R2: Orientation-reversing homeomorphisms of finite order, such as involutions, can be nonintuitive and subtle. One tends to think in terms of euclidean isometries, where there are two types: reflections and glides. For example, one can imagine cutting a torus in half, forming two dividing circles, and interchanging the halves by a reflection that leaves fixed the dividing circles. But it is also possible to interchange the halves by an antipodal map that also interchanges the circles. It is even possible to interchange the halves, leaving one dividing circle fixed, but rotating the other dividing circle a half turn (like a glide along the circle). See [GrTu87] for more examples.

7.5.2 The Riemann-Hurwitz Equation and Hurwitz's Theorem

Given a voltage graph imbedded in a surface S, the Euler characteristic for the surface of the derived imbedding is easily calculated, in terms of the order of the voltage group, the Euler characteristic of S, and the order of the net voltages on the faces. As derived imbeddings of one-vertex imbedded voltage graphs, strongly symmetric imbeddings of Cayley graphs can be handled the same way.

DEFINITIONS

D11: Suppose that the Cayley graph $C(\mathcal{A}, X)$ has a strongly symmetric imbedding in the surface T as the derived graph for a voltage graph imbedding of the bouquet B in the surface S. Suppose that the non-identity net voltages on the faces of this imbedding are $r_1, \cdots, r_n$. Then the Euler characteristic $\chi(T)$ can be computed by the ***Riemann-Hurwitz equation*** (where $|\mathcal{A}|$ is the order of the group $\mathcal{A}$):

$$\chi(T) = |\mathcal{A}|(\chi(S) - \sum \left(1 - \frac{1}{r_i}\right)$$

D12: The quantity $1 - \dfrac{1}{r_i}$ is sometimes called the ***deficiency of the branch point*** at the center of a face (of an imbedded voltage graph) whose excess voltage has order r_i. That is because there are only $|\mathcal{A}|/r_i$ copies of that face in the derived imbedding, instead of the "expected number" $|\mathcal{A}|$ copies, which is a deficiency of $|\mathcal{A}| - \dfrac{|\mathcal{A}|}{r_i}$. (Since vertices and edges generate $|\mathcal{A}|$ copies each, this explains the Riemann-Hurwitz equation.)

D13: A similar equation holds for symmetric imbeddings that are not strongly symmetric. If the associated imbedded voltage graph still has one vertex, then the surface

S is nonorientable, but the equation holds exactly as before. If the associated imbedded voltage graph has two vertices, then the ***Riemann-Hurwitz equation*** becomes

$$\chi(T) = \frac{|\mathcal{A}|}{2}(\chi(S) - \sum 1 - 1/r_i)$$

D14: A ***triangle group*** is a finite group with presentation

$$\langle x, r : x^p = y^q = (xy)^r = 1\rangle$$

This is the group of isometries of the plane generated by the rotations at the vertices of a triangle with angles $\pi/p, \pi, q, \pi/r$ (the geometry of the plane is spherical, Euclidean or hyperbolic, depending on whether the angle sum is greater than, equal to, or less than π).

D15: Any *quotient group* $\mathcal{A}$ of the triangle group $\langle x, r : x^p = y^q = (xy)^r = 1\rangle$ is said to be a ***(p,q,r)^o group***.

D16: The ***full triangle group*** has presentation

$$\langle x, y, z : x^2 = y^2 = z^2 = (xy)^r = (yz)^q = (xz)^r = 1\rangle$$

and is generated by reflections in the sides of a $\pi/p, \pi/q, \pi/r$ triangle.

D17: Any *quotient group* $\mathcal{A}$ of the full triangle group $\langle x, y, z : x^2 = y^2 = z^2 = (xy)^r = (yz)^q = (xz)^r = 1\rangle$ is said to be a ***(p,q,r) group***.

D18: A (p, q, r) group $\mathcal{A}$, with generators as above, is ***properly*** (p, q, r) if the subgroup generated by xy and yz has index two in $\mathcal{A}$. (The index is otherwise one.)

FACTS

F13: If $\mathcal{A}$ is a $(2, q, r)^o$ group, then by the Riemann-Hurwitz equation it has a strongly symmetric imbedding in a surface of Euler characteristic

$$\chi = |\mathcal{A}|\left[2 - \left(1 - \frac{1}{2}\right) - \left(1 - \frac{1}{p}\right) - \left(1 - \frac{1}{q}\right)\right] = |\mathcal{A}|\left[\frac{1}{p} + \frac{1}{q} - \frac{1}{2}\right]$$

and genus $g = 1 - \frac{\chi}{2}$ satisfying

$$g - 1 = \frac{|\mathcal{A}|}{2}\left[\frac{1}{2} - \frac{1}{p} - \frac{1}{q}\right]$$

F14: Similarly, if $\mathcal{A}$ is properly $(2, q, r)$, then it has a symmetric imbedding in a surface of genus g satisfying

$$g - 1 = \frac{|\mathcal{A}|}{4}\left[\frac{1}{2} - \frac{1}{p} - \frac{1}{q}\right]$$

F15: (Hurwitz's Theorem [GrTu87], [Hu: 1892], [Tu80]) If the group $\mathcal{A}$ has strong symmetric genus $\sigma^o(\mathcal{A}) > 1$, then

$$|\mathcal{A}| \leq 84(\sigma^o(\mathcal{A}) - 1)$$

with equality if and only if $\mathcal{A}$ is $(2, 3, 7)^o$. If the symmetric genus $\sigma(\mathcal{A}) > 1$, then $|\mathcal{A}| \leq 168(\sigma(\mathcal{A}) - 1)$, with equality if and only if $\mathcal{A}$ is properly $(2, 3, 7)$.

F16: [Tu80] As a corollary to Fact 15, there are only finitely many groups of a given symmetric genus or strong symmetric genus greater than one.

F17: [Tu80] (a Cayley-graph version of Hurwitz's theorem) If $\gamma(\mathcal{A}) > 1$, then

$$|\mathcal{A}| \leq 168(\gamma(\mathcal{A}) - 1)$$

with equality if and only if $\mathcal{A}$ is properly $(2,3,7)$.

F18: [Tu80] As a corollary to Fact 17, there are only finitely many groups of a given genus greater than one.

F19: [Ba95], [Th95] **Babai-Thomassen Theorem**: There are only finitely many vertex-transitive graphs of a given genus $g > 2$. In particular, there are only finitely many Cayley graphs of a given genus $g > 2$.

REMARKS

R3: The Hurwitz theorems have been stated here as an upper bound on $|\mathcal{A}|$, rather than as a lower bound on $\sigma(\mathcal{A})$ or $\gamma(\mathcal{A})$, since the traditional view was bounding the order of a group of conformal automorphisms on a Riemann surface of given genus. The proof of Hurwitz's original theorem is a brief analysis of the possibilities for the Riemann-Hurwitz equation when $\chi(T) < 0$. The same analysis can be refined to give detailed information about $\mathcal{A}$, whenever $|\mathcal{A}|$ is large compared to $\sigma(\mathcal{A})$ or $\sigma^o(\mathcal{A})$. For example, if $|\mathcal{A}| > 80(\sigma(\mathcal{A})-1)$, then $\mathcal{A}$ is $(2,3,7)^o$, properly $(2,3,7)$, or properly $(2,3,8)$. For more on refinements of Hurwitz's Theorem see [GrTu87], [Tu83].

R4: The Cayley graph version of Hurwitz's theorem is unexpected. The formula for the Euler characteristic guarantees that both the valence and average face size must be small, when the number of vertices of an imbedded graph is large compared to the Euler characteristic of the imbedding surface. For a Cayley graph, this means there must be a small number of generators and many short relators. The proof is a long, exhaustive case-by-case analysis with lots of "relation chasing". This analysis can be refined to give detailed information about $\mathcal{A}$, whenever $|\mathcal{A}|$ is large, in a manner analogous to Hurwitz's theorem, although again the proofs are much harder and longer (see [GrTu87], [Tu84b]).

7.5.3 Groups of Low Genus

For low genus, minimum imbeddings tend to be highly symmetric. For example, by Whitney's theorem that a 3-connected planar graph imbeds uniquely in the sphere, a Cayley graph imbedded in the sphere must be symmetrically imbedded. In addition, symmetries of the sphere and torus come from the natural geometry of the surfaces: spherical geometry for the sphere, and Euclidean geometry for the torus (viewed as the Euclidean plane rolled up by a pair of linearly independent translations).

DEFINITION

D19: A ***Euclidean space group*** or ***Euclidean crystallographic group*** is a group of isometries of the Euclidean plane that contains translations in independent directions and such that the orbit of any point under the group has no accumulation points (there is a minimum distance any point is moved by all the elements of the group that does not leave the point fixed).

NOTATION: The dihedral group of order $2n$ is denoted $\mathcal{D}_n$. The symmetric group and alternating group on n symbols are denoted $\mathcal{S}_n$ and $\mathcal{A}_n$, respectively.

FACTS

Finding all groups of a given small genus has a long history.

F20: There are exactly 17 Euclidean space groups, up to isomorphism, and presentations for the groups are well-known [CxMo80].

F21: (Planar groups [Ma96], [GrTu87]) The groups of strong symmetric genus 0 are $\mathcal{Z}_n$, $\mathcal{D}_n$, $\mathcal{A}_4$, $\mathcal{S}_4$, and $\mathcal{A}_5$. The groups of symmetric genus 0 are these groups together with their direct products with $\mathcal{Z}_2$. In both cases, the associated group actions can be realized by automorphisms of prisms and the platonic solids. For all groups, $\gamma(\mathcal{A}) = 0$ if and only if $\sigma(\mathcal{A}) = 0$.

F22: (Toroidal groups [Ba26], [Pr77], [Tu84a], [GrTu87]) Except for three groups, $\gamma(\mathcal{A}) = 1$ if and only if $\sigma(\mathcal{A}) = 1$. Moreover, $\sigma(\mathcal{A}) = 1$ if and only if $\mathcal{A}$ is a finite quotient of one of the 17 Euclidean space groups; this yields partial presentations for all toroidal groups (see [CxMo80] or [GrTu87]). The three exceptional groups, with $\gamma(\mathcal{A}) = 1$ but $\sigma(\mathcal{A}) > 1$ have orders 24, 48, and 48 and presentations

(a) $\mathcal{A} = \langle x, y : x^3 = y^3 = 1, xyx = yxy\rangle$,

(b) $\mathcal{A} = \langle x, y : x^3 = y^2 = 1, xyxyxy = yxyxyx\rangle$,

(c) $\mathcal{A} = \langle x, y : x^3 = y^2 = 1, (xyxyx^{-1})^2 = 1\rangle$.

F23: (Genus two [Tu84c]) There is exactly one group $\mathcal{A}$ such that $\gamma(\mathcal{A}) = 2$. It has order 96 and the $(2, 3, 8)$ presentation:

$$\mathcal{A} = \langle x, y, z : x^2 = y^2 = z^2 = 1, (xy)^2 = (yz)^3 = (xz)^8 = 1, (xy)^4 z = z(xy)^4\rangle$$

F24: (Symmetric genus 2 [MyZi95]) There are 4 groups of symmetric genus 2: the group of genus 2 and the three exceptional groups of genus 1.

F25: (Symmetric genus 3 [MyZi97]) There are 3 groups of symmetric genus 3: the proper $(2, 3, 7)$ group $PGL(2, 7)$, its $(2, 3, 7)^o$ subgroup $PSL(2, 7)$ (also known as Klein's simple group of order 168), and the proper $(2, 4, 6)$ group $\mathcal{Z}_2 \times \mathcal{Z}_2 \times \mathcal{S}_4$.

F26: (Strong symmetric low genus [MyZi00]) There are 6 groups of strong symmetric genus 2, 10 groups of strong symmetric genus 3, and 10 groups of strong symmetric genus 4.

7.5.4 Genus for Families of Groups

Abelian groups form an interesting case study for the genus of a group. Commutators in the generators provide many possible faces of size four (quadrilaterals), but it is not easy to see how to combine them to form an all-quadrilateral imbedding. In fact, a strongly symmetric imbedding of a Cayley graph of valence greater than 4 cannot have a face of the form $xyx^{-1}y^{-1}$: the rotation would have to be $xy^{-1}x^{-1}y$ with no room for any other generators. The canonical form for the abelian group plays a key role. The simplest case, when all the factors in the canonical form are even, was part of White's original paper [Wh72] where he introduced the genus of a group. At the other extreme are groups whose minimal genus imbeddings are symmetric. For example, any proper $(2, 3, 7)$ group $\mathcal{A}$ has genus $1 + |\mathcal{A}|/168$, by the Cayley graph version of Hurwitz's Theorem.

DEFINITIONS

D20: Any abelian group $\mathcal{A}$ can be written uniquely in the ***canonical form***

$$\mathcal{Z}_{m_1} \times \mathcal{Z}_{m_2} \times \cdots \mathcal{Z}_{m_r} \quad \text{where } m_j | m_{j+1} \text{ for } j = 1 \cdots r$$

D21: The factors $\mathcal{Z}_{m_i}$ in the canonical form of an abelian group $\mathcal{A}$ are called the ***canonical factors*** of $\mathcal{A}$, and the number r of factors is called the ***rank*** of $\mathcal{A}$.

D22: A $(2,3,7)^0$ group is called a ***Hurwitz group***.

D23: A proper $(2,3,7)$ group is called a ***proper Hurwitz group***.

FACTS

F27: [JuWh80] Suppose that the abelian group $\mathcal{A}$ does not have $\mathcal{Z}_3$ as a canonical factor and that m_1 is even if the rank $r = 3$. Then $\gamma(\mathcal{A}) = 1 + |\mathcal{A}|(r-2)/4$, whenever the right side of this equation is an integer.

F28: [BrSq94], [MoPiSkWh87] $\gamma(\mathcal{Z}_3 \times \mathcal{Z}_3 \times \mathcal{Z}_3) = 7$, and the minimal genus imbedding has very little symmetry, including faces of many different sizes.

F29: [PiTu89] Let $\mathcal{C}$ be any finite group. If $\mathcal{A}$ is an abelian group of rank r at least twice that of $\mathcal{C}$, then in most cases $\gamma(\mathcal{C} \times \mathcal{A}) = 1 + |\mathcal{C}||\mathcal{A}|(r-2)/4$.

F30: [Mc70], [MyZi93] The strong symmetric and symmetric genus are known for all abelian groups.

F31: [Co80] For $n > 167$, the symmetric group S_n is a proper Hurwitz group. In particular, $\gamma(S_n) = 1 + n!/168$ for $n > 167$.

F32: All the 26 sporadic simple groups are $(2, p, q)$ for some p and q; and 12 of them are Hurwitz groups. All the alternating groups $\mathcal{A}_n$ for $n > 167$ and many of the simple groups of Lie type and of large enough dimension are Hurwitz groups [LuTa99].

F33: The symmetric genus of all the sporadic groups [CoWiWo92], of all the alternating and symmetric groups [Co85], and of many other simple groups is known. Notice that if the group $\mathcal{A}$ is simple, then $\sigma(\mathcal{A}) = \sigma^o(\mathcal{A})$, since $\mathcal{A}$ has no subgroups of index two.

F34: [MyZi02] For every genus g, $\sigma^o(\mathcal{Z}_m \times D_n) = g$ for some m, n. In fact, $\sigma^o(\mathcal{Z}_3 \times D_n) = n$ if n is not divisible by 6.

REMARK

R5: Fact 34 is interesting, because it shows that the strong symmetric genus σ^o is free of gaps. It is still not known whether σ or γ have any gaps.

7.5.5 Nonorientable Surfaces

It is possible, of course, to imbed Cayley graphs in nonorientable surfaces, and it is natural to ask about minimal nonorientable imbeddings. It is also interesting to compare minimal imbeddings in orientable and nonorientable surfaces.

DEFINITIONS

D24: If S is a nonorientable surface of Euler characteristic $\chi(S)$, then $2-\chi(S)$ is called the ***crosscap number*** or ***nonorientable genus*** of S. (It follows from the classification of closed surfaces that every surface of crosscap number c can be obtained from the sphere by attaching c crosscaps, that is by removing c disks from the sphere and identifying the resulting c boundary components to the boundaries of c Möbius strips.)

D25: For any closed surface S, the quantity $2-\chi(S)$ is called the ***Euler genus*** of S. Thus if S is orientable of genus g, then its Euler genus is $2g$, and if S is nonorientable of crosscap number c, then its Euler genus is c.

D26: The ***nonorientable genus*** or ***crosscap number*** of the group $\mathcal{A}$, denoted $\tilde{\gamma}(\mathcal{A})$, is the smallest number c such that some Cayley graph for $\mathcal{A}$ imbeds in the nonorientable surface S of crosscap number c.

D27: The ***Euler genus*** of the group $\mathcal{A}$, denoted $\gamma^e(\mathcal{A})$, is the smallest number e such that some Cayley for $\mathcal{A}$ imbeds in a surface of Euler genus e.

D28: The ***symmetric Euler genus*** of the group $\mathcal{A}$, denoted $\sigma^e(\mathcal{A})$, is the smallest number e such that some Cayley graph for $\mathcal{A}$ imbeds symmetrically in a surface of Euler genus e.

D29: An imbedding for a Cayley graph $C(\mathcal{A}, X)$ in a nonorientable surface S is ***symmetric*** if the natural action of $\mathcal{A}$ on $C(\mathcal{A}, X)$ extends to an action on S.

D30: The ***symmetric crosscap number*** or ***symmetric nonorientable genus***, denoted $\tilde{\sigma}(\mathcal{A})$, is the smallest number c such that some Cayley graph for $\mathcal{A}$ imbeds symmetrically in a surface of nonorientable genus c.

D31: An imbedding of a graph in a nonorientable surface can be described by assigning to each vertex a cyclic order or rotation to the set of edges incident to that vertex and assigning to each edge a ***type*** of 0 or 1, telling whether the edge is orientation-preserving or orientation-reversing.

FACTS

F35: An imbedding of a Cayley graph $C(\mathcal{A}, X)$ in a nonorientable surface is symmetric if and only if the rotation is the same at each vertex in terms of the directed edge labels and if every directed edge labeled by the same generator has the same type.

F36: The definitions immediately imply the following:

(i) $\tilde{\gamma}(\mathcal{A}) \leq \tilde{\sigma}(\mathcal{A})$

(ii) $\gamma^e(\mathcal{A}) \leq \sigma^e(\mathcal{A})$

(iii) $\gamma^e(\mathcal{A}) = \min\{2\gamma(\mathcal{A}), \tilde{\gamma}(\mathcal{A})\}$

(iv) $\sigma^e(\mathcal{A}) = \min\{2\sigma(\mathcal{A}), \tilde{\gamma}(\mathcal{A})\}$

F37: Any imbedding of a graph in an orientable surface can be turned into an imbedding in a nonorientable surface, decreasing the number of faces by at most 1, by changing the type of a single edge. Thus $\tilde{\gamma}(\mathcal{A}) \leq 2\gamma(\mathcal{A}) + 1$.

F38: If $\mathcal{B}$ is a subgroup of $\mathcal{A}$, then $\tilde{\gamma}(B) \leq \tilde{\gamma}(\mathcal{A})$, by Babai's Theorem [Ba77].

F39: [Tu83] The group $\mathcal{A}$ has a symmetric imbedding in the nonorientable surface S if and only if $\mathcal{A}$ acts on S. In particular, if $\mathcal{B}$ is a subgroup of $\mathcal{A}$, then $\tilde{\sigma}(B) \leq \tilde{\sigma}(\mathcal{A})$.

F40: Hurwitz's Theorem and its Cayley graph version apply to nonorientable surfaces.

F41: [Tu83] If the group $\mathcal{A}$ acts on the nonorientable surface S of Euler characteristic χ, then $\mathcal{Z}_2 \times \mathcal{A}$ acts on the orientable double covering of S, the surface of Euler characterstic 2χ, with the $\mathcal{A}$ factor orientation-preserving. In particular, $\sigma^o(\mathcal{A}) - 1 \leq (\tilde{\sigma}(\mathcal{A}) - 2)$.

F42: The two groups of Euler genus 1 are $\mathcal{Z}_3 \times \mathcal{Z}_3$ and its $\mathcal{Z}_2$-extension

$$\langle x, y : x^3 = y^2 = 1, [x, yxy] = 1 \rangle$$

There are no groups of symmetric Euler genus 1, since any group acting on the projective plane also acts on its orientable double covering, the sphere. There are no groups of Euler genus 3 and one group of symmetric Euler genus 3. For $\gamma^e = 2$ or $\gamma^e = 4$, all minimal imbeddings are orientable, that is, $\gamma^e(\mathcal{A}) = 2$ if and only if $\gamma(\mathcal{A}) = 1$, and $\gamma^e(\mathcal{A}) = 4$ if and only if $\gamma(\mathcal{A}) = 2$. The only group $\mathcal{A}$ with $\sigma^e(\mathcal{A}) = 5$ is the symmetric group $\mathcal{S}_5$; it is also true that $\gamma^e(\mathcal{S}_5) = 5$ (see [Tu91], [MyZi01]).

F43: The groups with $\tilde{\gamma} = 1$ are the groups with $\gamma = 0$ together with the two groups with $\gamma^e = 1$. The groups with $\tilde{\sigma} = 1$ are the groups with $\sigma^O = 0$. It is conjectured that there are no groups with $\tilde{\gamma} = 2$. Other cases of low crosscap number are studied in [Tu91], [MyZi01].

F44: The crosscap number is known for many abelian groups.

F45: If $\mathcal{A}$ is an improper Hurwitz group, then $\tilde{\gamma}(\mathcal{A}) = 1 + |\mathcal{A}|/84$. In particular, $\tilde{\gamma}(A_n) = 1 + |\mathcal{A}|/84$ for all $n > 167$ [Co85].

References

[Ba77] L. Babai, Some applications of graph contractions, *J. Graph Theory* 1 (1977), 125–130.

[Ba91] L. Babai, Vertex-transitive graphs and vertex-transitive maps, *J. Graph Theory* 15 (1991), 587–627.

[Bk31] R. P. Baker, Cayley diagrams on the anchor ring, *Amer. J. Math.* 53 (1931), 645–669.

[BiWh79] N. Biggs and A. T. White, Permutation groups and combinatorial structures, *Math. Soc. Lect. Notes* 33, Cambridge University Press, 1979.

[BrSq88] M. G. Brin and C. C. Squier, On the genus of $\mathcal{Z}_3 \times \mathcal{Z}_3 \times \mathcal{Z}_3$, *European J. Combin.* 9 (1988), 431–443.

[Bu11] W. Burnside, *Theory of Groups of Finite Order*, Cambridge University Press, 1911.

[Co85] M. Conder, The symmetric genus of alternating and symmetric groups, *J. Combin. Theory, Ser. B* 39 (1985), 179–186.

[Co90] M. D. E. Conder, Hurwitz groups: a brief survey, *Bull. Amer. Math. Soc.* 23 (1990), 359–370.

[CoWiWo92] M. D. E. Conder, R. A. Wilson, and A. J. Woldar, The symmetric genus of sporadic groups, *Proc. Amer. Math. Soc.* 116 (1992), 653–663.

[CxMo80] H. S. M. Coxeter and W. O. J. Moser, *Genererators and Relations for Discrete Groups* (4th ed.), Springer-Verlag, 1980.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Dover, 2001. (First Edition, Wiley-Interscience, 1987.)

[Hu:1893] A. Hurwitz, Über algebraische gebilde mit eindeutigen transformationen in sich, *Math. Ann.* 41 (1893), 403–442.

[JuWh80] M. Jungerman and A. T. White, On the genus of finite abelian groups, *Europ. J. Combin.* 1 (1980), 243–251.

[LuTa99] A. Luchini and M. C. Tamburini, Classical groups of large rank as Hurwitz groups, *J. Algebra* 219 (1999), 531–546.

[Mc65] C. Maclachlin, Abelian groups of automorphisms of compact Riemann surfaces, *Proc. London Math. Soc.* 15 (1965), 699–712.

[Ma:1896] H. Maschke, The representation of finite groups, *Amer. J. Math.* 18 (1896), 156–194.

[MyZi93] C. L. May and J. Zimmerman, The symmetric genus of finite abelian groups, *Illinois J. Math.* 37 (1993), 400–423.

[MyZi95] C. L. May and J. Zimmerman, Groups of small symmetric genus, *Glasgow Math. J.* 37 (1995), 115–129.

[MyZi97] C. L. May and J. Zimmerman, The groups of symmetric genus three, *Houston J. Math.* 23 (1997), 573–590.

[MyZi00] C. L. May and J. Zimmerman, Groups of small strong symmetric genus, *J. Group Theory* 3 (2000), 233–245.

[MyZi01] C. L. May and J. Zimmerman, The group of Euler characteristic -3, *Houston J. Math.* 27 (2001), 737–752.

[MyZi03] C. L. May and J. Zimmerman, There is a group of every strong symmetric genus, *Proc. London Math. Soc.*, to appear.

[MoPiSkWh85] B. Mohar, T. Pisanski, M. Skoviera, and A. T. White, The cartesian product of three triangles can be embedded into a surface of genus 7, *Discrete Math.* 56 (1985), 87–89.

[Pr77] V. K. Proulx, The classification of toroidal groups, PhD Thesis, Columbia University, 1977.

[PiTu89] T. Pisanski and T. W. Tucker, The genus of a product of a group with an abelian group, *Europ. J. Combin.* 10 (1989), 469–475.

[Sa58] G. Sabidussi, On a class of fixed-point free graphs, *Proc. Amer. Math. Soc.* 9 (1958), 800–804.

[Th91] C. Thomassen, Tilings of the torus and the Klein bottle and vertex-transitive graphs on fixed surface, *Trans. Amer. Math. Soc.* 323 (1991), 89–105.

[Tu80] T. W. Tucker, The number of groups of a given genus, *Trans. Amer. Math. Soc.* 258 (1980), 167–179.

[Tu83] T. W. Tucker, Finite groups acting on surfaces and the genus of a group, *J. Combin. Theory, Ser. B* 34 (1983), 82–98.

[Tu84a] T. W. Tucker, On Proulx's four exceptional toroidal groups, *J. Graph Theory* 8 (1984), 29–33.

[Tu84b] T. W. Tucker, A refined Hurwitz theorem for imbeddings of irredundant Cayley graphs, *J. Combin. Theory, Ser. B* 36 (1984), 244–268.

[Tu84c] T. W. Tucker, There is one group of genus two, *J. Combin. Theory, Ser. B* 36 (1984), 269–275.

[Tu91] T. W. Tucker, Symmetric embeddings of Cayley graph in nonorientable surfaces, pp. 1105–1120 in *Graph Theory, Combinatorics, and Applications* (Kalamazoo 1988), Wiley-Interscience, 1991.

[Wh84] A. T. White, *Graphs, Groups and Surfaces*, Revised Edition, North-Holland, 1984.

[Wh72] A. T. White, On the genus of a group, *Trans. Amer. Math. Soc.* 173 (1972), 203–214.

7.6 MAPS

Andrew Vince, University of Florida

7.6.1 Maps and Polyhedra Maps
7.6.2 f-Vector, v- and p-Sequences, and Realizations
7.6.3 Map Coloring
7.6.4 Minimal Maps
7.6.5 Automorphisms and Coverings
7.6.6 Combinatorial Schemes
7.6.7 Symmetry of Maps
7.6.8 Enumeration
7.6.9 Paths and Cycles in Maps
References

Introduction

The theory of maps is likely the oldest topic in this volume, going back, not just to the 4-color problem posed in 1852 and to the theory of automorphic functions developed in the late 1800's, but to the Platonic solids dating to antiquity. Among the many contributors to the subject are Archimedes, Kepler, Euler, Poinsot, de Morgan, Hamilton, Dyck, Klein, Heawood, Hurwitz, Steinitz, Whitney, Koebe, Tutte, Coxeter and Grünbaum. General references on maps include [BoLi95], [BrSc97], [CoMo57], [GrTu87], [MoTh01], and [Wh01].

7.6.1 Maps and Polyhedra Maps

Basic notions are introduced: map and polyhedral map, duality, isomorphism, face and edge-width. The existence and uniqueness of a map with a given graph is addressed.

DEFINITIONS

D1: A ***map*** M ***on a surface*** S is a finite cell-complex whose underlying topological space is S. The surface of a map M is denoted $|M|$.

D2: The ***graph of the map*** M is its 1-skeleton. It is denoted $G := G(M)$.

D3: The ***vertices*** and ***edges of a map*** M are the vertices and edges, respectively, of its graph $G(M)$.

D4: The ***faces of a map*** M are the connected components of $|M| \setminus G(M)$.

D5: The ***0-, 1-, and 2-dimensional faces of a map*** M are its vertices, edges and faces, respectively.

D6: The ***dual map*** M^* of a map M on a surface S is a map on the same surface S whose vertex set V^* consists of one point interior to each face of M and whose edge set E^* consists of, for each edge e of M, an edge e^* crossing e and joining the vertices of V^* that correspond to the faces incident with e. (A more general definition of duality appears in [Vi95], for example.)

D7: A ***polyhedral map*** M, generalizing the notion of a convex polyhedron, is a map whose face boundaries are cycles, and such that any two distinct face boundaries are either disjoint or meet in either a single edge or vertex.

D8: Maps M_1 and M_2 are ***isomorphic***, denoted $M_1 \approx M_2$, if there is a homeomorphism of the respective surfaces that induces an isomorphism of the respective graphs.

D9: The ***face-width of a map*** M, denoted $fw(M)$, is the minimum number of points $|\tau \cap G(M)|$ over all noncontractible simple closed curves τ on the surface.

D10: The ***edge-width of a map*** M, denoted $ew(M)$, is the length of a shortest cycle in $G(M)$ that is noncontractible on the surface.

D11: A ***large-edge-width (LEW)*** map is a map whose edge-width is greater than the number of edges in any face boundary.

EXAMPLES

E1: A map M on the torus and the dual map M^* appear in Figure 7.6.1. (The torus is obtained by identifying like labeled edges on the boundary of the polygon.) Neither M nor M^* is polyhedral.

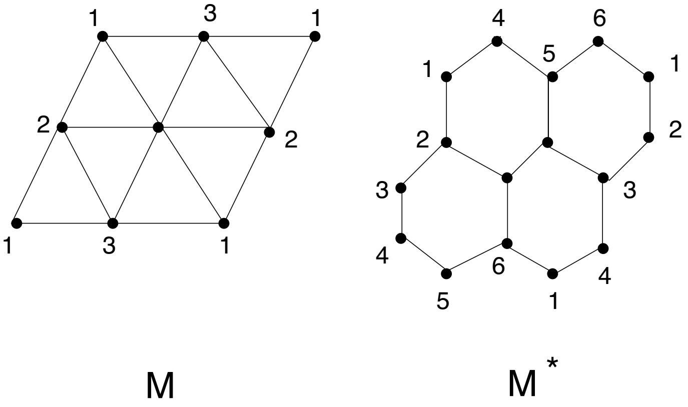


Figure 7.6.1 A torus map and its dual.

E2: Figure 7.6.2 shows two nonisomorphic maps on the sphere that have the same 2-connected but not 3-connected, graph. The maps are related by a *Whitney flip.* This example is relevant to Fact 6 below.

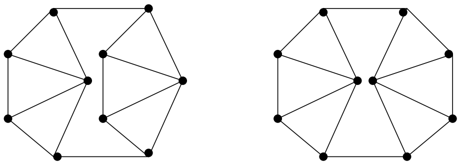


Figure 7.6.2 Maps on the sphere with the same 2-connected graph.

E3: Figure 7.6.3 shows two polyhedral maps on the projective plane with isomorphic 3-connected graphs. (The projective plane is depicted as a disc with antipodal points identified.) This example shows that the analogy to the Whitney uniqueness theorem (Fact 6) for projective planar graphs fails.

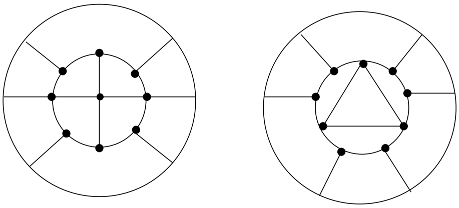


Figure 7.6.3 **Maps on the projective plane with the same 3-connected graph.**

REMARKS

R1: It is equivalent to regard a map as a 2-***cell imbedding*** of a graph G on a surface S, i.e., an imbedding such that the connected components of $S \setminus G$ are 2-cells.

R2: Face-width, introduced in [RoSe88], is a measure of locally planarity, or of how dense the graph is on the surface, or of how well the graph represents the surface.

R3: The concept of map has been extended to cell-complexes whose underlying topological space is a manifold of dimension greater than 2. This includes, in particular, the boundary complex of any polytope. The generalization to higher dimensions, though natural and interesting, is omitted here.

R4: A map M on the sphere S can be drawn in the plane via, for example, stereographic projection from any point of $S \setminus G(M)$.

R5: A map may have multiple edges, self-loops, and vertices of degree 1 or 2. A polyhedral map, however, can have none of these. Moveover, in a polyhedral map, the closure of each face is topologically a closed disc.

FACTS

F1: **Euler's formula** For any map M with f_0 vertices, f_1 edges, f_2 faces and characteristic $c(M)$,

$$f_0 - f_1 + f_2 = c(M)$$

F2: If M is a map, then $(M^*)^* = M$.

F3: If M is a map, then $fw(M^*) = fw(M)$.

F4: Map M is polyhedral if and only if its graph $G(M)$ is 3-connected and $fw(M) \geq 3$. Moreover, M is polyhedral if and only if its dual is polyhedral.

F5: Every connected graph G admits a map. The *rotation scheme* described in §7.6.6 gives a systematic method for obtaining all 2-cell imbeddings of G.

F6: [Wh32] **Whitney Uniqueness Theorem:** A 3-connected, planar graph has a unique imbedding on the sphere.

F7: [Th90] A uniqueness theorem for general surfaces: if M_1 and M_2 are LEW maps with the same graph, then $|M_1| = |M_2|$. Moreover, if the graph is 3-connected, then $M_1 \approx M_2$.

REMARKS

R6: According to Fact 5 above, every connected graph has a 2-cell imbedding on a surface. Whether a graph can be imbedding on a surface such that the face boundaries are (simple) cycles is problematic (see the conjectures below).

R7: [SeTh96] gives a uniqueness result similar to Fact 6 for maps with sufficiently large face-width as a function of the genus. However, [Ar92] provides an example, for every pair of integers k, b, of two maps M_1, M_2 with the same k-connected graph such that $fw(M_1), fw(M_2) > b$ and $|M_1| \neq |M_2|$.

CONJECTURES

The Cycle Double Cover Conjecture: Every 2-connected graph contains a set $\mathcal{C}$ of cycles such that every edge is contained in exactly two cycles of $\mathcal{C}$.

The Strong Imbedding Conjecture: Every 2-connected graph can be imbedded on a surface so that each face is bounded by a cycle in the graph. The strong imbedding conjecture implies the Cycle Double Cover Conjecture.

7.6.2 The f-Vector, v- and p-Sequences, and Realizations

Elementary equalities hold among the basic parameters of a map. The two questions addressed in this section are, first, when are these necessary conditions also sufficient for the existence of a map with these parameters and, second, when can the map be imbedded in Euclidean space E^3 or E^4 such that the faces are plane convex polygons. The classic results for maps on the sphere are Eberhard's theorem of 1891 and Steinitz's theorem of 1922.

DEFINITIONS

D12: A ***map*** is ***of type*** $\{p, q\}$ if each face has p edge incidences and each vertex has q edge incidences. (No global symmetry is implied; in fact, the automorphism group of the map, as defined in §7.6.5, may be trivial.)

D13: The ***cell-distribution vector (f-vector)*** of a map M is the 3-tuple (f_0, f_1, f_2), where f_0, f_1, f_2 are the numbers of vertices, edges, and faces of M, respectively.

D14: The ***face-size sequence (p-sequence)*** of a polyhedral map M is the sequence $\{p_i\}_{i\geq 3}$ where p_i is the number of i-gonal faces in M.

D15: The ***vertex-degree sequence (v-sequence)*** of a polyhedral map M is the sequence $\{v_i\}_{i\geq 3}$ where v_i is the number of vertices of degree i in M.

D16: A polyhedral map M is ***simplicial*** (or a ***triangulation***) if the boundary of each face is a 3-cycle.

D17: A polyhedral map M is ***simple*** if its graph is 3-regular.

D18: A ***geometric realization (realization)*** of a polyhedral map M is an imbedding of M into Euclidean space E^d (no self intersection) such that each face is a plane convex polygon and that adjacent faces are not coplanar.

REMARK

R8: Using a less stringent definition of realization than above, [Mc89] defined the *realization spaces* and studied its topological properties. Also see [BuSt00] and [MoWe00] for the realization space of the torus maps.

EXAMPLES

E4: The map M in Figure 7.6.1 is of type $\{3,6\}$ with face vector $(4,12,8)$. Its dual M^* is of type $\{6,3\}$ with face vector $(8,12,4)$. The maps in Figure 7.6.2 both have v-sequence $(6,3)$, but the first has p-sequence $(0,6,0,1)$ while the second has p-sequence $(1,3,3)$.

E5: Five maps on the sphere and their corresponding 3-dimensional realizations appear in Figure 7.6.4.

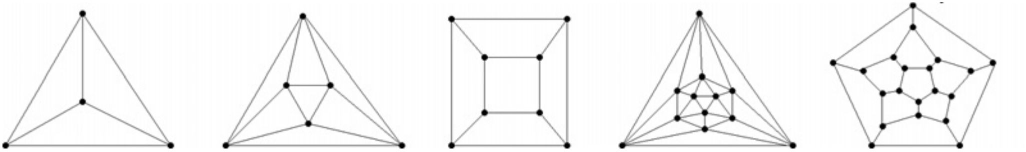


Figure 7.6.4 **The Platonic solids as realizations of maps.**

FACTS

F8: The f-vector, the p-sequence and the v-sequence satisfy the following elementary equalities:

$$\sum p_i = f_2, \qquad \sum v_i = f_0, \qquad \sum i p_i = 2f_1 = \sum i v_i$$

F9: For a map M on an orientable surface of genus g, $\alpha, \beta \geq 0$ such that $\alpha + \beta = 1$, Euler's formula implies that

$$\sum(\alpha i - 2)v_i + \sum(\beta i - 2)p_i = 4(g-1) \tag{1}$$

For example, taking $\alpha = 1/3$, $q = 0$, and M simple yields

$$\sum(6-i)p_i = 12 \tag{2}$$

F10: [Eb1891] **Eberhard's Theorem:** Condition (2) above is sufficient for the existence of a sphere map, in the following sense: if a sequence $\{p_i \,|\, i \geq 3,\ i \neq 6\}$ satisfies $\sum_{k \neq 6}(6-k)p_k = 12$, then there exist values of p_6 such that $\{p_i \,|\, i \geq 3\}$ is the p-sequence of a simple polyhedral map on the sphere. For variations on Eberhard's Theorem, see [Gr70] and [Je93]. There is no known generalization of Eberhard's theorem to arbitrary surfaces.

F11: [EdEwKu82] If S is a surface with Euler characteristic $c(S)$, if f_0, f_1, f_2, p, q are positive integers such that $f_0 - f_1 + f_2 = c(S)$, and if $pf_2 = 2f_1 = qf_0$, then there exists a map of type $\{p,q\}$ on S with f-vector (f_0, f_1, f_2), except when S is the projective plane and $\{p,q\} = \{3,3\}$, $f_0 = f_2 = 2$, $f_1 = 3$.

F12: [St22] **Steinitz's Theorem:** Every polyhedral map on the sphere is isomorphic to the boundary complex of a 3-dimensional polytope. Thus, any polyhedral map on the sphere has a realization in E^3.

F13: [Al71, Gr67] A simple polyhedral map M cannot be realized in Euclidean space of any dimension unless $|M|$ is the sphere.

F14: [BrSc95] Each simplicial polyhedral map on the torus or projective plane can be realized in E^4.

F15: [BrWi93] On any nonorientable surface N_g, there exists a simplicial map that cannot be realized in E^3. (When $g > 1$, it is an open question whether each simplicial polyhedral map of orientable genus g can be realized in E^3.)

F16: [Gr83] Equation (2) for the torus (with $\alpha = 1/3$) becomes

$$2\sum (i-3)\,v_i + \sum (i-6)\,p_i = 0$$

which leads to the following analogue of Eberhard's theorem for the torus. Given a sequence $\{p_i \mid i \geq 3,\ i \neq 6\}$ and a positive integer s, there is a realization in E^3 of some polyhedral map on the torus with p-sequence $\{p_i \mid i \geq 3\}$ and $\sum (i-3)v_i = s$ if and only if $\sum_{k \neq 6}(6-k)p_k = 2s$ and $s \geq 6$. Related results appear in [BaGrHö91].

F17: [St06] The vector (f_0, f_1, f_2) is the f-vector of a realization in E^3 of some polyhedral map on the sphere if and only if $f_0 - f_1 + f_2 = 2$, $4 \leq f_0 \leq 2f_2 - 4$, and $4 \leq f_2 \leq 2f_0 - 4$.

F18: [Gri83] The vector (f_0, f_1, f_2) is the f-vector of a realization in E^3 of some polyhedral map on the torus if and only if $f_0 - f_1 + f_2 = 0$, $f_2(11 - f_2)/2 \leq f_0 \leq 2f_2$, $f_0(11 - f_0)/2 \leq f_2 \leq 2f_0$, $2f_1 - 3f_0 \geq 6$, and $f_1 \neq 19$.

F19: [Ko36,An79,Th78] **Koebe-Andreev-Thurston circle packing theorem**: Every simplicial map M admits a circle packing representation, i.e., there exists a Riemannian metric of constant curvature $+1$, 0, or -1 on the surface and a collection of pairwise disjoint open disks on $|M|$ whose boundaries are geodesic circles such that the tangency graph of this collection of circles is $G(M)$. For a generalization to a larger class of maps, see [Mo97]. A circle packing representation of the octahedral map on the sphere appears in Figure 7.6.5.

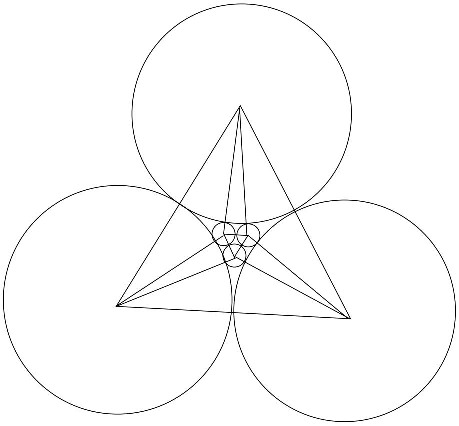


Figure 7.6.5 **A circle packing representation of the octahedral map.**

7.6.3 Map Coloring

The famous results on map coloring are the Four Color Theorem for the sphere and the Heawood Map Coloring Theorem, which is the generalization of the Four Color Theorem to surfaces of higher genus. Also in this section are a few results on coloring densely imbedded graphs.

DEFINITION

D19: The **chromatic number** $\chi(S)$ of a surface S is the least number of colors sufficient to properly color the faces of any map on S. By duality, it is also the least number of colors sufficient to properly color the vertices of any map on S. In this section, coloring will mean vertex coloring.

FACTS

F20: [ApHa76] **Four Color Theorem:** $\chi(S_0) = 4$.

F21: [Fr34] $\chi(N_2) = 6$.

F22: [RiYo68] **Heawood Map Coloring Theorem:** For every surface S except the Klein bottle N_2,

$$\chi(S) = \left\lfloor \frac{7 + \sqrt{49 - 24c}}{2} \right\rfloor$$

where c is the Euler characteristic of S. The right-hand side of the equation is called the ***Heawood formula***.

F23: A map M on the torus with $ew(M) \geq 4$ is 5-colorable. It is not known whether this same statement holds for surfaces of higher genus.

F24: [Th93] Any map M on S_g with $ew(M) \geq 2^{14g+6}$ is 5-colorable.

F25: [Th97] For a fixed surface S, there is a polynomial time algorithm to decide if a map on S can be 5-colored.

F26: Even on the sphere, the problem of deciding whether a map can be 3-colored is NP-complete.

F27: [RSST96] On the sphere, a 4-coloring can be found in $O(n^2)$ steps.

REMARKS

R9: The problem of determining the chromatic number of the sphere appeared in a 1852 letter from Augustus de Morgan to Sir William Hamilton, and was likely due to Francis Guthrie, the brother of a student of de Morgan. The computer dependent proof of Appel and Haken [ApHa76] that four colors suffice was simplified considerably [RSST97] (but still computer dependent).

R10: That the formula in the Heawood Map Coloring Theorem gives an upper bound on $\chi(S)$ was proved by Heawood [He1890]. That there exist graphs that actually require the number of colors given by that formula is a consequence of the formula for the genus of complete graphs due to Ringel and Youngs [RiYo68].

R11: Whether there is a polynomial time algorithm for deciding whether a map on an arbitrary surface can be 4-colored is unknown.

EXAMPLES

E6: Figure 7.6.9a is map on the projective plane that requires 6 colors for a proper coloring, and Figure 7.6.6 is map on the torus that requires 7. This shows that $\chi(N_1) \geq 6$ and that $\chi(S_1) \geq 7$. In fact, $\chi(N_1) = 6$ and $\chi(S_1) = 7$, in accordance with Fact 23. (The torus in Figure 7.6.6 is obtained by identifying left and right sides of the rectangle and the top and bottom sides with a 2/7 twist.)

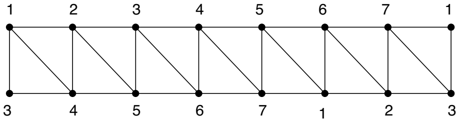


Figure 7.6.6 **A map on the torus whose graph is K_7.**

E7: An example of Fisk [Fi78] shows that no 4-color analogue of Thomassen's result (Fact 24 above) can hold. See Figure 7.6.7, where the torus is obtained by identifying opposite sides of the square.

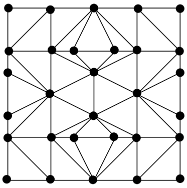


Figure 7.6.7 **A map M on the torus with exactly two odd-degree vertices is not 4-colorable.**

7.6.4 Minimal Maps

A map can be quite "degenerate", for example, the map on the sphere with 2 vertices, 1 edge, and 1 face. Polyhedral maps (and maps with edge-width or face-width bounded from below) cannot be this small. This section concerns maps that are in some sense minimal — either with respect to the number of vertices, or with respect to being polyhedral, or with respect to having edge-width k. Also covered in this section are weakly neighborly polyhedral maps.

DEFINITIONS

D20: A polyhedral map is ***neighborly*** if every pair of distinct vertices is joined by an edge.

D21: A polyhedral map is ***weakly neighborly*** (abbr. a ***wnp-map***) if every two vertices are contained on a face.

D22: The operation of ***edge contraction*** for a triangulation, and its inverse operation ***vertex splitting***, are depicted in Figure 7.6.8. After contracting an edge in a triangulation, the map may no longer be a triangulation, i.e., no longer polyhedral; this occurs if the edge is contained in a 3-cycle that is not a face boundary or if the map is the tetrahedral map.

D23: A ***minimal triangulation*** of a surface S is a triangulation such that the contraction of any edge results in a map that is no longer polyhedral.

D24: A ***k-minimal triangulation*** is a triangulation with edge-width k, such that each edge is contained in a noncontractible k-cycle. (Except on the sphere, minimal and 3-minimal are equivalent.)

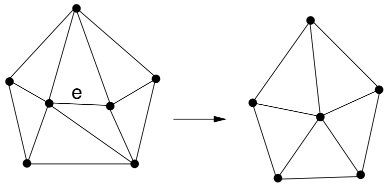


Figure 7.6.8 **Edge contraction and vertex-splitting in a triangulation.**

EXAMPLES

E8: The only wnp-maps on the sphere are the boundary complexes of the pyramids and triangular prism.

E9: There are 5 wnp-maps on S_1 and none on S_2.

E10: The wnp-maps on nonorientable surfaces up to genus 4 appear in [AlBr87].

E11: There is 1 minimal triangulation of the sphere (the tetrahedral map), 2 minimal triangulations of the projective plane (see Figure 7.6.9), 21 of the torus, and 25 of the Klein bottle.

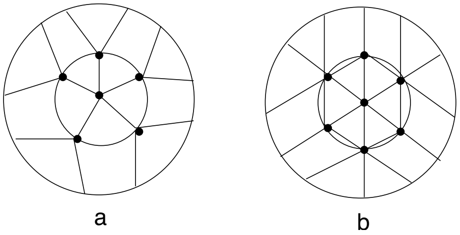


Figure 7.6.9 **The minimal triangulations of the projective plane.**

FACTS

F28: If the map M with f_0 vertices and Euler characteristic c is polyhedral, then

$$f_0 \geq \left\lceil \frac{7+\sqrt{49-24c}}{2} \right\rceil,$$

and this lower bound is attained for all surfaces except S_2, N_2, and N_3. By duality the same bound holds for f_2.

F29: The neighborly polyhedral maps attain the bound in Fact 28.

F30: [AlBr86] Each surface admits at most finitely many wnp-maps. (See Example 8.)

F31: [BaEd89] The set of minimal triangulations is finite for every fixed surface. (See Example 11.) In other words, for each surface, there is a finite set of triangulations from which any triangulation on that surface can be generated by vertex splittings.

F32: For any $k \geq 3$, the set of k-minimal graphs on a fixed surface is finite. ([MoTh01] provides a proof.)

REMARK

R12: [Br90] has provided a (non-tight) lower bound for f_1, for a polyhedral map of Euler characteristic c.

7.6.5 Automorphisms and Coverings

Every map M has a universal cover that is a classical tiling of the sphere, Euclidean plane, or hyperbolic plane (unit disc). This fact and its consequences are the subject of this section. Also addressed is the relation between a group acting as automorphisms of a map and a group acting as homeomorphisms of the surface.

In this section the classical Euclidean and hyperbolic tessellations are regarded as infinite maps, even though, by our definition, a map is a *finite* cell complex. For an expository article on connections between maps, Galois groups and Grothendieck's *dessins d'enfants*, see [JoSi96].

DEFINITIONS

D25: An ***automorphism*** of a map M is an isomorphism of M onto itself. The automorphisms form a group $\mathcal{A}ut(M)$ under composition.

D26: A ***map covering*** $f : M_1 \to M_2$ is a topological covering (see §7.4) of the respective surfaces that takes the graph of M_1 onto the graph of M_2, with ramification points possible only at vertices and face centers.

D27: The ***tessellation*** $\{p, q\}$ is the unique tesselation of the sphere or plane into regular p-gons, q incident at each vertex. This is a tiling of the sphere if $\frac{1}{p} + \frac{1}{q} > \frac{1}{2}$, of the Euclidean plane if $\frac{1}{p} + \frac{1}{q} = \frac{1}{2}$, or of the hyperbolic plane (unit disc) if $\frac{1}{p} + \frac{1}{q} < \frac{1}{2}$.

D28: The ***triangle group*** $(p, q, 2)$ is the symmetry group of the tessellation $\{p, q\}$.

D29: The ***Coxeter group*** $W(p, q)$ is the group with presentation by three generators ρ_0, ρ_1, ρ_2 and the relations

$$\rho_0^2 = \rho_1^2 = \rho_2^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_2\rho_0)^2 = 1 \qquad (3)$$

EXAMPLES

E12: Both torus maps M and M^* in Figure 7.6.1 are coverings of the tetrahedral map in Figure 7.6.4. The covering by M is ramified at vertices and the covering by M^* is ramified at face centers. Both are 2-fold coverings, that is, each unramified point of the sphere is covered by two points of the torus.

E13: Figure 7.6.10 shows all the (hyperbolic) mirors of reflection symmetries of the tessellation $\{6,4\}$ (or $\{4,6\}$). These lines form a subdivision (called the ***Coxeter complex***) of the hyperbolic plane into triangles (called ***flags***).

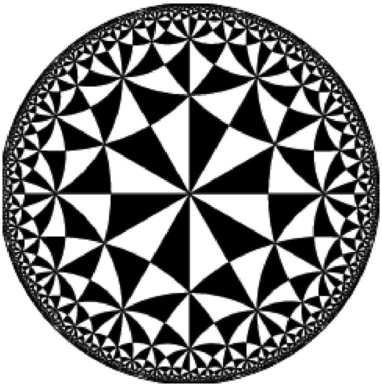


Figure 7.6.10 **Reflection symmetries of the hyperbolic tessellation $\{6,4\}$.**

FACTS

F33: The symmetry group $(p,q,2)$ of the tessellation $\{p,q\}$ is isomorphic to the Coxeter group $W(p,q)$. The Coxeter generators ρ_0, ρ_1, ρ_2 (and their conjugates) correspond to reflections in the three sides of a flag, as described in example E12; the products $\rho_1\rho_2, \rho_0\rho_2, \rho_0\rho_1$ (and their conjugates) correspond to rotations about vertices, midpoints of edges, and face centers, respectively.

F34: Every map M has a covering by a tessellation $\{p,q\}$ for some p,q. In other words, every map M is the quotient of a tesselation $\{p,q\}$ by a subgroup H_M of the Coxeter group $W(p,q)$.

F35: [Vi83a] The automorphism group $\mathcal{A}ut(M)$ of any map M is isomorphic to the quotient $N_W(H_M)/H_M$, where N_W denotes the normalizer and where H_M is the subgroup of Fact 34.

F36: Every map M of type $\{p,q\}$ has an unramified covering by the tessellation $\{p,q\}$. For example, the map on N_5 of type $\{5,5\}$ in Figure 7.6.11, is covered by the tessellation $\{5,5\}$ of the hyperbolic plane. (The map is obtained by identifying like labeled edges in the figure.)

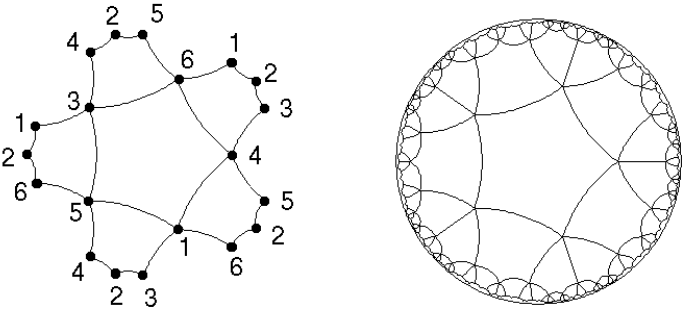


Figure 7.6.11 **The regular self-dual map $\{5,5\}_3$ and its universal cover $\{5,5\}$.**

F37: [Bi72] The automorphism group of an orientable map of genus $g > 1$ can be faithfully represented in the group of $2g \times 2g$ symplectic matrices with integral entries. From this it can be proved, for example, that if prime p divides $|\mathcal{A}ut(M)|$, then the genus of the map M is either $1, 2$ or at least $\frac{1}{2}(p-1)$.

F38: [Hu1892] ***Hurwitz formula:*** If a group Γ acts on a surface of Euler characteristic $c < 0$, then $|\Gamma| \leq -84c$.

F39: A map M with Euler characteristic $c < 0$ satisfies $|\mathcal{A}ut(M)| \leq -84c$ with equality if and only if M is a regular map of type $\{3, 7\}$ or $\{7, 3\}$ (see §7.6.7 for the definition of regular). This is a direct consequence of the Hurwitz formula.

F40: [Tu83] If a group Γ acts on an orientable surface S, then some Cayley graph G of Γ imbeds in S, and the natural action of Γ on G (by left multiplication) extends to an action of Γ on S.

REMARK

R13: [JoSi78] Fact 35 implies that the surface of any map M can be assumed to be a Riemann surface such that $\mathcal{A}ut(M)$ acts as a group of conformal homeomorphisms. The edges of $G(M)$ are geodesics of equal length with respect to a Riemannian metric of constant curvature (defined everywhere except perhaps at finitely many ramification points located at vertices and face centers) and the angles formed by successive edges incident with a vertex are equal.

7.6.6 Combinatorial Schemes

The definition of map in §7.6.1 as a cell complex is topological. A strictly combinatorial description, although less intuitive, is often easier to apply. Three such schemes are described: rotation scheme, permutation scheme, and graph encoded map.

DEFINITIONS

D30: A ***rotation scheme*** (G, ρ) consists of a graph G and a set $\rho = \{\rho_v\}_{v \in V(G)}$, where ρ_v is a cyclic permutation of the edges incident to v. This scheme [Ed60] encodes any map with graph G imbedded on an orientable surface (and can be extended to include nonorientable imbeddings).

D31: The ***map of a rotation scheme*** is obtained as follows. Given a directed edge $e_1 = (v_0, v_1)$ of G, consider the cycle consisting of successive directed edges $e_1 e_2 \ldots e_m = e_1$, where $e_i = (v_{i-1}, v_i)$ and

$$e_{i+1} = \rho_{v_i}(e_i)$$

Each (undirected) edge lies on exactly two such cycles. Regarding each cycle as the boundary of a polygonal 2-cell and gluing together 2-cells along paired edges results in an orientable surface in which G is imbedded. Conversely, the ***rotation scheme of a map*** M on an orientable surface is (G, ρ), where G is the graph of M and ρ_v is the cyclic permutation of the edge incidence on vertex v induced by the orientation of the surface, say clockwise.

D32: A ***permutation scheme*** (π, σ) ***on a finite set*** X consists of permutations π and σ acting on X, such that each orbit of π has length 2 and such that the permutation group $H\langle \pi, \sigma \rangle$ generated by σ and π is transitive on X.

D33: The ***vertices, edges*** and ***faces*** of the permutation scheme (π, σ) are the cycles of σ, π and $\sigma \circ \pi$, respectively.

D34: Two faces (of any dimension) of the permutation scheme (π, σ) on a set X are ***incident*** if the corresponding cycles have an element of X in common.

D35: The ***permutation scheme of a map*** M has as the elements of its object set X the "half edges" of M (see example E14). Each cycle of σ is the cyclic (say clockwise) order of the half edges incident to a given vertex on the surface $|M|$, and each cycle of π is the two "half edges" at a midpoint of an edge. In a permutation scheme for a map, the graph is not explicitly part of the data.

D36: A ***graph encoded map*** (abbr. ***GEM***) is a connected, finite graph $\mathcal{G}$, regular of degree 3, together with a proper 3-coloring of the edges in the color set $I = \{0, 1, 2\}$, and with subgraphs $\mathcal{G}_i$, each induced by all edges not colored i, such that the connected components of $\mathcal{G}_1$ are 4-cycles.

D37: The ***vertices, edges*** and ***faces of a GEM*** $\mathcal{G}$ are the connected components of $\mathcal{G}_0, \mathcal{G}_1$ and $\mathcal{G}_2$, respectively.

D38: Two faces (of any dimension) of a GEM are ***incident faces*** if the corresponding subgraphs have non-empty intersection.

D39: The ***graph encoding*** $\mathcal{G}$ ***of a given map*** M, is obtained from the barycentric subdivision Δ of M, by giving each vertex v of Δ the label $0, 1,$ or 2, according to the dimension of the face in M that vertex v represents; then $\mathcal{G}$ is the dual graph of Δ, with color i assigned to edge e if and only if the two endpoints of the edge of Δ that e crosses are not labeled i.

REMARKS

R14: Permutation schemes can represent any map on an orientable surface (and can be extended to include maps on nonorientable surfaces). They have been used by [Ja68], [Co75], [Tu79], [JoSi78], [Wa75], and [St80].

R15: GEM's were introduced (in arbitrary dimension) as "combinatorial maps" by [Vi83] and as "crystallizations" by [Fe76] and [Ga79] in a topological context. The terminology "graph encoded map" is due to [Li82]. A graph encoding can represent any orientable or nonorientable map.

R16: Lifting the restriction that each orbit of π in a permutation scheme must have length 2 or that each component of $\mathcal{G}_1$ in a graph encoded map must be a 4-cycle, results in the concept of ***hypermap*** or ***hypergraph imbedding***.

EXAMPLE

E14: On the left in Figure 7.6.12 below is a map on the sphere, and on the right is the corresponding graph encoded map. A rotation scheme for this map is given by the three cyclic permutations of edges incident to each of the three vertices:

$$\rho_1 = (1\ 2\ 3), \qquad \rho_2 = (1\ 4\ 5), \qquad \rho_3 = (5\ 4\ 3\ 2)$$

A permutation scheme for this same map is given by two permutations:

$\pi = (1^+\ 1^-)(2^+\ 2^-)(3^+\ 3^-)(4^+\ 4^-)(5^+\ 5^-) \quad \sigma = (1^+\ 2^+\ 3^+)(1^-\ 4^+\ 5^+)(5^-\ 4^-\ 3^-\ 2^-)$

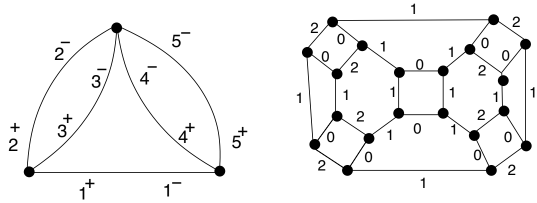


Figure 7.6.12 **A map on the sphere and its graph encoding.**

FACTS

F41: The notion of an automorphism of a map M can be translated into terms of each of the above schemes:

> If a map M is given as a rotation scheme (G, ρ), then an automorphism of M corresponds to an isomorphism $f : G \to G$ that preserves the cyclic permutations, i.e., $f \circ \rho_v = \rho_{f(v)} \circ f$ for all $v \in V(G)$.
>
> If M is given as a permutation scheme (π, σ) on a set X, then an automorphism of M corresponds to a bijection $f : X \to X$ such that $f \circ \phi = \phi \circ f$ for all $\phi \in H\langle \pi, \sigma \rangle$.
>
> If M is given as a GEM $\mathcal{G}$, an automorphism of M corresponds to a color preserving graph isomorphisms of $\mathcal{G}$.

F42: The notion of duality can also be easily translated:

> If M is given as a permutation scheme (π, σ) acting on the set X, the dual map M^* is given by the permutation scheme $(\pi, \sigma \circ \pi)$ acting on the same set X.
>
> If M is given as a GEM $\mathcal{G}$, the dual map M^* is encoded by the same graph $\mathcal{G}$ with edge colors 0 and 2 interchanged.

F43: The surface of a map is orientable if and only if the corresponding GEM is bipartite.

7.6.7 Symmetry of Maps

Regular maps, those enjoying the greatest symmetry, are the surface analogues of the Platonic solids. Also discussed are symmetrical and vertex-transitive maps.

DEFINITIONS

D40: A ***flag*** of a map M is an ordered triple (F_0, F_1, F_2) of mutually incident faces of dimensions $0, 1$ and 2, respectively.

D41: A map M is a ***regular map*** if $\mathcal{A}ut(M)$ acts transitively on the set of flags.

D42: A map M is a ***symmetrical map*** if $\mathcal{A}ut(M)$ has at most two orbits in its action on the set of flags.

D43: A map is a ***chiral map*** if it is symmetrical, but not regular.

D44: A ***Cayley map for a group*** Γ with generator set Δ, is an imbedding of the Cayley graph $G_\Delta\Gamma$, using a rotation scheme as defined in §7.6.6. The cyclic permutation on the edges $\Delta^* = \Delta \cup \Delta^{-1}$ incident at each vertex must be the same at each vertex (see Example 21).

REMARKS

R17: For a symmetrical map M, the automorphism group $\mathcal{A}ut(M)$ acts transitively on the set of vertices, on the set of edges, and on the set of faces.

R18: If a map M is given in terms of a GEM $\mathcal{G}$, then the flags of M are in bijective correspondence with the vertices of $\mathcal{G}$. Therefore the map M is regular if and only if the (graph) automorphism group of $\mathcal{G}$ is vertex-transitive.

EXAMPLES

E15: The regular maps on the sphere are the boundary complexes of the five ***Platonic solids*** (see Figure 7.6.4) which have types $\{3,3\}$, $\{3,4\}$, $\{4,3\}$, $\{3,5\}$, $\{5,3\}$, respectively, plus the infinite families of (non-polyhedral) maps of types $\{p,2\}, \{2,p\}$, $p > 0$.

E16: Since every map on the projective plane has a 2-fold covering by a map on the sphere (Fact 51), it follows from Example 15 that there are four regular maps on the projective plane of types $\{3,4\}, \{4,3\}, \{3,5\}, \{5,3\}$ and infinite families of types $\{p,2\}$ and $\{2,p\}$, where $p \equiv 2 \bmod 4$.

E17: There are three infinite families of regular torus maps of types $\{3,6\}, \{6,3\}$ and $\{4,4\}$. For example, in the notation of Fact 60, the maps in Figure 7.6.1, are $\{3,6\}_4$ and $\{6,3\}_4$.

E18: [CoDo01] used the bijection in Fact 57 and a network of computers to determine all regular maps on orientable surfaces of genus 2 to 15 and all regular maps on nonorientable surfaces from genus 4 to 30.

E19: The ***Kepler-Poinsot regular star polyhedra***, shown in Figure 7.6.13, are self-intersecting realizations of regular maps. In the notation of Fact 60 below, these maps are $\{5,5\,|\,3\}$ (12 pentagons on a surface of genus 4 — great dodecahedron and small stellated dodecahedron), $\{5,3\}_{10}$ (12 pentagons on the torus — great stellated dodecahedron) and $\{3,5\}_{10}$ (20 triangles on the torus — great icosahedron).

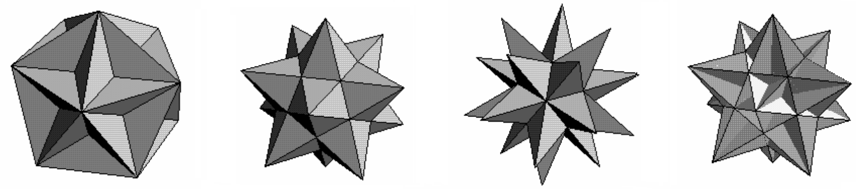


Figure 7.6.13 Star polyhedra.

E20: [ScWi85, ScWi86] From the history of automorphic functions come two regular maps of genus 3, the 1879 ***Klein map*** $\{7,3\}_8$ composed of 24 heptagons with automorphism group $PGL(2,7)$, and the 1880 ***Dyck map*** $\{8,3\}_6$ composed of 12 octagons (shown in dual form in Figure 7.6.14). The ***Coxeter regular skew polyhedra*** in E^4 also provide examples of regular maps; they are $\{4,6\,|\,3\}$, $\{6,4\,|\,3\}$, $\{4,8\,|\,3\}$, and $\{8,4\,|\,3\}$. The Klein, Dyck, and Coxeter maps all have realizations in E^3.

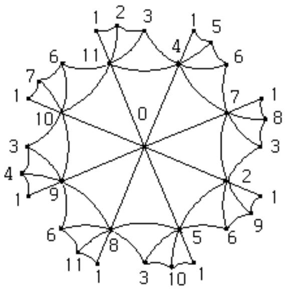


Figure 7.6.14 **Dyck's map $\{3,7\}_8$.**

E21: Figure 7.6.15 is a chiral map on the torus. Opposite sides of the square are to be identified. This map is presented as the Cayley map of the cyclic group $Z_5 = \{0,1,2,3,4\}$ with generating set $\Delta = \{1,2\}$ and cycle $\pi = (-1\ 2\ 1\ \ -2)$ on Δ^*.

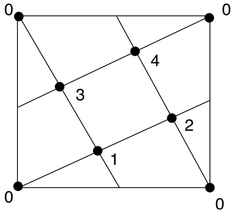


Figure 7.6.15 **A chiral map on the torus given as a Cayley map of Z_5.**

Denoting an edge by a pair of vertices and a face by its four vertices, the flags $(1,12,1234)$ and $(2,12,1234)$ are in two different orbits under the action of the automorphism group acting on the set of flags. There is no automorphism that leaves edge 12 and face 1234 fixed and takes vertex 1 to vertex 2.

E22: Coxeter and others noticed that regular maps frequently occur as coverings of smaller regular maps on other surfaces. For example, the regular torus maps $\{3,6\}_4$ and $\{6,3\}_4$ in Figure 7.6.1 are 2-fold coverings of the tetrahedral map $\{3,3\}$ on the sphere. Constructions of families of regular maps using coverings appear in [JoSu00], [Si00], [Vi84], and [Wi78] among others.

E23: There are various group theoretical constructions of regular maps, for example [Mc91], [McMoWe93]. McMullen constructed a family of maps related to the Klein map $\{7,3\}_8$. For each odd prime p there is a regular map of type $\{p,3\}$ with $\frac{1}{2}(p^2-1)$ faces and orientation preserving automorphism group PSL $(2,p)$.

E24: The vertex-transitive maps on the sphere, classified by [FlIm79], consist of the regular spherical maps and the boundary complexes of the Archimedean solids (semiregular polyhedra), of the prisms and antiprisms. [Ba91] gave a classification of the vertex-transitive maps on the Klein bottle.

FACTS

F44: A map M with f_1 edges has exactly $4f_1$ flags.

F45: In $\mathcal{A}ut(M)$, the stabilizer of any flag is trivial.

F46: For any map M with f_1 edges, the two immediately preceding facts imply that $|\mathcal{A}ut(M)| \leq 4f_1$, with equality if and only if M is regular. In this sense, the regular maps have the largest possible automorphism group.

F47: On each orientable surface there is a regular map.

F48: For a regular map on an orientable surface, half the automoprhisms act as orientation preserving homeomorphisms of the surface and half as orientation reversing.

F49: Not every nonorientable surface has a regular map; for example, there are no regular maps on the surfaces with nonorientable genus 2 and 3.

F50: [Vi83b], [Wi78a] Every nonorientable regular map has a unique 2-fold unramified covering by a regular orientable map.

F51: No chiral map exists on a nonorientable surface.

F52: For any surface S with Euler characteristic $c(S) < 0$, there are at most finitely many regular maps. This follows from the Hurwitz formula in §7.6.5.

F53: [Vi83b] For any pair (p, q) such that $\frac{1}{p} + \frac{1}{q} \leq \frac{1}{2}$, there are infinitely many regular maps of type $\{p, q\}$. [NeŠk01] subsequently showed that these maps may be chosen to have arbitrarily large face-width.

F54: [Wi89] There is a regular map with complete graph K_n if and only if $n = 2, 3, 4, 6$.

F55: [Bi71] There is a symmetrical map with complete graph K_n if and only if n is a prime power and, for each prime power, the symmetrical map is unique.

F56: [Vi83a, 83b] The regular maps M of type $\{p, q\}$ are in bijection with the conjugacy classes of normal subgroups N of finite index in the Coxeter group $W(p, q)$.

F57: A regular map M is the quotient of the tessellation $\{p, q\}$ by the corresponding normal subgroup N of symmetries of $\{p, q\}$; moreover $\mathcal{A}ut(M) \approx W(p, q)/N$.

F58: According to Fact 57, $\mathcal{A}ut(M)$, for a regular map M, has a presentation with three generators, the same relations as given for the Coxeter group $W(p, q)$ in Equation (3) together with some additional relations (except no additional relations in the case of a regular spherical map).

F59: Two special cases have received particular attention, the regular maps $\{p, q\}_r$ where the single relation $(\rho_0\rho_1\rho_2)^r$ has been added and the regular maps $\{p, q \,|\, m\}$ where the single relation $(\rho_0\rho_1\rho_2\rho_1)^m$ has been added. Coxeter and Moser [CoMo57] have provided partial tables of parameters p, q, r and p, q, m for which a *finite* regular map with those parameters exists. Figure 7.6.11 shows the regular map $\{5, 5\}_3$.

F60: Any Cayley map of a group Γ is vertex transitive, Γ acting as a group of automorphism of the Cayley map by left multiplication.

F61: The double torus S_2 has the interesting property that only finitely many groups act (as a group of homeomorphisms) on S_2, but there are infinitely many vertex-transitive (Cayley graphs) with genus 2.

F62: [Th91],[Ba91] For each $g \geq 3$, there are only finitely many vertex-transitive graphs of orientable genus g while there are infinitely many of genus $0, 1$ and 2.

7.6.8 Enumeration

W. T. Tutte [Tu68] pioneered map enumeration in the 1960's. Explicit results for maps on the sphere appear below. Results on generating functions and asymptotics for the number of such maps on general surfaces can be found in the texts [GoJa83], [Ya99] and the references therein. A connection between map enumeration, matrix integrals and 2-dimensional quantum gravity is explained in [Zv97].

DEFINITIONS

D45: A **rooted map** is a map in which a flag has been distinguished.

D46: A rooted map is a ***near triangulation*** if every nonroot face is a 3-gon.

EXAMPLES

E25: For the sphere, the 2-connected rooted maps with 4 edges are shown in the first row of Figure 7.6.16. The first four of these comprise all 2-connected rooted maps with 3 vertices and 3 faces. The root face is the outer face, the root vertex and edge are in boldface.

E26: On the second row of Figure 7.6.16 are the rooted near triangulations with 4 inner faces and a root face with 2 edges. The root face is the outer face; the root edge is the bottom edge; and the root vertex is in boldface.

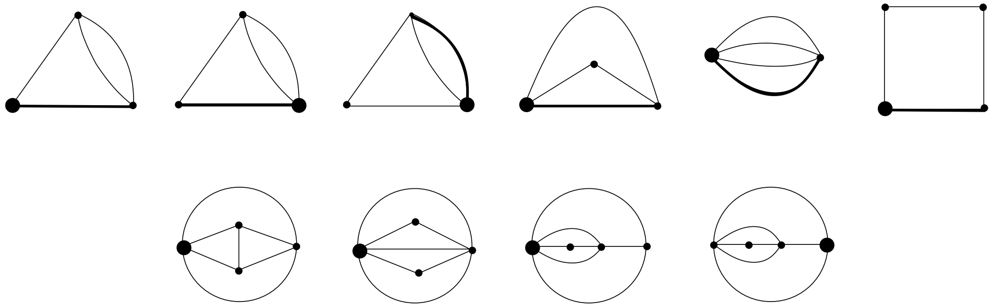


Figure 7.6.16 **Counting maps on the sphere.**

FACTS

F63: [Tut63] The number of rooted maps on the sphere with $n \geq 0$ edges is

$$g(n) = \frac{2 \cdot 3^n (2n)!}{n!(n+2)!}$$

F64: [Tut63] The number of 2-connected rooted maps on the sphere with $n \geq 1$ edges is

$$\frac{2(3n-3)!}{n!(2n-1)!}$$

F65: [N. Wormald] (see [GoJa83]) The number of 2-edge-connected rooted maps on the sphere with $n \geq 0$ edges is

$$\frac{2(4n+1)!}{(n+1)!(3n+2)!}$$

F66: [BrTu64] The number of 2-connected rooted maps on the sphere with $n \geq 1$ vertices and $k \geq 2$ faces is

$$\frac{(2n+k-5)!(2k+n-5)!}{(n-1)!(k-1)!(2n-3)!(2k-3)!}$$

F67: [Br63] The number of rooted near triangulations of the sphere with $n+2j$ inner faces and $n \geq 2$ edges on the root face is

$$\frac{2^{j+2}(2n+3j-1)!(2n-3)!}{(j+1)!(2n+2j)!((n-2)!)^2}, \; j \geq -1$$

7.6.9 Paths and Cycles in Maps

This section covers three topics involving paths and cycles: the Lipton-Tarjan separator theorem, the existence of nonrevisiting paths in polyhedral maps, and the decomposition of maps along cycles in the graph. The third topic is related to a result of Robertson and Seymour on minors.

DEFINITIONS

D47: A path p in the graph of a map M is said to be ***nonrevisiting*** if $p \cap F$ is connected for each face F of M.

D48: A surface S has the ***nonrevisiting path property*** if, for any polyhedral map M on S, any two vertices of M are joined by a nonrevisiting path.

D49: A map M is a ***map minor*** of a map M' if M can be obtained from M' by a sequence of edge contractions and deletions. The operations of edge deletion and edge contraction on a graph can be extended to a surface imbedding of the graph in an obvious way.

EXAMPLE

E27: A polyhedral map on the surface S_2 that fails to have the nonrevisiting path property appears in Figure 7.6.17 below. There is no nonrerevisiting path from x to y. (The map is obtained by gluing along like labeled edges.)

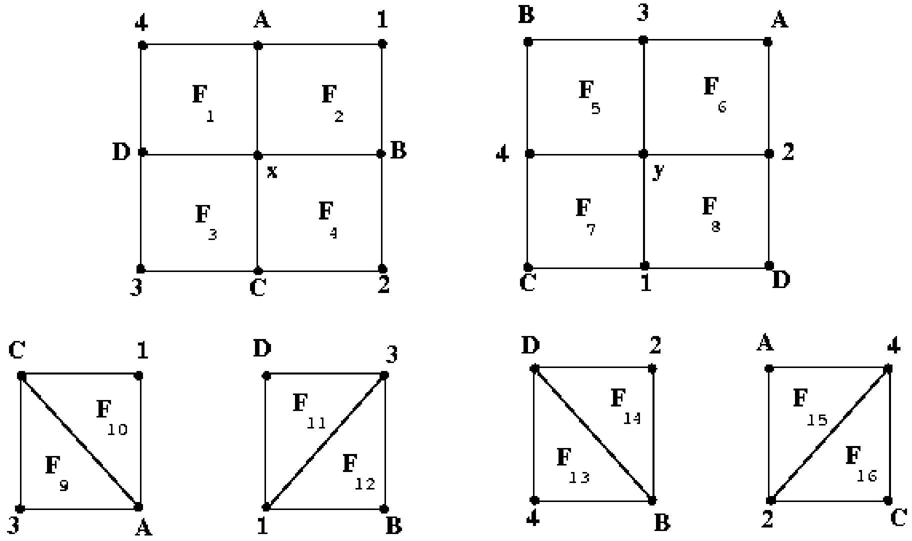


Figure 7.6.17 **A map on S_2 that does not satisfy the non-revisiting path property.**

FACTS

F68: [LiTa79] **Planar Separator Theorem**: A planar graph with n vertices has a set of at most $2\sqrt{2n}$ vertices whose removal leaves no component with more than $2n/3$ vertices.

F69: [AlSeTh94] Let M be a loopless map on the sphere with n vertices. Then there is a simple closed curve τ on the surface of the sphere passing through at most $k \leq 3\sqrt{2n}/2$ vertices (and no other points of the graph) such that each of the two open disks bounded by τ contain at most $2n/3-k/2$ vertices. This result slightly improves the Lipton-Tarjan separator theorem.

F70: [GiHuTa84] A map of genus g contains a set of at most $O(\sqrt{gn})$ vertices whose removal leaves no component of the graph with more than $2n/3$ vertices. This generalizes the Lipton-Tarjan theorem to maps on orientable surfaces of higher genus.

F71: [PuVi98] For polyhedral maps, the nonrevisiting path property holds for the sphere, torus, projective plane and Klein bottle. It fails for all other surfaces except possibly the nonorientable surface of genus 3 (see [PuVi96] and example E27).

F72: The nonrevisiting path property holds for every polyhedral map with face-width at least 4.

F73: [RoSe88] Let M_0 be a map on a surface S other than the sphere. There exists a constant k such that, for any map M on S with $fw(M) \geq k$, M_0 is a map minor of M. The following two results provide values for the constant k when the given M_0 consists of certain sets of disjoint cycles.

F74: [Sc93] A map M on the torus with face-width w contains $\lfloor 3w/4 \rfloor$ disjoint noncontractible cycles.

F75: [BrMoRi96] For general surfaces there exist $\lfloor w/2 \rfloor$ pairwise disjoint contractible cycles in the graph of any map M, all containing a particular face, $\lfloor (w-1)/2 \rfloor$ pairwise disjoint, pairwise homotopic, surface nonseparating cycles, and $\lfloor (w-1)/8 \rfloor - 1$ pairwise disjoint, pairwise homotopic, surface separating, noncontractible cycles. (It is unknown whether any map of orientable genus $g \geq 2$ with face-width at least 3 must contain a noncontractible surface separating cycle.)

F76: [Bar88] Every polyhedral map on the torus (projective plane, Klein bottle) is isomorphic to the complex obtained by identifying the boundaries of two faces of a 3-polytope (cross identifying one face of a 3-polytope, cross identifying two faces of a 3-polytope).

F77: [Yu97] (see also [Th93]) If d is a positive integer and M is a map on S_g of face-width at least $8(d+1)(2^g-1)$, then the graph of M contains a collection of induced cycles $C_1, C_2, \ldots, C_g$ such that the distance between distinct cycles is at least d and cutting along the cycles results in a map on the sphere. This generalizes Fact 76.

F78: [Sc91] Schrijver proved necessary and sufficient conditions (conjecteure by Lovasz and Seymour) for the existence of pairwise disjoint cycles $\tilde{C}_1, \ldots, \tilde{C}_k$ on the graph of a map M homotopic to given closed curves $C_1, \ldots, C_k$ on the surface.

REMARK

R19: The Lipton-Tarjan separator theorem has applications to divide-and-conquer algorithms. Nonrevisiting paths arise in complexity issues for edge following linear programming algorithms like the simplex method.

References

[AlSeTh94] N. Alon, P. Seymour, and R. Thomas, Planar separators, *SIAM J. Discrete Math.* 7 (1994), 184–193.

[Al71] A. Altshuler, Polyhedral realization in R^3 of triangulaltions of the torus and 2-manifolds in cyclic 4-polytopes, *Discrete Math.* 1 (1971), 211–238.

[AlBr86] A. Altshuler and U. Brehm, On weakly neighborly polyhedral maps of arbitrary genus, *Israel J. Math.* 53 (1986), 137–157.

[AlBr87] A. Altshuler and U. Brehm, The weakly neighborly polyhedral maps on the nonorientable 2-manifold with Euler characteristic -2, *J. Combin. Theory Ser.* A 45 (1987), 104–124.

[An70] E. M. Andreev, On convex polyhedra in Lobacevskii spaces, *Mat. Sb.* (N.S.) 81 (1970), 445–478.

[ApHa76] K. Appel and W. Haken, Every planar map is four colorable, *Bull. Amer. Math. Soc.* 82 (1976), 711–712.

[Ar92] D. Archdeacon, Densely embedded graphs, *J. Combin. Theory, Ser. B* 54 (1992), 13–36.

[Bab91] L. Babai, Vertex-transitive graphs and vertex-transitive maps, *J. Graph Theory* 15 (1991), 587–627.

[Bar88] D. W. Barnette, Decomposition theorems for the torus, projective plane and Klein bottle, *Discrete Math.* 70 (1988), 1–16.

[BaEd89] D. Barnette and A. L. Edelson, All 2-manifolds have finitely many minimal triangulations, *Israel J. Math.* 67 (1989), 123–128.

[BaGrHö91] D. Barnette, P. Gritzmann, and R. Höhne, On valences of polyhedra, *J. Combin. Theory Ser. A* 58 (1991), 279–300.

[Bi71] N. Biggs, Automorphisms of embedded graphs, *J. Combin. Theory* 11 (1971), 132–138.

[Bi72] N. Biggs, The symplectic representation of map automorphisms, *Bull. London Math. Soc.* 4 (1972), 303–306.

[BoLi95] C. P. Bonnington and C. H. C. Little, *The Foundations of Topological Graph Theory*, Springer-Verlag, New York, 1995.

[Br90] U. Brehm, Polyhedral maps with few edges, Topics in combinatorics and graph theory (Oberwolfach, 1990), 153–162, *Physica*, Heidelberg, 1990.

[BrSc95] U. Brehm and G. Schild, Realizability of the torus and the projective plane in R^4, *Israel J. Math.* 91 (1995), 249–251.

[BrSh97] U. Brehm and E. Schulte, Polyhedral maps, pp 345–358 in *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL, 1997.

[BrWi93] U. Brehm and J. M. Wills, Polyhedral manifolds, *Handbook of Convex Geometry*, Vol. A, B, 535–554, North-Holland, Amsterdam, 1993.

[Br63] W. G. Brown, Enumeration of non-separable planar maps, *Canad. J. Math.* 15 (1963), 526–545.

[BrTu64] W. G. Brown and W. T. Tutte, On the enumeration of rooted non-separable planar maps, *Canad. J. Math.* 16 (1964), 572–577.

[BrMoRi96] R. Brunet, B. Mohar and R. B. Richter, Separating and nonseparating disjoint homotopic cycles in graph embeddings, *J. Combin. Theory Ser. B* 66 (1996), 201–231.

[BuSt00] H. Burgiel and D. Stanton, Realizations of regular abstract polyhedra of types $\{3,6\}$ and $\{6,3\}$, *Discrete Comput. Geom.* 24 (2000), 241–255.

[CoDo01] M. Conder and P. Dobcsányi, Determination of all regular maps of small genus, *J. Combin. Theory, Ser. B* 81 (2001), 224–242.

[Co75] R. Cori, Un code ppour les graphes planaires et ses applications, *Asterisk* 27 (1975).

[CoMo57] H. S. M. Coxeter and W. O. J. Moser, *Generators and Relations for Discrete Groups*, Springer-Verlag, Berlin, 1957.

[Eb1891] V. Eberhard, *Sur Morphologie der Polyeder*, Teubner, Leipzig, 1891.

[EdEwKu82] A. L. Edmonds, J. H. Ewing, and R. S. Kulkarni, Regular tessellations of surfaces and $(p, q, 2)$-triangle groups, *Ann. of Math.* 116 (1982), 113–132.

[Ed60] J. R. Edmonds, A combinatorial representation for polyhedral surfaces, *Notices Amer. Math. Soc.* 7 (1960), 646.

[Fe76] M. Ferri, Una rappresentazione delle n-varieta topologiche triangolabili mediante grafi $(n+1)$-colorati, *Boll. Un. Math. Ital.* 13B (1976), 250–260.

[Fi78] S. Fisk, The nonexistence of colorings. *J. Combinatorial Theory, Ser. B* 24 (1978), 247–248.

[FlIm79] H. Fleischner and W. Imrich, Transitive planar graphs, *Math. Slovaca* 29 (1979), 97–106.

[Fr34] P. Franklin, A six color problem, *J. Mat. Phys.* 16 (1934), 363–369.

[Ga79] C. Gagliardi, A combinatorial characterization of 3-manifold crystallizations, *Boll. Un. Mat. Ital.* 16A (1979) 441–449.

[GiHuTa84] J. R. Gilbert, J. P. Hutchinson, and R. E. Tarjan, A separator theorem for graphs of bounded genus, *J. of Alg.* 5 (1984), 391–407.

[GoJa83] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, John Wiley & Sons, New York, 1983.

[Gri83] P. Gritzmann, The toroidal analogue of Eberhard's theorem, *Mathematika*, 30 (1983), 274–290.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, John Wiley & Sons, New York, 1987.

[Grü70] B. Grünbaum, Polytopes, graphs, and complexes, *Bull. Amer. Math. Soc. 76 (1970)*, 1131–1201.

[Grü67] B. Grünbaum, *Convex Polytopes*, Interscience, 1967.

[He1890] P. J. Heawood, Map-colour theorem, *Quart. J. Pure Appl. Math.* 24 (1890), 332–338.

[Hu1892] A. Hurwitz, Uber algebraische Gebilde mit eindeutigen Transformationen in sich, *Math. Ann.* 41 (1892), 403–442.

[Ja68] A. Jacques, Sur le genre d'une paire de substitutions, *C. R. Acad. Sci. Paris* 367 (1968) 625–627.

[Je93] S. Jendrol, On face vectors and vertex vectors, *Discrete Math.* 118 (1993), 119–144.

[JoSu00] G. A. Jones and D. B. Surowski, Regular cyclic coverings of the Platonic maps, *European J. Combin.* 21 (2000), 333–345.

[JoSi78] G. A. Jones and D. Singerman, Theory of maps on orientable surfaces, *Proc. Lond. Math. Soc.* 37 (1978), 273–301.

[JoSi96] G. A. Jones and D. Singerman, Belyǐ functions, hypermaps and Galois groups, *Bull. London Math. Soc.* 28 (1996), 561–590.

[Ko36] P. Koebe, Kontaktprobleme der konformen Abbildung, *Ber. Verh. Sächs. Akad. Wiss. Leipzig*, Math.-Phys. Kl. 88 (1936), 141–164.

[Li82] S. Lins, Graph-encoded maps, *J. Combin Theory Ser. B*, 32 (1982), 171–181.

[LiTa79] R. J. Lipton,and R. E. Tarjan, A separator theorem for planar graphs, *SIAM J. Appl. Math.* 36 (1979), 177–189.

[Mc89] P. McMullen, Realizations of regular polytopes, *Aequationes Math.* 37 (1989), 38–56.

[Mc91] P. McMullen, Regular polyhedra related to projective linear groups, *Discrete Math.* 91 (1991), 161–170.

[McMoWe93] P. McMullen, B. Monson, and A. I. Weiss, Regular maps constructed from linear groups, *European J. Combin.* 14 (1993), 541–552.

[Mo97] B. Mohar, Circle packings of maps in polynomial time, *European J. Combin.* 18 (1997), 785–805.

[MoTh01] B. Mohar and C. Thomassen, *Graphs on Surfaces*, The Johns Hopkins University Press, 2001.

[MoWe00] B. Monson and A. I. Weiss, Realizations of regular toroidal maps of type $\{4,4\}$, *Discrete Comput. Geom.* 24 (2000), 453–465.

[NeŠk01] R. Nedela and M. Škoviera, Regular maps on surfaces with large planar width, *Europ. J. Combinatorics* 22 (2001), 243–261.

[PuVi96] H. Pulapaka and A. Vince, Nonrevisiting paths on surfaces, *Discrete Comput. Geom.* 15 (1996), 352–257.

[PuVi98] H. Pulapaka and A. Vince, Nonrevisiting paths on surfaces with low genus, *Discrete Math.* 182 (1998) 267–277.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Natl. Acad. Sci. U.S.A.* 60 (1968), 438–445.

[RoSe88] N. Robertson and P. Seymour, Graph minors. VII. Disjoint paths on a surface, *J. Combin. Theory, Ser. B* 45 (1988), 212–254.

[RSST96] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, Efficient four-coloring planar graphs, *Proc. ACM Symp. Theory Comput. 28* (1996), 571–575.

[RSST97] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, The four-colour theorem, *J. Combin. Theory, Ser. B* 70 (1997), 2–44.

[Sc91] A. Schrijver, Disjoint circuits of prescribed homotopies in a graph on a compact surface, *J. Combin. Theory, Ser. B* 51 (1991), 127–159.

[Sc93] A. Schrijver, Graphs on the torus and geometry of numbers, *J. Combin. Theory, Ser. B* 58 (1993) 147–158.

[ScWi85] E. Schulte and J. M. Wills, A polyhedral realization of Felix Klein's map $\{3,7\}_8$ on a Riemann surface of genus 3, *J. London Math. Soc.* 32 (1985), 539–547.

[ScWi86] E. Schulte and J. M. Wills, On Coxeter's regular skew polyhedra, *Discrete Math.* 60 (1986), 253–262.

[SeTh96] P. Seymour and R. Thomas, Uniqueness of highly representative surface embeddings, *J. Graph Theory* 23 (1996), 337–349.

[Si01] J. Siráň, Coverings of graphs and maps, orthogonality, and eigenvectors, *J. Algebraic Combin.* 14 (2001), 57–72.

[Sta80] S. Stahl, Permutations-partition pairs: a combinatorial generalization of graph embedding, *Trans. Amer. Math. Soc.* 259 (1980), 129–145.

[Sti06] E. Steinitz, Über die Eulersche Polyederrelationen, *Arch. Math. Phys.* 11 (1906), 86–88.

[Sti22] E. Steinitz, Polyeder and Raumeinteilungen, *Enzykl. Math. Wiss.* 3 (1922), 1–139.

[Th90] C. Thomassen, Embeddings of graphs with no short noncontractible cycles, *J. Combin. Theory, Ser. B*, 48 (1990) 155–177.

[Th91] C. Thomassen, Tilings of the torus and the Klein bottle and vertex-transitive graphs on a fixed surface, *Trans. Amer. Math. Soc.* 323 (1991), 605–635.

[Th93] C. Thomassen, Five-coloring maps on surfaces, *J. Combin. Theory, Ser. B*, 59 (1993), 89–105.

[Th97] C. Thomassen, Color-critical graphs on a fixed surface, *J. Combin. Theory, Ser. B* 70 (1997), 67–100.

[Thu78] W. P. Thurston, *The Geometry and Topology of 3-manifolds*, Princeton Univ. Lect. Notes, Princeton, NJ, 1978.

[Tu83] T. W. Tucker, Finite groups acting on surfaces and the genus of a group, *J. of Combin. Theory, Ser. B* 34 (1983), 82–98.

[Tut79] W. T. Tutte, Combinatorial oriented maps, *Canad. J. Math.* 5 (1979), 986–1004.

[Tut63] W. T. Tutte, A census of planar maps, *Canad. J. Math.* 15 (1963), 249–271.

[Vi83a] A. Vince, Combinatorial maps, *J. Combin. Theory, Ser. B*, 34 (1983), 1–21.

[Vi83b] A. Vince, Regular combinatorial maps, *J. Combin. Theory, Ser. B*, 35 (1983), 256–277.

[Vi84] A. Vince, Flag transitive maps, *Congressus Numerantium* 45 (1984), 235–250.

[Vi95] A. Vince, Map duality and generalizations, *Ars Combinatoria* 39 (1995), 211–229.

[Wa75] T. R. S. Walsh, Hypermaps versus biparite maps, *J. Combin. Theory, Ser. B*, 18 (1975), 155-163.

[Wh01] A. T. White, *Graphs of Groups on Surfaces, Interactions and Models*, North-Holland Publishing Co., Amsterdam, 2001.

[Whi32] H. Whitney, Nonseparable and planar graphs, *Trans. Amer. Math. Soc.* 34 (1932), 339–362.

[Wi78a] S. Wilson, Non-orientable regular maps, *Ars Combinatoria* 5 (1978), 213–218.

[Wi78b] S. Wilson, Riemann surfaces over regular maps, *Canad. J. Math.* 30 (1978), 763–782.

[Wi89] S. Wilson, Cantankerous maps and rotary embeddings of K_n, *J. Combin. Theory, Ser. B* 3 (1989), 262–273.

[Ya99] L. Yanpei, *Enumerative Theory of Maps*, Kluwer Academic Publishers, Boston, 1999.

[Yu97] X. Yu, Disjoint paths, planarizing cycles, and spanning walks, *Trans. Amer. Math. Soc.* 349 (1997), 1333–1358.

[Zv97] A. Zvonkin, Matrix integrals and map enumeration: an accessible introduction, *Mathl. Comput. Modelling* 26 (1997), 281–304.

7.7 REPRESENTATIVITY

Dan Archdeacon, University of Vermont

7.7.1 Basic Concepts
7.7.2 Coloring Densely Imbeddable Graphs
7.7.3 Finding Cycles, Walks, and Spanning Trees
7.7.4 Re-Imbedding Properties
7.7.5 Minors of Imbedded Graphs
7.7.6 Minor-Minimal Maps
References

Introduction

Consider the graph $C_{100} \times C_{100}$, imbedded on the torus as a grid so that all faces are quadrilaterals. For any vertex v, the graph induced by all vertices of distance at most 49 from v is imbedded exactly as if it were in the plane. In other words, it is locally planar, so it may share some properties of planar graphs. Representativity measures the extent of local planarity of an imbedded graph. Alternatively, an imbedded graph with large representativity may reveal properties of its surface.

The theory of representativity was first introduced by Robertson and Seymour [RoSe88] in their work on graph minors, although hints occur in earlier works. See [GrTu87] and [MoTh01] for some of the proofs and for general reference in topological graph theory and in representativity.

7.7.1 Basic Concepts

There are several different, but related, ways to measure the local planarity of a graph imbedded on a non-spherical surface. We present these measures and their relations.

DEFINITIONS

D1: An ***open face*** (sometimes "open" is omitted) of a graph imbedding $G \to S$ is a connected component of $S - G$. The set of faces is denoted $F(G \to S)$, or sometimes, simply F.

D2: The union of a face f and its boundary walk is denoted $\bar{f}$ and is called a ***closed face***.

D3: A ***cellular*** imbedding is one where each face is homeomorphic to 2-dimensional Euclidean space.

D4: An imbedding is ***circular*** if the boundary of every face is a simple cycle.

D5: An imbedding is ***polyhedral*** if it is circular and the intersection of any two closed faces is a path. (Equivalently, we observe that each vertex has a ***wheel-neighborhood*** – with a possibly subdivided rim.)

D6: A cycle C on a surface S is ***contractible*** if it separates the surface, and if one side of the separation is a disk.

D7: If S is not a sphere, then the side of a contractible cycle C that is a disk is called the ***interior*** of C and is denoted $int(C)$.

D8: A ***k-nest*** of disjoint contractible cycles is a sequence $C_1, \ldots, C_k$ such that $C_i \subset int(C_{i+1})$.

D9: The ***edge-width*** $ew(G)$ of a graph imbedding $G \to S$ is the (graph-theoretic) length of the shortest cycle in the graph that is non-contractible on the surface.

D10: The ***face-width*** $fw(G)$ of a graph imbedding $G \to S$ (alternatively, the ***representativity***, denoted $\rho(G)$) is the minimum cardinality $|C \cap G|$, taken over all non-contractible cycles C in S. That is, it is the smallest number k such that there exists faces $f_1, \ldots, f_k$ with a non-contractible cycle C contained in $\bar{f}_1 \cup \ldots \cup \bar{f}_k$. (The cycle C may intersect vertices, not just edges.) Informally, ***width*** is a synonym.

D11: An imbedded graph is ***dense*** on the surface if it has large face-width, with "large" to be interpreted in context.

D12: The ***medial graph*** $M(G)$ of a graph imbedding $G \to S$ is the imbedded graph whose vertex set is $E(G)$, and whose edge set joins each pair of edges of that are consecutive in a face boundary of G.

D13: The ***radial graph***, $R(G)$, is the imbedded dual of the medial graph. Equivalently, the radial graph has vertex set $V(G) \cup F(G \to S)$, with edges joining incident elements. The radial graph is also called the ***vertex-face*** graph, or the ***vertex-face incidence graph***.

D14: The ***Euler genus***, $\bar{\gamma}(S)$, of a surface S is twice the number of handles if the surface is orientable, and is the number of crosscaps if the surface is non-orientable. (This definition reflects the old maxim "a handle is worth two crosscaps".)

EXAMPLES

E1: The graph in Figure 7.7.1 is drawn in the torus, where the top edge is identified with the bottom edge and the left with the right. It has edge-width 4 as shown by the cycle on the bold edges. It has face-width 2 as shown by the dotted line. All line segments in the boundary of the square are edges in the graph.

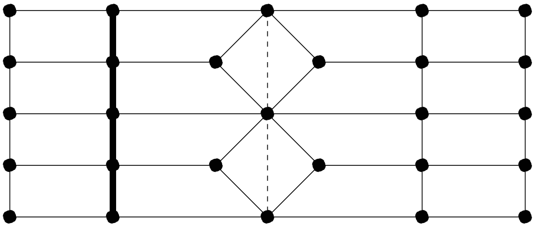


Figure 7.7.1 **A graph with edge-width 4 and face-width 2.**

E2: The graph in the left of Figure 7.7.2 shows K_4 imbedded in the plane. The dotted lines show its medial graph, the octahedron. On the right side is K_4 and its radial graph, the cube.

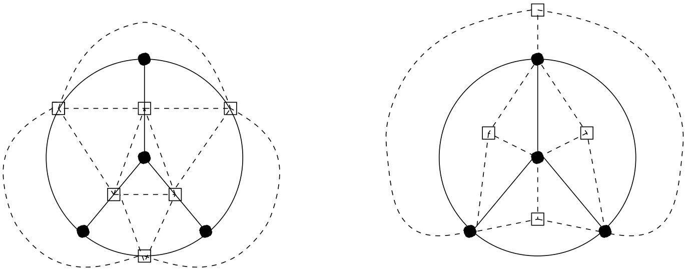


Figure 7.7.2 **The medial graph and radial graph of K_4.**

FACTS

F1: In any cellular imbedding, any curve in the surface is ambient isotopic (or freely homotopic, i.e., can be continuously transformed) to a closed walk in the graph. Moreover, any walk W in the graph is ambient isotopic to a curve C in the surface that meets the graph only at its vertices. The number of points in $C \cap G$ will not exceed the length of W. This gives the following two facts.

F2: For any imbedding $fw(G) \leq ew(G)$.

F3: For triangulations $fw(G) = ew(G)$.

F4: An imbedded graph M is a medial graph of some graph G if and only if M is 4-regular and the faces can be properly 2-colored.

F5: An imbedded graph R is a radial graph of some graph G if and only if R is bipartite and every face is a quadrilateral.

F6: The medial graph M of an imbedded graph G is identical to the medial graph of the dual G^*. The radial graph R of G is identical to the radial graph of the dual G^*. The imbedded graph and its dual are the only two graphs whose medial and radial graphs are M and R respectively.

F7: The face-width of an imbedded graph G is equal to half the edge-width of its radial graph, i.e., $fw(G) = ew(R(G))/2$.

F8: The face-width of an imbedded graph G is equal to the face-width of its dual G^*, i.e., $fw(G) = fw(G^*)$.

F9: An imbedding of a graph G is cellular if and only if G is connected and $fw(G) \geq 1$.

F10: The following are equivalent: 1) an imbedding of a graph G is circular, 2) G is 2-connected and $fw(G) \geq 2$, and 3) the radial graph $R(G)$ has no multiple edges.

F11: The following are equivalent: 1) an imbedding of a graph G is polyhedral; 2) G is 3-connected and $fw(G) \geq 3$; and 3) the radial graph $R(G)$ has no multi-edges and every 4-cycle bounds a face.

F12: In a polyhedral imbedding of a 3-connected graph, the face boundaries are simple chordless cycles.

F13: [RoSe88] 1) Let v be a vertex of an imbedded graph G and let $k = \lfloor (fw(G) - 1)/2 \rfloor$. Then there exists a k-nest with $v \in int(C_1)$. 2) Let f be an open face of an imbedded graph G and let $k = \lfloor fw(G)/2 \rfloor - 1$. Then there exists a k-nest with $f \in int(C_1)$.

See [Mo97] for additional results on face-width.

7.7.2 Coloring Densely Imbeddable Graphs

All planar graphs are vertex-4-colorable. Nonplanar graphs may have arbitrarily high chromatic number; for instance, the chromatic number of the complete graph K_n is n. However, all imbeddings of complete graphs have many non-contractible triangles, which implies that they have edge-width 3, so complete graphs are not densely imbeddable. As an illustration of the relationship between high representativity and planarity, we consider upper bounds on the chromatic numbers of graphs with imbeddings of sufficiently high representativity.

Coloring with Few Colors

We first consider two 5-color theorems for graphs with imbeddings of high edge-width. If we allow six colors, then the required edge-width drops significantly.

FACTS

F14: [AlSt82] (combined with [Th94a]) If a graph G has a toroidal imbedding such that $ew(G) \geq 4$, then G is 5-colorable.

F15: [Th93] If a graph G admits an imbedding $G \to S_g$ such that $ew(G) \geq 2^{14g+6}$, then G is 5-colorable.

F16: [Fi78] Suppose that a graph G has exactly two vertices of odd degree, that these two vertices are adjacent, and it is imbeddable so that every face is a triangle. Then G is not 4-colorable.

F17: [FiMo94] There is a constant c such that every graph G imbeddable on a surface of Euler genus $g > 0$ so that $ew(G) \geq c \log g$ is a 6-colorable graph.

F18: [Th97] For each fixed surface S there is a polynomial-time algorithm that decides if a given graph imbedded on that surface is 5-colorable.

REMARK

R1: It is not hard to construct imbeddings on any surface with exactly two vertices of odd degree that are adjacent, as in Fact 16, so the 5-color theorem of Fact 15 is best possible.

Coloring Graphs That Quadrangulate

We next examine the chromatic number of imbedded graphs with no face a triangle, or more specifically, where every face is of even length. These generalize the classical theorem of Grötzsch, which says that every planar graph of girth at least 4 is 3-colorable.

DEFINITION

D15: A ***quadrangulation*** is a graph imbedding such that each face is a quadrilateral.

FACTS

F19: If every face of a planar graph G is bounded by a cycle of even length, then G is bipartite (and hence has chromatic number 2).

F20: The complete graph K_5 is a non-bipartite graph that can quadrangulate the torus S_1. By iteratively applying cut-and-paste techniques to this imbedding, we can construct quadrangulations of all of the higher surfaces S_n by non-bipartite graphs, hence, of chromatic number at least 3. Similarly, any non-orientable surface has a non-bipartite quadrangulation.

F21: [Hu95] There is a function $f(g)$ such that every graph G imbeddable on an orientable surface of genus g with every face of even size and with $ew(G) \geq f(g)$ has chromatic number at most 3.

F22: [FiMo94] There is a function $f(g)$ such that every graph G of girth at least 4 imbeddable on a non-orientable surface with g crosscaps with $ew(G) \geq f(g)$ has chromatic number at most 4.

F23: [Yo96] Let G be a graph that has a simple quadrangulation of the projective plane. Then G has either chromatic number 2 or 4.

F24: [Yo96] There exist quadrangulations of the projective plane of arbitrarily large edge-width and chromatic number 4.

F25: ([ArHuNaNeOt99] and [MoSe99]) For any non-orientable surface there exist quadrangulations of arbitrarily large edge-width and chromatic number 4.

REMARK

R2: The fact there is no quadrangulation of the projective plane with chromatic number exactly 3 is surprising. So is the contrast between orientable surfaces, where a large width implies 3-chromatic, and non-orientable surfaces, where 4-chromatic is the best possible.

Coloring Graphs That Triangulate

DEFINITIONS

D16: A ***triangulation*** is a graph imbedding with every face a triangle.

D17: An ***Eulerian triangulation*** is a triangulation of a surface such that the skeleton is Eulerian.

FACTS

F26: A graph that triangulates the plane is 3-colorable if and only if it is Eulerian.

F27: [HuRiSe99] There is a function $f(g)$ such that every Eulerian triangulation G of an orientable surface of genus g with $ew(G) \geq f(g)$ has chromatic number at most 4.

F28: [ArHuNaNeOt99] For every non-orientable surface S and every k there exists an Eulerian triangulation G with $ew(G) \geq k$ and with chromatic number at least 5.

PROBLEMS

P1: [Th97] Is there a surface S with chromatically 5-critical triangulations of arbitarily large edge-width?

P2: [Al81] For each fixed surface S, does there exist a constant $c(S)$ such that there is a proper 4-coloring of all but $c(S)$ vertices of any imbedded graph?

P3: For each fixed surface S, is there a polynomial-time algorithm that decides whether a graph imbedded on S is 4-colorable?

P4: (N. Robertson) Does there exist a constant k such that each cubic graph imbedded with face-width at least k is 3-edge-colorable? Grunbaum [Gr69] conjectured that $k = 3$ suffices.

7.7.3 Finding Cycles, Walks, and Spanning Trees

A fundamental result by Tutte [Tu56] says that 4-connected planar graphs are Hamiltonian. We look for analogous theorems for locally planar graphs.

DEFINITIONS

D18: A ***spanning walk*** is a walk that visits every vertex.

D19: A ***spanning k-walk*** is a spanning walk that visits no vertex more than k times.

FACTS

F29: A graph has a Hamiltonian path if and only if it contains a spanning 1-walk.

F30: If a graph G contains a k-walk, then G contains a spanning tree of maximum degree at most $k + 1$.

F31: [Ba66] Every 3-connected planar graph contains a spanning tree of maximum degree 3.

F32: [BrElGaMeRi95] Every 3-connected planar graph imbeddable on the torus or on the Klein bottle contains a spanning tree of maximum degree 3.

F33: Results analogous to Fact 32 do not hold for surfaces of Euler genus three or more. In particular, $K_{3,n}$ quadrangulates a surface with Euler genus $(n-2)/2$ when n is even. If n is at least 8, then any spanning tree of $K_{3,n}$ contains at least one vertex of degree at least 4.

F34: [Yu97] Let G be a 3-connected graph imbedded on a surface of Euler genus g. If $fw(G) \geq 48(2^g - 1)$, then G contains a spanning 3-walk (and hence a spanning tree of maximum degree 4).

F35: [Yu97] Let G be a 4-connected graph imbedded on a surface of Euler genus g. If $fw(G) \geq 48(2^g - 1)$, then G contains a spanning 2-walk (and hence a spanning tree of maximum degree 3, see also [ElGa94]).

F36: [Yu97] Let G be a 5-connected triangulation on a surface of Euler genus g. If $fw(G) \geq 96(2^g - 1)$, then G contains a Hamiltonian cycle (and hence a spanning 1-walk and a spanning tree of maximum degree 2).

CONJECTURE

C1: There exists a function $f(S)$ such that every 5-connected imbedding of a graph G on a surface S with $fw(G) \geq f(S)$ is hamiltonian.

7.7.4 Re-Imbedding Properties

Whitney [Wh33] proved that every 3-connected graph has an essentially unique imbedding in the plane, and he proved a similar theorem about imbeddings of graphs of connectivity 2 (see Fact 42). Do locally planar graphs have similar properties?

LEW-Imbeddings

DEFINITIONS

D20: A ***large edge-width imbedding*** or ***LEW-imbedding*** is an imbedding $G \to S$ whose edge-width is strictly larger than the length of the longest boundary walk.

D21: A ***rooted imbedding*** of a graph is an imbedding with a distinguished vertex v, an edge e incident with vertex v, and a face f incident with edge e.

FACTS

F37: [Th90] A graph has an LEW-imbedding in at most one (homeomorphism type of) surface; and if such an imbedding exists, then it is a minimum Euler genus imbedding for that graph.

F38: [Th90] In a LEW-imbedding of a 3-connected graph G, the face boundaries are chordless cycles of length is strictly less that $ew(G)$.

F39: [Th90] There is a polynomial-time algorithm that given a 3-connected graph G, either constructs an LEW-imbedding of G or concludes that no such imbedding exists.

F40: [BeGaRi94] For each surface S there is a constant c_S such that almost all rooted imbeddings of G in S have $ew(G) \geq c_S \log(|E(G)|)$. (Thus, for a fixed surface, we can expect the edge-width of a random imbedding to be reasonably large.)

Imbeddings and Connectivity

DEFINITIONS

D22: Let G be a graph of connectivity two, and let C_1, C_2 be two subgraphs, each with a cycle, that partition the edge set of G and have only two vertices in common. Suppose that G is imbedded on a surface. Then we can replace the induced imbedding of one component, say C_2, with its mirror image as shown in Figure 7.7.3 below. This is called a ***Whitney 2-flip***, or more succinctly a ***2-flip***.

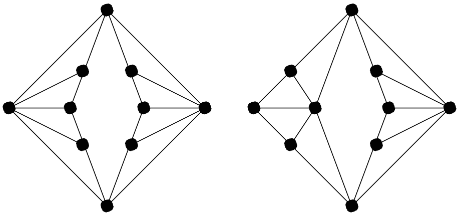


Figure 7.7.3 **A Whitney 2-flip.**

D23: Two imbeddings are ***Whitney-similar*** if one can be obtained from the other by a sequence of Whitney 2-flips.

FACTS

F41: Graphs with a cutpoint do not have unique imbeddings in the plane. If v is a cut-vertex of G incident with blocks B_1 and B_2, then B_2 can be placed in any face of an imbedding of B_1 incident with v.

F42: [Wh33] Any two imbeddings of a 2-connected graph in the plane are Whitney-similar.

F43: [Th90] If a 2-connected graph G has an LEW-imbedding on a surface S, then any other imbedding of G on S is Whitney-similar to it.

F44: [Th90] Let G be a subdivision of a 3-connected graph with an LEW-imbedding in a surface S. Then G is uniquely imbeddable in S, up to homeomorphism of pairs.

Imbeddings and Genus

We ask when an imbedding of sufficiently large face-width is a minimum genus imbedding and when it is unique.

FACTS

F45: [SeTh96] Let G be a graph imbedded on a surface of Euler genus g. If $fw(G) \geq 100 \log g / \log \log g$, then G does not imbed on any surface of smaller Euler genus.

F46: [Ar92] For each integer k there is a graph G_k that has two imbeddings on two different surfaces each of face-width at least k. (Hence in Fact 45 the bound on the face-width cannot be replaced by a constant.)

Re-Imbedding Results

DEFINITION

D24: Let G be a graph imbedded on the Klein bottle. The following ***planarizing orders*** measure how difficult it is to obtain a planar graph by cutting along the (image of the) graph. We define

$$ord_2(G) = \min \lfloor |C|/4 \rfloor$$

taken over all two-sided non-contractible cycles C in the Klein bottle, where $|C|$ denotes the length of C. We define

$$ord_1(G) = \min \lfloor |C_1|/4 \rfloor + \lfloor |C_2|/4 \rfloor$$

taken over all pairs of one-sided non-contractible non-homotopic cycles C_1, C_2.

FACTS

F47: [FiHuRiRo95] Let the graph G be imbeddable in the projective plane with $fw(G) \neq 2$. Then the (orientable) genus of G is $\lfloor fw(G)/2 \rfloor$.

F48: [RoTh91] Let G be a graph imbeddable in the Klein bottle, such that

$$\min\{ord_1(G), ord_2(G)\} \geq 4.$$

Then this minimum is the orientable genus of G.

F49: [RoVi90] and [Th90] Any non-planar imbedding of a planar graph G has face-width less than or equal to 2.

F50: [MoRo98] There is a function $f(g)$ such that every 3-connected graph has at most $f(g)$ imbeddings of face-width at least 3 in a surface of Euler genus g.

F51: No such function $f(g)$ exists if we consider 2-connected graphs, or if we consider imbeddings of face-width at least 2.

For other results on re-imbedding planar graphs see [MoRo96].

CONJECTURE

C2: [FiHuRiRo95]: For each fixed non-orientable surface S, the (orientable) genus of graphs that imbed in S can be computed in polynomial-time.

7.7.5 Minors of Imbedded Graphs

The concept of surface minors and representativity is very important in the sequence of papers by Robertson and Seymour culminating in the proof of Wagner's Conjecture. A nice background is given in [Di00]. A survey of results in this area is given in [Mo01].

Surface Minors

DEFINITIONS

D25: A ***minor of a graph*** G is a graph H that can be formed by a sequence of edge deletions and edge contractions in G.

D26: A graph property P is ***hereditary*** under minors if whenever G has that property, then so does every minor of G. Alternatively, one says that the class of graphs with property P is ***closed*** under the taking of minors.

D27: A ***surface minor*** $H \to S$ of the graph imbedding $G \to S$ is an imbedding of H constructed by a sequence of edge deletions and edge contractions in G. (If edge e is not a loop, then the edge-contracted graph G/e is constructed by contracting e in S. For the purposes of surface minors, we consider the contraction of a loop to be equivalent to its deletion.)

FACTS

F52: For each fixed surface S the property "G imbeds on S" is hereditary under minors. (We do not require that these imbeddings are cellular.)

F53: For each fixed surface S the property "$fw(G) \leq k$" is hereditary under surface minors.

The following results are fundamental in the theory of surface minors.

F54: [RoSe86] For any plane imbedding of a graph G, there is a number k such that G is a surface minor of the natural plane imbedding of the Cartesian product $P_k \times P_k$. See [DiEaTaTo94] for additional references.

F55: [RoSe88] Let $G \to S$ be a graph imbedding on a surface other than the sphere. Then there is a number k such that every graph imbedding on S with face-width at least k has the imbedding $G \to S$ as a surface minor.

Finding Imbedded Cycles

The last fact ensures the existence of any imbedded minor, provided that the width of the given imbedding is large enough. However, the proof is existential and does not provide explicit bounds. Such bounds are known for certain types of graphs.

DEFINITION

D28: A set of cycles $C_1, \ldots, C_k$ in a graph G imbedded on a surface S is a ***planarizing collection*** if cutting along all C_i's simultaneously yields a connected graph imbedded in the plane.

FACTS

F56: [Sc93] Every graph imbedded on the torus with face-width k contains $\lfloor 3k/4 \rfloor$ disjoint non-contractible cycles. (Note that any two disjoint non-contractible cycles in the torus are homotopic.)

F57: [BrMoRi96] Every graph imbedded on a surface with face-width k contains $\lfloor (k-1)/2 \rfloor$ disjoint non-contractible homotopic cycles.

F58: [Th93] Let S be an orientable surface of genus g and let G be a triangulation with $ew(G) \geq 8(d+1)(2^g - 1)$. Then G has a planarizing collection $C_1, \ldots, C_g$ of chordless cycles such that any two of these cycles are of distance at least d.

F59: [Yu97] Let S be a surface with Euler genus g and let G be an imbedded graph with $fw(G) \geq 8(d+1)(2^g - 1)$. Then for some k (with $g/2 \leq k \leq g$), G has a planarizing collection $C_1, \ldots, C_k$ of chordless cycles such that any two of these cycles are of distance at least d.

7.7.6 Minor-Minimal Maps

Fact 52 asserts that for each fixed surface S, the property "$fw(G) \leq k$" is hereditary under surface minors. We look for the minor-minimal imbedded graphs with the property $fw(G) \geq k$, that is, imbedded graphs of face-width k but such that the deletion or surface contraction of any edge lowers the face-width. Any graph imbedded on S with face-width at least k must contain one of these minor-minimal imbeddings as a surface minor. The concept of minor-minimal graphs is often useful in inductive proofs.

DEFINITIONS

D29: A ***minor-minimal imbedded graph*** is an imbedded graph such that the deletion or surface contraction of any edge lowers the face-width.

D30: In a graph G, let v be a vertex adjacent to exactly three other vertices a, b, c. A ***$Y\Delta$-transformation*** deletes v and its three incident edges and adds three new edges ab, bc, ca.

D31: A ***ΔY-transformation*** is the inverse of a $Y\Delta$ operation.

D32: Two graphs are ***$Y\Delta Y$-equivalent*** if there is a sequence of $Y\Delta$- and ΔY-transformations changing one into the other.

FACTS

F60: For every surface S and every $k \geq 1$ the number of minor-minimal maps on S with face-width k is finite.

F61: Let G_Y be a graph and let G_Δ be formed from G_Y by a $Y\Delta$-transformation. If G_Y imbeds on a surface S, then G_Δ also imbeds on S. The converse is not necessarily true, but it is true if the 3-cycle being deleted is a face boundary. (When considering imbedded graphs, the ΔY-transformation is usually restricted to 3-cycles that bound a face.)

F62: If an imbedded G is $Y\Delta Y$-equivalent to an imbedded G', then $fw(G) = fw(G')$. Moreover if G is minor-minimal with face-width k, then so is G'.

F63: [Ra97] Any two graphs in the projective-plane that are minimal with face-width k are $Y\Delta Y$-equivalent. In particular, they have $2k^2 - k$ edges.

F64: [Ba87] There are exactly two graphs in the projective plane that are minimal with face-width 2. They are K_4 and its geometric dual.

F65: ([Ba91] and [Vi92]) There are exactly 7 minor-minimal maps in the projective plane with face-width 3.

F66: [Ba87] There are exactly 7 minor-minimal maps on the torus with face-width 2.

F67: [Hi] There are exactly 56 minor-minimal maps on the torus with face-width 3. These fall into 7 classes under $Y\Delta Y$-equivalence.

Similarity Classes on the Torus

DEFINITION

D33: Two imbeddings are ***similar*** if they are related by a sequence of operations, each a $Y\Delta$-transformations, a ΔY-transformations, or the taking of a geometric dual.

REMARK

R3: In general, the geometric dual of a graph is not necessarily $Y\Delta Y$-equivalent to the primal imbedding. Hence the number of similarity classes might be smaller than the number of $Y\Delta Y$-equivalence classes.

FACT

F68: [Sc94] For odd k there are exactly $(k^3 + 5k)/6$ similarity classes of maps on the torus with face-width k. For even k there are exactly $(k^3 + 8k)/6$ similarity classes.

Kernels

A "kernel" for a surface is an imbedding such that deleting or contracting any edge lowers the face-width in some direction, in the sense that a free homotopy class of closed curvers may be regarded as a direction.

DEFINITIONS

D34: Let $G \to S$ be a graph imbedding, and let C be a curve in the surface S. We define the ***μ-function*** $\mu(G, C)$ as

$$\min\{|W|/2\}$$

where this minimum ranges over all closed walks W in the radial graph $R(G)$ that are freely homotopic to C. This is similar to the face-width of G, $\min\{|C' \cap G|\}$, except that we now restrict the minimum to those curves C' freely homotopic to C.

D35: An imbedded graph G is a ***kernel*** if for every proper imbedded minor H, there is a curve C such that $\mu(H, C) < \mu(G, C)$.

FACTS

F69: $\mu(G, C) = \mu(G^*, C)$ where G^* is the geometric dual.

F70: $\mu(G, C)$ is invariant under $Y\Delta$-exchanges.

F71: If H is a surface minor of G, $\mu(H, C) \leq \mu(G, C)$ for any curve C.

F72: ([Sc92] and [Gr94]) Suppose that G and G' are kernels on the same surface such that $\mu(G, C) = \mu(G', C)$ for all curves C. Then G' can be obtained from G by a sequence of $Y\Delta$ exchanges and the taking of duals.

References

[Al81] M. O. Albertson, Open Problem 2, p. 609 in *The Theory and Applications of Graphs*, ed. by G. Chartrand et al., Wiley, 1981.

[AlSt82] M. O. Albertson and W. R. Stromquist, Locally planar toroidal graphs are 5-colorable, *Proc. Amer. Math. Soc.* 84 (1982), 449–457.

[Ar92] D. Archdeacon, Densely embedded graphs, *J. Combin. Theory Ser. B* 54 (1992), 13–36.

[ArHuNaNeOt99] D. Archdeacon, J. P. Hutchinson, A. Nakamoto, S. Negami, and K. Ota, Chromatic numbers of quadrangulations on closed surfaces, *J. Graph Theory* 37 (2001), 100–114.

[Ba66] D. W. Barnette, Trees in polyhedral graphs, *Canad. J. Math.* 18 (1966), 731–736.

[Ba87] D. W. Barnette, Generating closed 2-cell embeddings in the torus and the projective plane, *Discr. Comput. Geom.* 2 (1987), 233–247.

[Ba91] D. W. Barnette, Generating projective plane polyhedral maps, *J. Combin. Theory Ser. B* 51 (1991), 277–291.

[BeGaRi94] E. A. Bender, Z. Gao, and L. B. Richmond, Almost all rooted maps have large representativity, *J. Graph Theory* 18 (1994), 545–555.

[BrElGaMeRi95] R. Brunet, M. N. Ellingham, Z. Gao, A. Metzlar, R. B. Richter, Spanning planar subgraphs of graphs in the torus and Klein bottle, *J. Combin. Theory Ser. B* 65 (1995), 7–22.

[BrMoRi96] R. Brunet, B. Mohar, and R. B. Richter, Separating and nonseparating disjoint homotopic cycles in graph embeddings, *J. Combin. Theory Ser. B* 66 (1996), 201–231.

[DiEaTaTo994] G. DiBattista, P. Eades, R. Tamassia, I. G. Tollis, Algorithms for drawing graphs: An annotated bibliography, *Comput. Geom.* 4 (1994), 235–282.

[Di00] R. Diestel, *Graph Theory*, Second edition, Springer-Verlag, 2000.

[ElGa94] M. N. Ellingham and Z. Gao, Spanning trees in locally planar triangulations, *J. Combin. Theory Ser. B* 61 (1994), 178–198.

[FiHuRiRo95] J. R. Fiedler, J. P. Huneke, R. B. Richter, and N. Robertson, Computing the orientable genus of projective graphs, *J. Graph Theory* 20 (1995), 297–308.

[FiMo94] S. Fisk and B. Mohar, Coloring graphs without short non-bounding cycles, *J. Combin. Theory Ser. B* 60 (1994), 268–276.

[Gr94] M. de Graaf, Graphs and curves on surfaces, Ph.D. Thesis, University of Amsterdam, Amsterdam, 1994.

[GrTu87] J. L. Gross and T. W. Tucker, *Topological Graph Theory*, Wiley-Interscience, 1987. Dover reprint edition, 2001.

[Gr69] B. Grünbaum, Conjecture 6, p. 343 in *Recent Progress in Combinatorics*, ed. by W. T. Tutte, Academic Press, 1969.

[Ha77] G. Haggard, Edmonds' characterization of disc embeddings, *Congr. Numer.* 19 (1977), 291–302.

[Hi] Y. Hirachi, Minor-minimal 3-representative graphs on the torus, Master's Thesis, Yokohama National University, Yokohama, 1996.

[Hu95] J. P. Hutchinson, Three-coloring graphs embedded on surfaces with all faces even-sided, *J. Combin. Theory Ser. B* 65 (1995), 139–155.

[HuRiSe84] J. P. Hutchinson, R.B. Richter, and P.D. Seymour, Coloring Eulerian triangulations, *J. Combin. Theory Ser. B* 84 (2002), 225–239.

[Ja85] F. Jaeger, A survey of the cycle double cover conjecture, in "Cycles in Graphs" (B. Alspach and C. Godsil, Eds.), *Ann. Discrete Math.* 27 (1985), 1–12.

[Mo97] B. Mohar, Face-width of embedded graphs, *Math. Slovaca* 47 (1997), 3–63.

[Mo01] B. Mohar, Graph minors and graphs on surfaces, in "Surveys in Combinatorics", ed. by J. W. P. Hirschfeld, London Mathematical Society Lecture Note Series 288 (2001), 145–163.

[MoRo96] B. Mohar and N. Robertson, Planar graphs on nonplanar surfaces, *J. Combin. Theory Ser. B* 68 (1996), 87–111.

[MoRo98] B. Mohar and N. Robertson, Flexibility of polyhedral embeddings of graphs in surfaces, *J. Combin. Theory Ser. B* 83 (2001), 38–57.

[MoSe99] B. Mohar and P. D. Seymour, Coloring locally bipartite graphs on nonorientable surfaces, *J. Combin. Theory Ser. B* 84 (2002), 301–310.

[MoTh01] B. Mohar and C. Thomassen, *Graphs on Surfaces*, Johns Hopkins University Press, 2001.

[Ra97] S. P. Randby, Minimal embeddings in the projective plane, *J. Graph Theory* 25 (1997), 153–163.

[RoSe86] N. Robertson and P. D. Seymour, Graph minors V: Excluding a planar graph, *J. Combin. Theory Ser B* 41 (1986), 92–114.

[RoSe88] N. Robertson and P. D. Seymour, Graph minors VII: Disjoint paths on a surface, *J. Combin. Theory Ser B* 48 (1990), 255–288.

[RoTh91] N. Robertson and R. Thomas, On the orientable genus of graphs embedded in the Klein bottle, *J. Graph Theory* 15 (1991), 407–419.

[RoVi90] N. Robertson and R. Vitray, Representativity of surface embeddings, pp. 293–328 in *Paths, Flows, and VLSI-Layout*, ed. by B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, Springer-Verlag, Berlin, 1990.

[Sc92] A. Schrijver, On the uniqueness of kernels, *J. Combin. Theory Ser. B* 55 (1992), 146–160.

[Sc93] A. Schrijver, Graphs on the torus and geometry of numbers, *J. Combin. Theory Ser. B* 58 (1993), 147–158.

[Sc94] A. Schrijver, Classification of minimal graphs of given face-width on the torus, *J. Combin. Theory Ser. B* 61 (1994), 217–236.

[SeTh96] P. D. Seymour and R. Thomas, Uniqueness of highly representative surface embeddings, *J. Graph Theory* 23 (1996), 337–349.

[Th99] R. Thomas, Recent excluded minor theorems for graphs, pp. 201–222 in *Surveys in Combinatorics, 1999*, ed. by J. D. Lamb and D. A. Preece, Cambridge University Press, 1999.

[Th90] C. Thomassen, Embeddings of graphs with no short noncontractible cycles, *J. Combin. Theory Ser. B* 48 (1990), 155–177.

[Th93] C. Thomassen, Five-coloring maps on surfaces, *J. Combin. Theory Ser. B* (1993), 89–105.

[Th94a] C. Thomassen, Five-coloring graphs on the torus, *J. Combin. Theory Ser. B* 62 (1994), 11–33.

[Th94b] C. Thomassen, Grötzsch's 3-color theorem and its counterparts for the torus and the projective plane, *J. Combin. Theory Ser. B* 62 (1994), 268–279.

[Th95] C. Thomassen, Embeddings and minors, pp. 301–349 in *Handbook of Combinatorics*, Elsevier, Amsterdam, 1995.

[Th97] C. Thomassen, Color-critical graphs on a fixed surface, *J. Combin. Theory Ser. B* 70 (1997), 67–100.

[Tu56] W. T. Tutte, A theorem on planar graphs, *Trans. Amer. Math. Soc.* 82 (1956), 99–116.

[Vi92] R. Vitray, The 2- and 3-representative projective planar embeddings, *J. Combin. Theory Ser. B* 54 (1992), 1–12.

[Wh33] H. Whitney, 2-isomorphic graphs, *Amer. J. Math.* 55 (1933), 245–254.

[Yo96] D. A. Youngs, 4-Chromatic projective graphs, *J. Graph Theory* 21 (1996), 219–227.

[Yu97] X. Yu, Disjoint paths, planarizing cycles, and spanning walks, *Trans. Amer. Math. Soc.* 349 (1997), 1333–1358.

7.8 TRIANGULATIONS

Seiya Negami, Yokohama National University, Japan

7.8.1 Basic Concepts
7.8.2 Constructing Triangulations
7.8.3 Irreducible Triangulations
7.8.4 Diagonal Flips
7.8.5 Rigidity and Flexibility
References

Introduction

Triangles can be used as elementary pieces to build up a surface. Such construction of a surface generalizes to a simplicial complex in combinatorial topology. In topological graph theory, we regard the skeleton of a triangulation as a graph dividing a surface into triangles.

7.8.1 Basic Concepts

Although we are primarily interested in closed surfaces, it is worth noting that the basic concepts for triangulations on closed surfaces also work for surfaces-with-boundary, after suitable modifications. Imbeddings are implicitly taken to be cellular.

DEFINITIONS

D1: A ***triangulation of a closed surface*** S is an imbedding $\rho : G \to S$ of a simple graph G, such that

- each face is bounded by a 3-cycle, and
- any two faces share at most one edge.

The latter condition excludes K_3 on the sphere.

D2: A graph imbedding $G \to S$ is ***triangular*** if every face is 3-sided. (Some triangular imbeddings are *not* triangulations: e.g., perhaps the skeleton is not simple, or perhaps a face-boundary has a repeated edge.)

D3: The ***skeleton of an imbedding*** $\rho : G \to S$ is the image $\rho(G)$ of the imbedded graph. Informally we write G for $\rho(G)$.

D4: A graph is said to ***triangulate*** a surface if it can be imbedded on the surface as a triangulation.

D5: A ***triangulation*** $G \to S$ ***of a surface-with-boundary*** S is subject to the additional requirement that each boundary component of the surface is the image of a cycle of the skeleton.

D6: A ***Catalan triangulation*** is a triangulation $G \to S$ of a surface-with-boundary such that every vertex of G lies in a boundary component of S. (E. Catalan counted the number of such triangulations when the surface S is a disk.)

D7: The ***link in a triangulation*** $\rho : G \to S$ of a vertex v is the cycle through the neighbors of v whose edges lie on triangles incident on v. It is usually denoted by $\mathrm{lk}(v, \rho : G \to S)$, or simply by $\mathrm{lk}(v)$.

D8: The ***star neighborhood in a triangulation*** $\rho : G \to S$ of a vertex v is the wheel subgraph in G obtained by joining v to each vertex in its link. It is denoted by $\mathrm{st}(v)$.

D9: A ***clean triangulation*** is a triangulation $G \to S$ such that every 3-cycle in G bounds a face.

D10: Two triangulations $\rho_1 : G \to S$ and $\rho_2 : G \to S$ with the same vertex set are ***combinatorially equivalent*** if they have the same set of face boundary cycles. (Precisely speaking, combinatorial equivalence is for labeled triangulations.)

D11: Two triangulations $G_1 \to S$ and $G_2 \to S$ are ***isomorphic*** if there is a homeomorphism $h : S \to S$ such that $h(G_1) = G_2$.

D12: Two triangulations $G_1 \to S$ and $G_2 \to S$ are ***isotopic*** if there is a homeomorphism $h : S \to S$ with $h(G_1) = G_2$ that is isotopic to the identity mapping on S. (Roughly speaking, this means that one can be transformed continuously on the surface into the other.)

D13: An imbedding is said to be ***k-representative*** if it has *face-width* at least k. (See § 7.7.)

NOTATIONS

NOTATION: S_g denotes the orientable closed surface of genus g, as usual in topological graph theory, and N_k denotes the nonorientable closed surface of crosscap number k. The Euler characteristic of a surface S is denoted by $\chi(S)$.

NOTATION: $F(G \to S)$ denotes the set of faces of a triangulation $G \to S$. However, we usually let $F(G)$ or F denote the set of faces, when only one imbedding of the graph G is under consideration.

NOTATION: Each face of a triangulation $G \to S$ can be specified by listing the three vertices u, v, w at its corners. Thus, it is often identified with the triple $\{u, v, w\}$, and hence one may write $F(G) \subset \binom{V(G)}{3}$. In other contexts, a face may be denoted by its boundary cycle uvw.

EXAMPLES

E1: The 1-skeletons of the tetrahedron, the octahedron and the icosahedron all triangulate the sphere.

E2: The unique imbedding of K_7 on the torus, as shown in Figure 7.8.1, is a triangulation. (Its dual graph is the Heawood graph.)

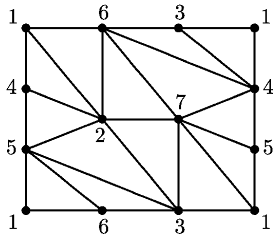


Figure 7.8.1 **The complete graph K_7 on the torus.**

E3: The unique imbedding of K_6 on the projective plane, as shown in Figure 7.8.2, is a triangulation. (Its dual is isomorphic to the Petersen graph.)

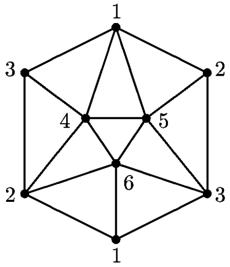


Figure 7.8.2 **The complete graph K_6 on the projective plane.**

E4: There exists only one 7-vertex triangulation on the torus, up to isomorphism, but there exist infinitely many up to isotopy. The skeleton is isomorphic to K_7 as a graph. Twisting it along a simple closed curve yields an infinite series of those. There exist 120 toroidal triangulations over the vertex set $\{1, \ldots, 7\}$ up to combinatorial equivalence.

E5: There exists only one 6-vertex triangulation on the projective plane, up to isomorphism, and also only one up to isotopy, since any auto-homeomorphism on the projective plane is isotopic to the identity mapping. The skeleton is isomorphic to K_6 as a graph. There exist 12 projective-planar triangulations over the vertex set $\{1, \ldots, 6\}$ up to combinatorial equivalence.

FACTS

F1: A graph with at least four vertices can triangulate the sphere if and only if it is maximal planar.

F2: A graph is isomorphic to a Catalan triangulation of the disk if and only if it is maximal outer-planar.

F3: A graph cellularly imbedded on the sphere is a triangulation if and only if its dual is 3-regular and 3-edge-connected.

F4: A graph cellularly imbedded on a closed surface other than the sphere is a triangulation if and only if its dual is 3-regular, 3-edge-connected, and 3-representative.

F5: The link of a vertex in the interior of the surface is a cycle, and the link of a vertex at the boundary is a path.

F6: Every triangulation of any closed surface has a 3-connected skeleton. More generally, a graph is 3-connected if for each vertex v all the neighbors of v lie on a cycle.

F7: The skeleton G of a triangulation on a closed surface is n-connected ($n = 4$ or 5) if and only if every cycle in G that separates the surface into two pieces, each of which includes at least one vertex, has length at least n.

F8: Every triangulation on any closed surface except the sphere is 3-representative.

F9: A triangulation on a closed surface except the sphere is 4-connected, and it is 4-representative if and only if it is clean.

F10: A triangulation with n vertices on a closed surface S with Euler characteristic $\chi(S)$ has exactly $3(n - \chi(S))$ edges and $2(n - \chi(S))$ faces.

F11: Let G be a triangulation on a closed surface S with Euler characteristic $\chi(S)$ and let V_i denote the number of vertices of degree i in G. Then we have:

$$\sum_{i\geq 3}(6 - i)V_i = 6\chi(S)$$

F12: The equivalence up to isomorphism can be rephrased combinatorially as follows; two triangulations G_1 and G_2 are isomorphic if there is a graph isomorphism $\varphi : V(G_1) \to V(G_2)$ which induces a bijection $\varphi^* : F(G_1) \to F(G_2)$ with $\varphi^*(\{u, v, w\}) = \{\varphi(u), \varphi(v), \varphi(w)\}$.

F13: Let $G_1 \to S_1$ and $G_2 \to S_2$ be triangulations, and let f_1 and f_2 be triangles in G_1 and G_2, respectively. Discard the interiors of those two faces, and paste the boundary of f_1 to the boundary of f_2, thereby producing a connected surface with an imbedded graph. That resulting imbedding is a triangulation.

EXAMPLE

E6: Let $K \to S$ and $H \to S'$ be two 6-connected triangulations on different closed surfaces. Let v be a vertex of degree $d \geq 6$ in K with link $v_1v_2 \cdots v_d$, and choose two faces of $H \to S'$ sufficiently apart from each other. Identify these with two faces vv_1v_2 and vv_4v_5 of the imbedding $K \to S$. Then the resulting triangulation has a 5-cut $\{v, v_1, v_2, v_4, v_5\}$, but it contains no separating cycle of length less than 6. Therefore, Fact 7 does not hold for $n = 6$.

7.8.2 Constructing Triangulations

What is the minimum number of triangles needed to build up a given surface? This question must have been asked frequently, but it is difficult to answer it precisely.

Triangulations with Complete Graphs

The solution of the "Map Color Theorem" gave us the precise formulas of the genus and the nonorientable genus of K_n; namely

$$\gamma(K_n) = \lceil (n-3)(n-4)/12 \rceil \quad \text{and} \quad \tilde{\gamma}(K_n) = \lceil (n-3)(n-4)/6 \rceil$$

The complete graph K_n triangulates a suitable surface exactly when the inside of each ceiling function becomes an integer. The constructions give us triangulations on many closed surfaces and also a hint to answer the minimum triangulation question.

DEFINITION

D14: A triangulation is said to be ***tight*** if

- the skeleton G is a complete graph, and
- for any partition of $V(G)$ into three nonempty subsets V_1, V_2 and V_3, there is a face $v_1v_2v_3 \in F(G)$ with $v_i \in V_i$.

It is ***untight*** otherwise.

FACTS

F14: [Ri74] The complete graph K_n over n vertices triangulates an orientable closed surface if and only if $n \equiv 0, 3, 4$ or $7 \pmod{12}$. The genus of such a surface is equal to $(n-3)(n-4)/12$.

F15: [Ri74] The complete graph K_n over n vertices triangulates a nonorientable closed surface if and only if $n \equiv 0, 1, 3$ or $4 \pmod{6}$ and $n \neq 7$. The genus of such a surface is equal to $(n-3)(n-4)/6$.

F16: [Fr34] No complete graph triangulates the Klein bottle.

F17: [BrSt01] The minimum order of a complete graph that admits nonisomorphic triangulations on an orientable closed surface is 9; for the nonorientable case, the minimum is 12.

F18: [BoGrGrSi00] The complete graph K_n triangulates an orientable closed surface with bipartite duals in at least $2^{n^2/54-O(n)}$ ways if $n \equiv 7$ or $19 \pmod{36}$ and in at least $2^{2n^2/81-O(n)}$ ways if $n \equiv 19$ or $55 \pmod{108}$.

EXAMPLES

E7: [LaNeWh94] The complete graph K_{19} triangulates the orientable closed surface S_{20} in at least three ways.

E8: [BrSt01] The complete graph K_{10} triangulates the nonorientable closed surface N_7 in at least 14 ways.

E9: [ArBrNe95] The complete graph K_{30} triangulates the nonorientable closed surface N_{117} in at least 2 ways; they are tight and untight.

REMARK

R1: A tight triangulation is necessarily isomorphic to a complete graph as a graph. As a natural generalization of the tightness which works for general triangulations, a notion called the "looseness" has been introduced in [NeMi96] so that a tight triangulation has looseness 0.

Minimum Triangulations

Here we shall show the answer to our question on the minimum number of triangles to build up a surface. The corresponding formula is expressed below in terms of the number of vertices.

DEFINITION

D15: A ***minimum triangulation*** of a surface is a triangulation on the surface that has the fewest vertices (or equivalently, the fewest faces).

FACTS

F19: [JuRi80, Ri55] Let $V_{\min}(S)$ denote the order of minimum triangulations of a closed surface S. If $S \neq S_2$, N_2, N_3, then:

$$V_{\min}(S) = \left\lceil \frac{7+\sqrt{49-24\chi(S)}}{2} \right\rceil$$

For the three exceptions, we have:

$$V_{\min}(S_2) = 10, \quad V_{\min}(N_2) = 8, \quad V_{\min}(N_3) = 9$$

F20: If the complete graph K_n triangulates a closed surface, then the skeleton of any minimum triangulation is isomorphic to K_n.

F21: [HaRi91] The minimum number of faces in a clean triangulation of S_2 is 24.

F22: [HaRi91] The minimum number of faces in a clean triangulation of S_g is asymptotically equal to $4g$ as $g \to \infty$.

EXAMPLES

E10: The only minimum triangulations of the sphere, the projective plane and the torus are the unique imbeddings of K_4, K_6 and K_7, respectively.

E11: There exist precisely six minimum triangulations of the Klein bottle, up to isomorphism.

Covering Constructions

Shortly after the solution of Map Color Theorem, the theory of voltage graphs (see § 7.4) provided a unified topological analysis of that solution, as a "branched covering", and an extensive generalization of its constructive method. In fact, many triangular imbeddings of complete graphs constructed for Map Color Theorem can now be obtained as coverings of small graphs that triangulate suitable surfaces. There are also other ways to build triangulations from triangulations.

DEFINITIONS

D16: [Gr74] A ***voltage graph*** $\langle G = (V, E), \alpha \rangle$ is a directed graph G with an assignment $\alpha : E \to \mathcal{B}$ of elements of a group $\mathcal{B}$ to its arcs. The group $\mathcal{B}$ is called the ***voltage group***.

D17: The ***net voltage on a walk*** in a graph is the product (or sum, if the voltage group is abelian) of the voltages along that walk.

D18: The ***Kirchhoff voltage law*** (abbr. KVL) holds for an imbedded voltage graph if on every face boundary walk, the net voltage equals the identity of the voltage group.

D19: The ***composition*** $G[H]$ of a graph G with a graph H is the graph with vertex set $V(G) \times V(H)$ such that (u_1, v_1) is adjacent to (u_2, v_2) whenever either u_1 is adjacent to u_2, or v_1 is adjacent to v_2 with $u_1 = u_2$. In particular, we denote $G[\overline{K}_m]$ simply by $G_{(m)}$, where $\overline{K}_m$ is the graph over m vertices with no edge.

D20: A natural projection $p : G_{(m)} \to G$ is called a ***covering with folds***. This is an m-to-1 surjective homomorphism mapping (u, v) to u for each vertex $u \in V(G)$. (There will be a more general definition of a covering with folds in other contexts.)

FACTS

F23: [Gr74] An imbedded voltage graph $\langle G \to S, \alpha \rangle$ "lifts" to an imbedding $G^\alpha \to S^\alpha$ of a covering graph of G into a branched covering of the surface S, such that branch points occur only in the interiors of the faces, with at most one branch point per face.

F24: [Gr74] Let $G \to S$ be a triangular imbedding with voltage assignment α such that the Kirchhoff voltage law holds. Then the resulting graph imbedding in the covering surface of S is also a triangular imbedding. (If the covering graph G^α is simple and not K_3, then the resulting imbedding is a triangulation.)

F25: [Bo82a] Let G be a triangulation on a closed surface S. If a positive integer m is not divisible by 2, 3 or 5, then $G_{(m)}$ triangulates another closed surface with the same orientability as S.

F26: [Bo82b] If a triangulation G on a closed surface S is eulerian, then $G_{(m)}$ triangulates another closed surface with the same orientability as S.

F27: [Ar92] If the complete graph K_n triangulates a closed surface S and if each prime factor of m is at least $n - 1$ except the case of $n = 4$, $m = 3$, then $K_{n(m)}$ triangulates another closed surface, where $K_{n(m)}$ stands for the n-partite graph $K_{m,\ldots,m}$ with partite sets of size m.

EXAMPLE

E12: We observe in Figure 7.8.3 that the imbedding $B_3 \to S_1$ is a KVL triangular imbedding. Moreover, for any cyclic group $\mathbb{Z}_n$, and that for $n \geq 7$, the covering graph is simple. Accordingly, Fact 24 implies that the covering imbedding is a triangulation. For sufficiently large n, the face-width of the covering imbedding is arbitrarily large.

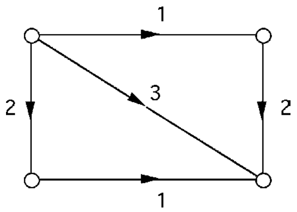


Figure 7.8.3 **A KVL imbedding $B_3 \to S_1$ with voltages in arbitrary $\mathbb{Z}_n$.**

7.8.3 Irreducible Triangulations

Look at one edge in a triangulation. There are two triangles incident to the edge from both sides and two wheels cover them. With such a local picture around an edge, one will guess that shrinking this edge yields another triangulation smaller than the original. Moreover, he might consider that this fact can be used for some proofs with induction on the number of vertices or edges. What is the first step of such induction? That is, what can we get, repeating this deformation as far as possible? "Irreducible triangulations" are exactly the answer. There is a strong connection with studies on irreducible triangulations and graph minor theory.

Edge Contraction

DEFINITIONS

D21: Let ac be an edge in a triangulation $G \to S$, and let acb and acd be the two faces sharing the edge ac. ***Contraction of the triangulation on the edge*** ac is to shrink the adjacent triangles acb and acd to a path $bad = bcd$, as in Figure 7.8.4. We do not contract an edge of a triangulation unless it results in another triangulation on the surface S.

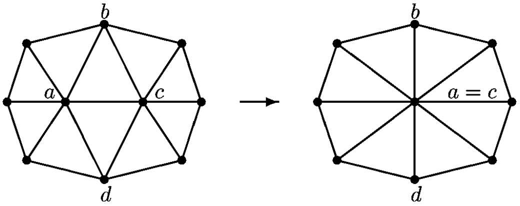


Figure 7.8.4 Edge contraction.

D22: A ***vertex splitting*** is the inverse operation of an edge contraction.

D23: A ***contractible edge in a triangulation*** $G \to S$ on a closed surface is an edge whose contraction would yield another triangulation on S.

D24: A triangulation $G \to S$ is said to be ***contractible to another triangulation*** $H \to S$ if it can be obtained from $G \to S$ by a sequence of edge contractions.

FACTS

F28: An edge in a triangulation G on a closed surface, except K_4 on the sphere, is contractible if and only if it is contained in exactly two cycles of length 3, which are the boundary cycles of two faces sharing the edge.

F29: [Ne94] A triangulation $G \to S$ is contractible to a triangulation $H \to S$ of the same surface if and only if the skeleton H of the latter triangulation is a minor of the graph G.

REMARK

R2: An edge contraction in a triangulation is different from that in graph minor theory. The former always decreases the number of edges by 3 at a time, while the latter does so by 1 at a time.

Classification and Finiteness in Number

DEFINITION

D25: An ***irreducible triangulation*** on a closed surface is one that has no contractible edge.

FACTS

F30: Every irreducible triangulation on any closed surface, except the sphere, has minimum degree at least 4.

F31: Every triangulation is contractible to an irreducible triangulation. Equivalently, it can be obtained from an irreducible triangulation by a sequence of vertex splittings.

F32: [StRa34] The only irreducible triangulation on the sphere is the tetrahedron, whose skelton is isomorphic to K_4.

F33: [Ba82] There are precisely two irreducible triangulations on the projective plane up to isomorphism, shown in Figure 7.8.5. Their skeletons are isomorphic to K_6 and $K_4 + \overline{K}_3$.

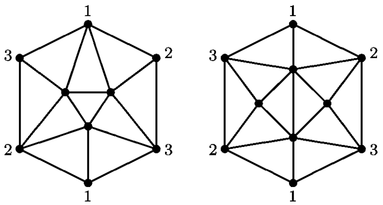


Figure 7.8.5 **The two irreducible triangulations on the projective plane.**

F34: [La87] There are precisely 21 irreducible triangulations on the torus up to isomorphism, which are given in Figure 7.8.6 below, where each pair of parallel sides of the rectangles should be identified. They are denoted by T1 to T21. The skeleton of T1 is isomorphic to K_7.

F35: [LaNe97] There are precisely 25 irreducible triangulations on the Klein bottle up to isomorphism. They are classified into two classes, namely ***handle types*** Kh1 to Kh21 in Figure 7.8.7 below and ***crosscap types*** Kc1 to Kc4 in Figure 7.8.8 below. Identify each horizontal pair of sides in parallel and each vertical pair in anti-parallel to recover the handle types and identify each antipodal pair of vertices lying on the hexagons for crosscap types.

F36: [BaEd89] There are only finitely many irreducible triangulations on any closed surface, up to isomorphism.

F37: [NaOt95] Any irreducible triangulation of a closed surface S with $\chi(S) \leq 0$ has at most $171(2 - \chi(S)) - 72$ vertices.

EXAMPLES

E13: All minimal triangulations on a closed surface are irreducible, but there are irreducible triangulations that are not minimal, in general.

E14: Triangulations of closed surfaces by K_n and $K_{n(m)}$ are irreducible.

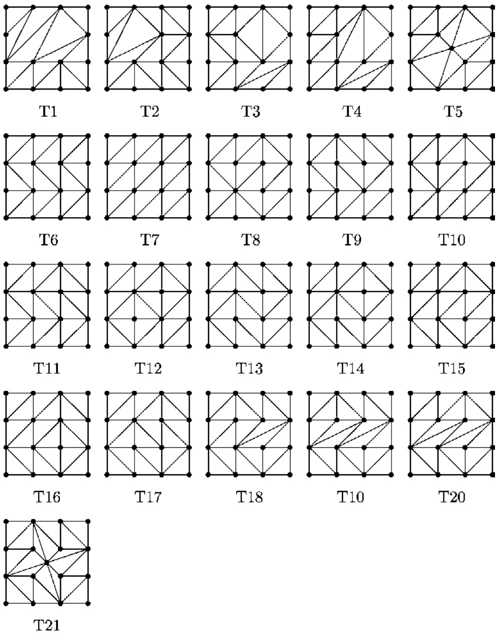


Figure 7.8.6 The 21 irreducible triangulations on the torus.

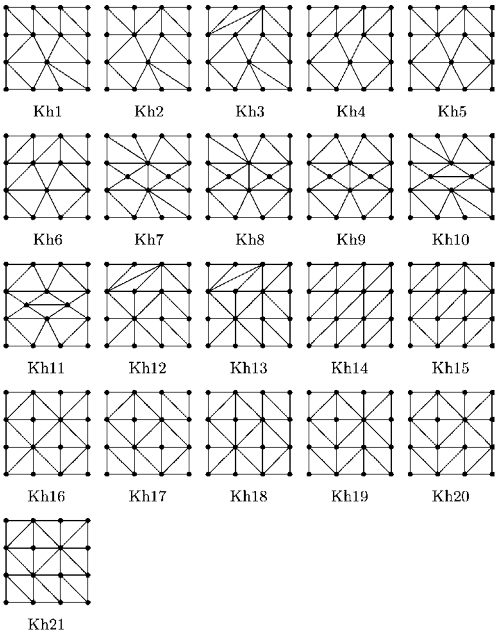


Figure 7.8.7 The 21 irreducible triangulations on N_2, handle types.

E15: Let $G_1 \to S_1$ and $G_2 \to S_2$ be two irreducible triangulations on closed surfaces. Form the connected sum of these triangulations by discarding the interiors of a face in each and pasting on the boundary cycles. The resulting triangulation on the surface $S_1\#S_2$ is irreducible. In particular, all four crosscap types of irreducible triangulations on the Klein bottle are obtained from two of those on the projective plane in this way.

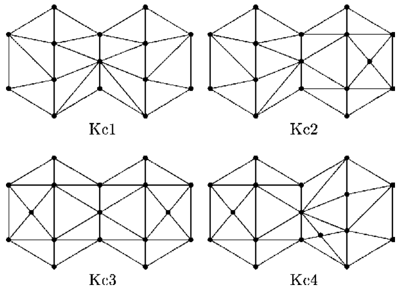


Figure 7.8.8 **The 4 irreducible triangulations on N_2, crosscap types.**

REMARKS

R3: No pair of skeletons for distinct irreducible triangulations on the torus are isomorphic as graphs.

R4: The only pair of isomorphic skeletons for distinct irreducible triangulations on the Klein bottle is (Kh2, Kh5).

R5: The only pair of isomorphic skeletons for distinct irreducible triangulations on the torus and the Klein bottle is (T3, Kh1).

Other Irreducibility

There are some studies on triangulations with prescribed properties which are irreducible or minimal with respect to edge contractions; those of higher representativity or with large minimum degree, for example.

DEFINITIONS

D26: An ***essential cycle in a graph imbedding*** on a closed surface is a cycle that bounds no 2-cell region on S.

D27: A ***k-irreducible triangulation*** on a closed surface is a triangulation such that each edge is contained in an essential cycle of length at least k.

D28: A clean triangulation is ***minimal*** if no edge contraction results in a clean triangulation. (This is the same as the 4-irreducibility.)

FACTS

F38: On any closed surface S except the sphere, every triangulation containing no essential cycle of length less than k can be obtained from some k-irreducible triangulation by a sequence of vertex splittings.

F39: [MaNe95] There are only finitely many k-irreducible triangulations on any closed surface, up to isomorphism.

F40: [MaMo92] There are only finitely many minimal clean triangulations on any closed surface, up to isomorphism.

F41: [FiMoNe94] There are exactly five minimal clean triangulations on the projective plane.

F42: [NaNe02] On any closed surface S except the sphere, every triangulation with minimum degree at least 4 is obtainable from some irreducible triangulation by splitting vertices preserving their degrees at least 4 and adding octahedra to faces.

F43: [NaNe02] Every triangulation on the sphere with minimum degree at least 4 can be obtained from the octahedron in the same way as in Fact 42.

REMARKS

R6: On any closed surface S except the sphere, every irreducible triangulation is 3-irreducible.

R7: The finiteness of the number of minor-minimal k-representative imbeddings follows from Fact 39 ([MaNe95], where a k-irreducible triangulation is called a ***k*-minimal triangulation**). A simple proof of the same result can be found in [GaRiSe96].

7.8.4 Diagonal Flips

An edge in a triangulation can be regarded as a diagonal in a quadrilateral, and switched to the other diagonal, which is called "a diagonal flip". All sufficiently large triangulations of the same size differ from each other only by diagonal flips ([Ne94]).

DEFINITIONS

D29: A ***diagonal flip*** of an edge ac shared by triangles acb and acd in a triangulation G means to replace ac with the other diagonal bd in the quadrilateral $abcd$, as shown in Figure 7.8.9. We do not perform a diagonal flip unless the resulting skeleton would be a simple graph.

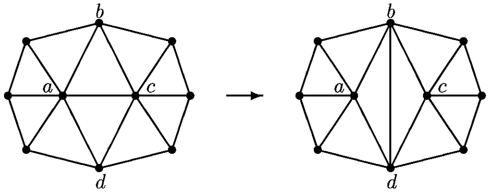


Figure 7.8.9 Diagonal flip.

D30: Two triangulations on a surface are ***equivalent under diagonal flips*** if one can be transformed into the other by a finite sequence of diagonal flips.

D31: A ***frozen triangulation*** is a triangulation such that no edge can be flipped.

D32: The ***standard triangulation on the sphere*** with n vertices is shown in Figure 7.8.10, and is denoted by Δ_{n-3}. The skeleton is isomorphic to $P_{n-2} + K_2$.

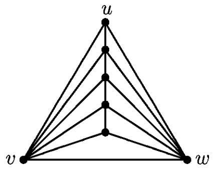


Figure 7.8.10 The standard triangulation Δ_4 on the sphere.

NOTATION: The result of adding Δ_m to an arbitrarily chosen face of a triangulation $G \to S$ on a closed surface is denoted by $G + \Delta_m \to S$. (All possible triangulations $G + \Delta_m \to S$ are equivalent under diagonal flips. See [Ne94].)

FACTS

F44: [Wa36] Any two triangulations on the sphere with the same number of vertices are equivalent under diagonal flips, up to isotopy.

F45: [NeWa90] Any two triangulations on the projective plane with the same number of vertices are equivalent under diagonal flips, up to isotopy.

F46: [De73] Any two triangulations on the torus with the same number of vertices are equivalent under diagonal flips, up to isomorphism.

F47: [NeWa90] Any two triangulations on the Klein bottle with the same number of vertices are equivalent under diagonal flips, up to isomorphism.

F48: [Ne94] For every closed surface S, there exists a natural number $N(S)$ such that two triangulations G_1 and G_2 with $|V(G_1)| = |V(G_2)| \geq N(S)$ are equivalent under diagonal flips, up to isomorphism.

F49: [NaOt97] For every closed surface S, there exists a natural number $\tilde{N}(S)$ such that two triangulations G_1 and G_2 with $|V(G_1)| = |V(G_2)| \geq \tilde{N}(S)$ are equivalent under diagonal flips, up to isotopy.

F50: [Ne01] Let G_1 and G_2 be two triangulations on a closed surface S with the same number n of vertices, and let $m \geq 18(n-\chi(S))$. Then any two subdivided triangulations of the form $G_1 + \Delta_m$ and $G_2 + \Delta_m$ are equivalent under diagonal flips, up to isotopy.

F51: [Ne94] A frozen triangulation is irreducible.

EXAMPLES

E16: For the sphere S_0, the projective plane N_1, the torus S_1 and the Klein bottle N_2, we have the following numbers:

$$N(S_0) = 4, \quad N(N_1) = 6, \quad N(S_1) = 7, \quad N(N_2) = 8$$

E17: The irreducible triangulations on the torus can be partitioned into five equivalence classes, as follows, under sequences of diagonal flips:

$$\{\mathrm{T1}\}, \quad \{\mathrm{T2}, \mathrm{T3}, \mathrm{T4}, \mathrm{T5}\}, \quad \{\mathrm{T7}\}, \quad \{\mathrm{T6}, \mathrm{T8}, \ldots, \mathrm{T20}\}, \quad \{\mathrm{T21}\}$$

E18: If K_n or $K_{n(m)}$ triangulates a closed surface, then it is a frozen triangulation.

REMARKS

R8: The problem of equivalence would be nearly trivial, and the lower bound $N(S)$ for the order of triangulations would be meaningless, if we allowed diagonal flips that resulted in non-simple skeletons. If the requirement of simpleness is removed, then there is a greedy algorithm to transform one of two triangulations into the other. See [Ne01] for details.

R9: Within the theory of diagonal flips in topological graph theory, the positions of vertices may be moved on surfaces, up to homeomorphism or isotopy. However, in computational geometry, there are studies of diagonal flips in triangulations, in which the vertices have fixed positions in the plane.

R10: The bound $N(S)$ in Fact 48 is actually necessary, because there exist frozen triangulations on infinitely many surfaces, as discussed in [Ne99b]. The arguments needed to prove the theorem also work for labeled triangulations and for triangulations with boundary, with suitable modifications. See [Ne99a].

R11: The bound $\tilde{N}(S)$ in Fact 49 is large and unknown, even when S is the torus or the Klein bottle.

Estimating Bounds

We consider how many diagonal flips are necessary to transform one triangulation into another.

DEFINITION

D33: A ***pseudo-minimal triangulation*** is a triangulation such that no sequence of diagonal flips transforms it into one having a vertex of degree 3.

FACTS

F52: [Ne94] A pseudo-minimal triangulation is irreducible.

F53: [Ne94] Let $\{T_i\}$ be the set of the pseudo-minimal triangulations of a closed surface S. The precise value of $N(S)$ is equal to the minimum number N such that all the subdivisions $T_i + \Delta_{N-|V(T_i)|}$ can be transformed into one another by diagonal flips, up to isomorphism.

F54: [Ne01] Let $V_{\rm pse}(S)$ denote the maximum order taken over all the pseudo-minimal triangulations of a closed surface S with Euler characteristic $\chi(S)$. Then we have:

$$\tilde{N}(S) \leq 19\, V_{\rm pse}(S) - 18\, \chi(S)$$

F55: [Ne98] Given a closed surface S, there are two constants α_1 and α_0, depending only on S, such that any two triangulations $G_1 \to S$ and $G_2 \to S$ with $n \geq N(S)$ vertices can be transformed into each other by at most $2n^2 + \alpha_1 n + \alpha_0$ diagonal flips, up to isomorphism.

F56: [MoNaOt03] Any two triangulations with n vertices on the sphere can be transformed into each other, up to isotopy, by at most $6n - 30$ diagonal flips if $n \geq 5$.

F57: [MoNa03] Any two triangulations with n vertices on the projective plane can be transformed into each other, up to isotopy, by at most $8n - 26$ diagonal flips.

F58: [GaUrWa01] Any two labeled triangulations with n vertices on the sphere can be transformed into each other, up to isotopy, by $O(n \log n)$ diagonal flips.

EXAMPLE

E19: All pseudo-minimal triangulations on the sphere, the projective plane, the torus and the Klein bottle are minimal triangulations on these surfaces.

REMARK

R12: Since there is a linear upper bound for the order of irreducible triangulations with respect to the genus of S, the upper bound for $\tilde{N}(S)$ given in Fact 54 also is linear.

Catalan triangulations

There are some studies on diagonal flips in Catalan triangulations with the same framework as above, although the lack of interior vertices is an obstacle to the general arguments in [Ne94]. Furthermore, there has been shown an amazing method for Catalan triangulations of polygons in [SlTaTh88], in harmony with combinatorics, hyperbolic geometry and computer science.

DEFINITION

D34: A ***punctured surface*** is a closed surface with one hole, that is, a surface with connected boundary.

FACTS

F59: [SlTaTh88] Any two Catalan triangulations of an n-gonal disk with $n \geq 13$ can be transformed into each other by at most $2n - 10$ diagonal flips. There exists an example attaining this bound.

F60: [EdRe97] Any two Catalan triangulations of the Möbius band with the same number of vertices are equivalent under diagonal flips, up to isomorphism.

F61: [CoNa00a] Any two Catalan triangulations of the punctured torus with the same number of vertices are equivalent under diagonal flips, up to isomorphism.

F62: [CoNa00b] Any two Catalan triangulations of the punctured Klein bottle with the same number of vertices are equivalent under diagonal flips, up to isomorphism.

F63: [CoGlMaNa02] Given a puctured surface S, there exists a natural number $M(S)$ such that two Catalan triangulations G_1 and G_2 of S are equivalent under diagonal flips, up to isomorphism, if $|V(G_1)| = |V(G_2)| \geq M(S)$.

Preserving Properties

Any diagonal flip preserves the order of triangulations while an edge contraction decreases it by one. Nevertheless, the former is closely related to the latter, as Facts 51 and 52 suggest. This makes a connection of the theory of diagonal flips to graph minor theory and leads us to more general or formal arguments on conditional generating of triangulations.

DEFINITIONS

D35: A class $\mathcal{P}$ of triangulations on S is said to be ***splitting-closed*** if it is closed under vertex splittings.

D36: Let $\mathcal{P}$ be a class of triangulations. A triangulation is called a ***$\mathcal{P}$-triangulation*** (or a triangulation with property $\mathcal{P}$) if it belongs to $\mathcal{P}$.

D37: A ***$\mathcal{P}$-diagonal flip*** in a $\mathcal{P}$-triangulation G is a diagonal flip such that the resulting graph is also a $\mathcal{P}$-triangulation.

D38: Two $\mathcal{P}$-triangulations G_1 and G_2 are said to be ***$\mathcal{P}$-equivalent*** under diagonal flips if they can be transformed into each other by a finite sequence of $\mathcal{P}$-diagonal flips.

D39: A class $\mathcal{P}$ of triangulations on a closed surface S is said to be ***closed under homeomorphism*** if $h(G) \in \mathcal{P}$ for any member $G \in \mathcal{P}$ and for any homeomorphism $h : S \to S$.

FACTS

F64: [BrNaNe96] For any closed surface S and for any splitting-closed class $\mathcal{P}$ of triangulations on S, there exists a natural number $N_{\mathcal{P}}(S)$ such that if G_1 and G_2 are two $\mathcal{P}$-triangulations with $|V(G_1)| = |V(G_2)| \geq N_{\mathcal{P}}(S)$, then G_1 and G_2 are $\mathcal{P}$-equivalent under diagonal flips, up to isomorphism.

F65: [BrNaNe96] For any closed surface S and for any splitting-closed class $\mathcal{P}$ of triangulations on S which is closed under homeomorphism, there exists a natural number $\tilde{N}_{\mathcal{P}}(S)$ such that if G_1 and G_2 are two $\mathcal{P}$-triangulations with $|V(G_1)| = |V(G_2)| \geq \tilde{N}_{\mathcal{P}}(S)$, then G_1 and G_2 are $\mathcal{P}$-equivalent under diagonal flips, up to isotopy.

F66: [KoNaNe99] For any closed surface S except the sphere, there exists a natural number $N_4(S)$ such that two triangulations G_1 and G_2 on S with minimum degree at least 4 can be transformed into each other by a finite sequence of diagonal flips, up to isomorphism, through those triangulations if $|V(G_1)| = |V(G_2)| \geq N_4(S)$.

F67: [KoNaNe99] Two triangulations on the sphere, except the double wheels, with minimum degree at least 4 can be transformed into each other, up to isotopy, by a finite sequence of diagonal flips through those triangulations if they have the same number of vertices.

EXAMPLES

The following properties are splitting-closed and closed under homeomorphism.

E20: Being k-representative.

E21: Intersecting any non-separating simple closed curve in at least k points.

E22: Containing at least k disjoint homotopic cycles.

E23: Containing at least k disjoint cycles.

E24: Containing k distinct spanning trees.

REMARK

R13: The class consisting of triangulations on a closed surface with minimum degree at least 4 is not splitting-closed, and hence the meta-theorems in [BrNaNe96] cannot be used to prove the theorems in [KoNaNe99].

7.8.5 Rigidity and Flexibility

A triangulation may seem quite rigid. So one might guess that it is hardly possible for a graph that triangulates a closed surface to have another imbedding on that same surface, which is actually true for the sphere. However, the complete graph triangulates a closed surface in numerous ways. Here we shall consider many facts on the rigidity and flexibility of triangulations.

Equivalence over Imbeddings

To analyze many imbeddings of a graph, it is often useful to deal with an imbedding as a map rather than a drawing on a surface. That is, an imbedding of a graph G into a surface S is an injective continuous map $f : G \to S$ from a 1-dimensional topological space G to S.

DEFINITONS

D40: Two imbeddings $f_1, f_2 : G \to S$ of a graph into a surface are ***equivalent*** if there exists a homeomorphism $h : S \to S$ with $hf_1 = f_2$.

D41: Two imbeddings $f_1, f_2 : G \to S$ of a graph into a surface are ***congruent*** if there exists a homeomorphism $h : S \to S$ and a graph automorphism $\sigma : G \to G$ with $hf_1 = f_2\sigma$.

D42: An automorphism $\sigma \in \text{Aut}(G)$ is called a ***symmetry*** of an imbedding $f : G \to S$ if there is a homeomorphism $h : S \to S$ with $hf = f\sigma$.

D43: The ***symmetry group*** $\text{Sym}(f)$ ***of an imbedding*** $f : G \to S$ is the subgroup in $\text{Aut}(G)$ consisting of the symmetries of the imbedding.

FACTS

F68: If a simple graph has a triangular imbedding on a closed surface, then all of its imbeddings on that surface are triangular.

F69: Equivalent triangular imbeddings of a graph G have the same set of face boundary cycles over $V(G)$.

F70: Congruent triangular imbeddings of a graph G correspond to isomorphic triangulations.

F71: An imbedding $f : G \to S$ is equivalent to $f\sigma$ for any symmetry $\sigma \in \text{Sym}(f)$.

F72: An imbedding $f : G \to S$ is congruent but is not equivalent to $f\bar{\sigma}$ for any automorphism $\bar{\sigma} \in \text{Aut}(G) - \text{Sym}(f)$.

F73: The number of inequivalent imbeddings of a graph G congruent to a fixed imbedding $f : G \to S$ is equal to $|\text{Aut}(G)|/|\text{Sym}(f)|$.

Uniqueness of Imbeddings

It has been known that the skeleton of an imbedding of sufficiently large representativity is rigid, that is, it has a unique imbedding on the surface that contains it. However, relatively simple conditions force any skeleton of a triangulation to be rigid.

DEFINITIONS

D44: A graph is said to be ***uniquely imbeddable*** on a surface S if all of its imbeddings into S are equivalent.

D45: A ***skew vertex*** in a triangulation G on a closed surface S is a vertex v such that there are at least two cycles each of which contains all the neighbors of v.

FACTS

F74: The skeleton of every triangulation on the sphere is uniquely imbeddable on the sphere, up to equivalence. This is an easy consequence of the well-known fact that every 3-connected planar graph is uniquely imbeddable on the sphere.

F75: [NeNaTa97] A graph that triangulates a closed surface is uniquely imbeddable on that surface, up to equivalence, if any face has at most two skew vertices as its corners.

F76: [Ne83] The skeleton of a 4-representative triangulation on a closed surface is uniquely imbeddable on that surface, up to equivalence.

F77: [Ne83] The skeleton of a 6-connected toroidal triangulation is uniquely imbeddable on the torus, up to congruence, and also up to equivalence, with three exceptions.

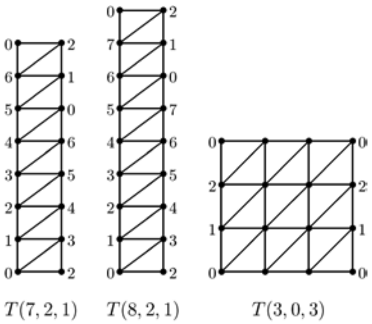


Figure 7.8.11 **The three exceptional 6-connected toroidal triangulations.**

F78: [Ne84a] The skeleton of a 5-connected projective-planar triangulation is uniquely imbeddable on the projective plane up to equivalence, unless it is isomorphic to K_6.

F79: [Ne84b] The skeleton of a 6-connected Klein-bottle triangulation is uniquely imbeddable on the Klein bottle up to congruence, and likewise up to equivalence, with one exception.

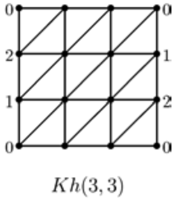


Figure 7.8.12 **The exceptional 6-connected Klein-bottle triangulation.**

Re-Imbedding Structures

It is hardly possible to classify all the mechanisms that generate inequivalent imbeddings of a graph. However, there is a theory to describe the flexibility of triangulations.

DEFINITIONS

D46: A face of a triangulation $G \to S$ is a ***panel*** if its boundary cycle bounds a face in every imbedding of G in S.

D47: The ***panel structure*** of an imbedding $G \to S$ is a pair $(G \to S, \mathcal{P})$ in which $\mathcal{P}$ is the set of all the panels.

D48: The panel structures of two triangulations $G_1 \to S$ and $G_2 \to S$ are said to be ***equivalent*** if the 2-simplicial complexes obtained from the skeletons G_1 and G_2 by inserting all of their panels are homeomorphic.

FACTS

F80: [LaNe99] Two faces incident to a contractible edge of a triangulation are panels.

F81: [NeNaTa97] A face that has at most two skew vertices at its corners is a panel.

F82: [NeNaTa97] Two triangulations on a closed surface having equivalent panel structures admit the same number of inequivalent imbeddings on the surface.

F83: The number of inequivalent imbeddings of a triangulation $G \to S$ (with S closed) does not exceed that of an irreducible triangulation to which $G \to S$ is contractible.

F84: [NeNaTa97] There exist only finitely many panel structures on each closed surface, up to equivalence.

F85: [La92] Every projective-planar triangulation admits exactly 1, 2, 3, 4, 6 or 12 inequivalent imbeddings on the projective plane.

F86: [Sa03] Every toroidal triangulation admits exactly 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 24, 48 or 120 inequivalent imbeddings on the torus.

EXAMPLE

E25: Let $Q(S)$ denote the maximum number of inequivalent imbeddings taken over all graphs that triangulate a given closed surface S. For the sphere S_0, the projective plane N_1, the torus S_1 and the Klein bottle N_2, we have:

$$Q(S_0) = 1, \quad Q(N_1) = 12, \quad Q(S_1) = 120, \quad Q(N_2) = 36$$

The first three are attained by K_4, K_6, and K_7 on these surfaces in order while the last one is attained by the triangulation obtained from two copies of K_6 on the projective plane by pasting them along one pair of faces.

REMARK

R14: To determine the maximum number of inequivalent imbeddings taken over all triangulations on a closed surface, it suffices to investigate irreducible triangulations, by Fact 83. On the other hand, the classification of panel structures exhibits all "reimbedding structures" and enables us to decide all possible values that appear as the number of inequivalent imbeddings of triangulations. See [NeNaTa97] for the theory of panel structures.

Imbeddings into Other Surfaces

What happens when we imbed the skeleton of a triangulation into other surfaces?

DEFINITION

D49: A graph G is said to ***quadrangulate*** a surface S if G can be imbedded on S so that each face is a 4-cycle.

FACTS

F87: [HoGl77] The skeleton of a triangulation on an orientable closed surface is an upper imbeddable graph. That is, it can be cellularly imbedded on a suitable orientable closed surface with one or two faces.

F88: [LaNe99] A graph triangulates both the torus and the Klein bottle if and only if it has the structure as shown in Figure 7.8.13, where each triangle with ○ may be divided into many triangles.

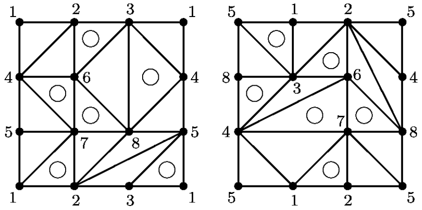


Figure 7.8.13 **Triangulating both the torus and the Klein bottle.**

F89: [NaNeOtSi03] For any closed surface S, there is a triangulation of the sphere whose skeleton quadrangulates the surface S.

F90: [NaNeOtSi03] No 5-connected graph triangulates the sphere and quadrangulates another orientable closed surface.

F91: [NeSu00] There is a 5-connected graph that triangulates the sphere which quadrangulates the nonorientable closed surface N_k if and only if $k = 10$ or $k \geq 12$. Such a triangulation for $k = 10$ is unique up to isomorphism.

F92: [Su03] If two closed surfaces S_1 and S_2 satisfy the relation $2\chi(S_1) - \chi(S_2) \geq 4$, then there is a graph that triangulates S_1 and quadrangulates S_2.

EXAMPLES

E26: The octahedron $K_{2,2,2}$ triangulates the sphere and quadrangulates the torus.

E27: The complete graph K_4 triangulates the sphere and quadrangulates the projective plane.

E28: The only irreducible triangulation on the torus whose skeleton also triangulates the Klein bottle is T3, and the triangulation on the Klein bottle is isomorphic to Kh1.

E29: [HaRi89] The complete graph K_n triangulates and quadrangulates two different closed surfaces if and only if $n \equiv 0, 1, 4$ or $9 \pmod{12}$. This is exactly what Euler's formula requires.

References

[Ar92] D. Archdeacon, The medial graph and voltage-current duality, *Discrete Math.* 104 (1992) 111–141.

[ArBrNe95] J. L. Arocha, J. Bracho, and V. Neumann-Lara, Tight and untight triangulated surfaces, *J. Combin. Theory, Ser. B* 63 (1995), 185–199.

[Ba82] D. Barnette, Generating the triangulations of the projective plane, *J. Combin. Theory, Ser. B* 33 (1982), 222–230.

[BaEd89] D. W. Barnette and A. L. Edelson, All 2-manifolds have finitely many minimal triangulations, *Isr. J. Math.* 67 (1989), 123–128.

[BoGrGrSi00] C. P. Bonnington, M. J. Grannell, T. S. Griggs, and J. Širáň, Exponential families of non-isomorphic triangulations of complete graphs, *J. Combin. Theory, Ser. B* 78 (2000), 169–184.

[Bo82a] A. Bouchet, Constructing a covering triangulation by means of a nowhere-zero flow, *J. Combin. Theory, Ser. B* 32 (1982), 316–325.

[Bo82b] A. Bouchet, Constructions of covering triangulations with folds, *J. Graph Theory* 6 (1982), 57–74.

[BrSt01] J. Bracho and R. Strausz, Nonisomophic complete triangulations, *Discrete Math.* 232 (2001), 11–18.

[BrNaNe96] R. Brunet, A. Nakamoto and S. Negami, Diagonal flips of triangulations on closed surfaces preserving specified properties, *J. Combin. Theory, Ser. B* 68 (1996), 295–309.

[CoNa00a] C. Cortés and A. Nakamoto, Diagonal flips in outer-torus triangulations, *Discrete Math.* 216 (2000), 71–83.

[CoNa00b] C. Cortés and A. Nakamoto, Diagonal flips in outer-Klein-bottle triangulations, *Discrete Math.* 222 (2000), 41–50.

[CoGlMaNa02] C. Cortés, C. Glima, A. Márquez, and A. Nakamoto, Diagonal flips in outer-triangulations of surfaces, *Discrete Math.* 254 (2002), 63–74.

[De73] A. K. Dewdney, Wagner's theorem for the torus graphs, *Discrete Math.* 4 (1973), 139–149.

[EdRe97] P. H. Edelman and V. Reiner, Catalan triangulations of Möbius band, *Graphs Combin.* 13 (1997), 231–243.

[FiNoNe94] S. Fisk, B. Moha,r and R. Nedela, Minimal locally cyclic triangulations of the projective plane, *J. Graph Theory* 18 (1994), 25–35.

[Fr34] P. Franklin, A six color problem, *J. Math. Phys.* 16 (1934), 363–369.

[GaRiSe96] Z. Gao, R. B. Richter and P. Seymour, Irreducible triangulations of surfaces, *J. Combin. Theory Ser. B* 68 (1996), 206–217.

[GaUrWa01] Z. Gao, J. Urrutia and J. Wang, Diagonal flips in labelled planar triangulations, *Graphs Combin.* 17 (2001), 647–657.

[Gr74] J. L. Gross, Voltage graphs, *Discrete Math.* 9 (1974), 239–246.

[HaRi89] N. Hartsfield and G. Ringel, Minimal quadrangulations of nonorientable surfaces, *J. Combin. Theory, Ser. A* 50 (1989), 185–195.

[HaRi91] N. Hartsfield and G. Ringel, Clean triangulations, *Combinatorica* 11 (1991), 145–155.

[HoGl77] N. P. Homenko and A. D. Gluhov, Upper imbeddable graphs (Russian), *Graph Theory* (Russian), pp.85–89, Inst. Math., Akad. Nauk Ukrain. SSR, Kiev, 1977.

[JuRi80] M. Jungerman and G. Ringel, Minimal triangulations on orientable surfaces, *Acta Math.* 145 (1980), 121–154.

[KoNaNe99] H. Komuro, A. Nakamoto and S. Negami, Diagonal flips in triangulations on closed surfaces with minimum degree at least 4, *J. Combin. Theory, Ser. B* 76 (1999), 68–92.

[La87] S. Lawrencenko, The irreducible triangulations of the torus, *Ukrain. Geom. Sb.* 30 (1987), 52–62. [In Russian; MR 89c:57002; English translation: *J. Soviet Math.* 51, No.5 (1990), 2537–2543.]

[La92] S. Lawrencenko, The variety of triangular embeddings of a graph in the projective plane, *J. Combin. Theory, Ser. B* 54 (1992), 196–208.

[LaNe97] S. Lawrencenko and S. Negami, Irreducible triangulations of the Klein bottle, *J. Combin. Theory, Ser. B* 70 (1997), 265–291.

[LaNe99] S. Lawrencenko and S. Negami, Constructing the graphs that triangulate both the torus and the Klein bottle, *J. Combin. Theory, Ser. B* 77 (1999), 211–218.

[LaNeWh94] S. Lawrencenko, S. Negami, and A. T. White, Three nonisomorphic triangulations of an orientable surface with the same complete graph, *Discrete Math.* 135 (1994), 367–369.

[MaMo92] A. Malnič and B. Mohar, Generating locally cyclic triangulations of surfaces, *J. Combin. Theory Ser. B* 56 (1992), 147–164.

[MaNe95] A. Malnic and R. Nedela, k-minimal triangulations of surfaces, *Acta Math. Univ. Comenian* (N.S.) 64 (1995), 57–76.

[MoNaOt03] R. Mori, A. Nakamoto and K. Ota, Diagonal flips in Hamiltonian triangulations on the sphere, *Graphs Combin.* in press.

[MoNa03] R. Mori and A. Nakamoto, Diagonal flips in Hamiltonian triangulations on the projective plane, preprint.

[NaOt95] A. Nakamoto and K. Ota, Note on irreducible triangulations of surfaces, *J. Graph Theory* 20 (1995), 227–233.

[NaOt97] A. Nakamoto and K. Ota, Diagonal transformations of graphs and Dehn twists of surfaces, *J. Combin. Theory, Ser. B* 70 (1997), 292–300.

[NaNe02] A. Nakamoto and S. Negami, Generating triangulations on closed surfaces with minimum degree at least 4, *Discrete Math.* 244 (2002), 345–349.

[NaNeOtSi03] A. Nakamoto, S. Negami, K. Ota and J. Širáň, Planar triangulations which quadrangulate other surfaces, preprint.

[Ne83] S. Negami, Uniqueness and faithfulness of embedding of toroidal graphs, *Discrete Math.* 44 (1983), 161–180.

[Ne84a] S. Negami, Uniquely and faithfully embeddable projective-planar triangulations, *J. Combin. Theory, Ser. B* 36 (1984), 189–193.

[Ne84b] S. Negami, Classification of 6-regular Klein-bottlal graphs, *Res. Rep. Inf. Sci. T.I.T.* A-96 (1984).

[NeWa90] S. Negami and S. Watanabe, Diagonal transformations of triangulations on surfaces, *Tsukuba J. Math.* 14 (1990), 155–166.

[Ne94] S. Negami, Diagonal flips in triangulations of surfaces, *Discrete Math.* 135 (1994), 225–232.

[NeMi96] S. Negami and T. Midorikawa, Loosely-tightness of triangulations of closed surfaces, *Sci. Rep. Yokohama Nat. Univ., Sec. I* 43 (1996), 25–41.

[NeNaTa97] S. Negami, A. Nakamoto and T. Tanuma, Re-embedding structures of triangulations on closed surfaces, *Sci. Rep. Yokohama Nat. Univ., Sec. I* 44 (1997), 41–55.

[Ne98] S. Negami, Diagonal flips in triangulations on closed surfaces, estimating upper bounds, *Yokohama Math. J.* 45 (1998), 113–124.

[Ne99a] S. Negami, Diagonal flips of triangulations on surfaces, a survey, *Yokohama Math. J.* 47, special issue (1999), 1–40.

[Ne99b] S. Negami, Note on frozen triangulations on closed surfaces, *Yokohama Math. J.* 47, special issue (1999), 191–202.

[Ne01] S. Negami, Diagonal flips in pseudo-triangulations on closed surfaces, *Discrete Math.* 240 (2001), 187–196.

[NeSu00] S. Negami and Y. Suzuki, 5-Connected planar triangulations quadrangulating other surfaces, *Yokohama Math. J.* 47 (2000), 187–193.

[Ri55] G. Ringel, Wie man die geschlossenen nichtorientierbaren Flächen in möglichst wenig Dreiecke zerlengen kann, *Math. Annalen* 130 (1955), 317–326.

[Ri74] G. Ringel, “Map Color Theorem”, Springer Verlag, 1974.

[Sa03] A. Sasao, Panel structures of triangulations on the torus, preprint.

[SlTaTh88] D. Sleator, R. Tarjan and W. Thurston, Rotation distance, triangulations, and hyperbolic geometry, *J. Amer. Math. Soc.* 1 (1988), 647–681.

[StRa34] E. Steinitz and H. Rademaher, “Vorlesungen über die Theorie der Polyeder”, Springer, Berlin, 1934.

[Su03] Y. Suzuki, Triangulations on closed surfaces which quadrangulate other surfaces, preprint.

[Wa36] K. Wagner, Bemekungen zum Vierfarbenproblem, *J. der Deut. Math.*, Ver. 46, Abt. 1, (1936), 26–32.

7.9 GRAPHS AND FINITE GEOMETRIES

Arthur T. White, Western Michigan University

7.9.1 Finite Geometries
7.9.2 Associated Graphs
7.9.3 Surface Models
References

Introduction

To every finite geometry, there are associated two distinct graphs, the Levi graph and the Menger graph. Imbeddings of these graphs into closed 2-manifolds can lead to models of the geometries. The emphasis is on imbeddings that are optimal with respect to euler characteristic, or that utilize important collineations of the geometry as map automorphisms.

The graphs in this section are simple.

7.9.1 Finite Geometries

A finite geometry might be of intrinsic interest; it might produce a useful block design; or, it might have an aesthetically pleasing model.

DEFINITIONS

D1: A ***geometry*** (P, L) consists of a non-empty set P called ***points***, together with a non-empty collection L of subsets of P called ***lines***.

D2: A ***finite geometry*** is a geometry (P, L) whose point-set P is finite.

D3: A (v, b, r, k, λ)-***balanced incomplete block design*** (abbr. ***BIBD***) is a finite geometry (P, L) with $v = |P|$ and $b = |L|$ that satisfies the following three ***axioms of uniformity***:

- Every point is in exactly r lines.
- Every line consists of exactly k points.
- Every pair of points belong to exactly λ common lines.

D4: The ***axiom of uniqueness*** for a (v, b, r, k, λ)-BIBD is $\lambda = 1$.

D5: A ***Steiner triple system*** is a (v, b, r, k, λ)-BIBD with $k = 3$ and $\lambda = 1$.

D6: A $(v, b, r, k; \lambda_1, \lambda_2)$-***partially balanced incomplete block design*** is just like a BIBD, except that the points can be regarded as the vertices of a fixed strongly regular graph, in which two points are non-adjacent (or adjacent) if they belong to exactly λ_1 (or λ_2) common lines. (abbr. ***PBIBD***)

D7: An (r, k)-***configuration*** is a finite geometry (P, L) that satisfies the first two axioms of uniformity, but replaces the third axiom with the following:

- Every pair of points belong to at most one common line.

D8: A ***3-configuration*** is an (r, k)-configuration such that $k = 3$.

D9: A $(v, b, r, k; 0, 1)$-***block design*** (abbr. ***BD***) is an (r, k)-configuration (P, L), where $v = |P|$ and $b = |L|$, such that no or some pairs of distinct points are not collinear $(\lambda_1 = 0)$, and the other pairs of distinct points are uniquely collinear $(\lambda_2 = 1)$.

D10: A ***symmetric configuration*** $(v)_k$ is a $(v, b, r, k; 0, 1)$ block design such that $r = k$, which implies $v = b$, by Fact 1 below.

D11: For each natural number $n > 1$, the n-***point geometry*** has n points, and all the 2-subsets of those points as its lines.

D12: A ***finite affine plane of order*** n is a finite geometry (P, L) that satisfies these axioms:

- Two distinct points are in a unique common line.
- For a given point not in a given line, there is a unique parallel (non-intersecting) line containing that point.
- There exist four distinct points, no three collinear.
- There exists a line having exactly n points.

D13: A ***finite projective plane of order*** n (abbr. $\Pi(n)$) is a finite geometry (P, L) that satisfies these axioms:

- Two distinct points are in a unique common line.
- Two distinct lines contain a unique common point.
- There exist four distinct points, no three collinear.
- There exists a line having exactly n + 1 points.

FACTS

F1: In any (v, b, r, k, λ)-BIBD, $(v, b, r, k; 0, 1)$-PBIBD or $(v, b, r, k; 0, 1)$-BD, the axioms of uniformity imply that $vr = bk$.

F2: A 3-configuration (P, L) exists if and only if $vr = 3b$ and $v \geq 2r + 1$. Thus, a symmetric configuration $(v)_3$ exists if and only if $v \geq 7$.

F3: For a prime power $n \geq 9$ that is not prime, there are at least two non-isomorphic projective planes of order n. There are exactly four, for $n = 9$.

F4: For $n = 2, 3, 4, 5, 7,$ and 8, $\Pi(n)$ is uniquely $PG(2, n)$.

F5: Neither a $\Pi(6)$ nor a $\Pi(10)$ exists.

F6: There is no known finite projective plane $\Pi(n)$ where n is not a prime power.

F7: [BrRy49] If $n \equiv 1, 2 (\text{mod } 4)$, and if n is not a sum of two squares, then no projective plane $\Pi(n)$ exists.

F8: The affine plane $AG(2, q)$ is a resolvable $(q^2, q^2 + q, q + 1, q, 1)$-BIBD. Such planes exist for every prime power q. The resolvable feature is that the $q^2 + q$ lines partition into $q + 1$ parallel classes of q lines each, each class partitioning the point set.

F9: Every triangle in the Euclidean plane has the following four triples of concurrent lines:

- The perpendicular bisectors of the three sides meet in the ***circumcenter***.
- The altitudes meet in the ***orthocenter***.
- The internal angle bisectors meet in the ***incenter***.
- The medians meet in the ***centroid***.

If the triangle is equilateral, then all four points of concurrency coincide. This fact is used as background for Example 10.

EXAMPLES

E1: The Euclidean plane is an (infinite) affine plane.

E2: The n-point geometry is an $\left(n, \frac{n(n-1)}{2}, n-1, 2, 1\right)$-BIBD.

E3: An r-regular graph is representable as an $(r, 2)$-configuration.

E4: An r-regular, k-uniform hypergraph is representable as an (r, k)-configuration.

E5: The Theorem of Pappus in Euclidean geometry states that if A, B, C are distinct points on line L and A', B', C' are three different distinct points on line $L' \neq L$, then the three points $D = AB' \cap A'B, E = AC' \cap A'C, F = BC' \cap B'C$ are collinear. This gives a 3-configuration on the nine points $\{A, B, C, A', B', C', D, E, F\}$ called the ***geometry of Pappus***, which is a $(9, 9, 3, 3; 0, 1)$-PBIBD. (See [Wh01], for example, for a diagram of the geometry of Pappus.)

E6: The Theorem of Desargues in Euclidean geometry states that if triangles ABC and $A'B'C'$ are in perspective from point P, then the three points $D = AB \cap A'B', E = AC \cap A'C', F = BC \cap B'C'$ are collinear. This gives a 3-configuration on the ten points $\{P, A, B, C, A', B', C', D, E, F\}$ called the ***geometry of Desargues***, which is a $(10, 10, 3, 3; 0, 1)$-PBIBD. (See [Wh01], for example, for a diagram of the geometry of Desargues.)

E7: For each prime power q, there is a classical finite affine plane $AG(2, q)$, which is a $(q^2, q^2 + q, q + 1, q, 1)$-BIBD with $\lambda = 1$. The projective plane $AG(2, q)$ has as its points the 1-dimensional affine subspaces of the 3-dimensional vector space over $GF(q)$, and as its lines the 2-dimensional affine subspaces. In particular, $AG(2, 2)$ is the 4-point geometry.

E8: For each prime power q, there is a classical finite projective plane $\Pi(q) = PG(2, q)$, which is a $(q^2 + q + 1, q^2 + q + 1, q + 1, q + 1, 1)$-BIBD with $\lambda = 1$. The projective plane $PG(2, q)$ has as its points the 1-dimensional vector subspaces of the 3-dimensional vector space over $GF(q)$, and as its lines the 2-dimensional vector subspaces.

E9: To every projective plane Π there corresponds an affine plane Π' (obtained by deleting one edge and all points on that edge from Π), and conversely. The affine plane obtained from $PG(2, q)$ is $AG(2, q)$.

E10: The ***Fano plane***, a familiar 3-configuration that models $PG(2,2)$, is shown in Figure 7.9.1. It has seven points. The three sides, the three medians, and the incircle form its seven lines. This $(7,7,3,3,1)$-BIBD is the smallest non-trivial Steiner triple system.

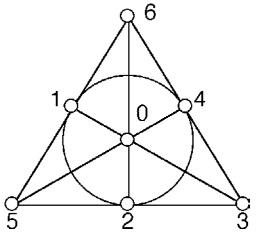


Figure 7.9.1 **The Fano plane.**

The line $\{0,1,3\}$ is called a ***perfect difference set***, since each non-identity element of Z_7 appears uniquely as a difference of two elements in the set. It generates the other six lines, using translations by Z_7. But note that this model has several defects (to be remedied in §7.9.3):

- The circular line $\{1,2,4\}$ is differently depicted. Yet Z_7 acts transitively on the line set L, as a subgroup of the full collineation group (of order 168).
- The point 0 is distinguished by its central position, yet Z_7 acts transitively on the point set P as well.
- The point 2 (for example) seems to be between points 3 and 5. Yet there is no concept of betweenness in this geometry.
- One cannot discern that $r = 3$, by looking at small neighborhoods of points $0,1,2$, and 4.
- There are three extraneous intersections of lines (i.e., of the three cevians with the circle) that have no meaning in the geometry.

E11: The projective plane $PG(2,4)$ can be cyclically generated, using Z_{21}, from the perfect difference set $\{0,1,6,8,18\} = L_0$. Then line $L_i = L_0 + i$, for $0 \le i \le 20$. Let $L'_0 = \{0,6,18\}$, with $L'_k = L'_0 + 3k$, for $0 \le k \le 6$. Then using the Z_7 subgroup of Z_{21}, we find the Fano plane $PG(2,2)$ contained within $PG(2,4)$.

E12: The Theorem of Desargues applies to the full projective plane $PG(2,4)$; for example, the triangles 3 5 7 and 10 13 9 are in perspective from point 2, producing the three collinear points $19, 6, 20$ and the geometry of Desargues, contained within $PG(2,4)$. Next, deleting line L_0 and its five points from $PG(2,4)$ yields $AG(2,4)$, with its five parallel classes, as shown below.

2	**7**	**9**	**19**	2	**3**	**10**	**20**	**3**	4	**9**	**11**	**5**	**11**	**13**	2	**10**	**11**	16	**7**
4	5	10	12	7	13	15	4	**13**	14	**19**	**10**	7	12	14	3	12	**13**	**20**	**9**
14	15	20	11	9	14	16	5	15	16	2	12	9	10	15	17	17	2	4	14
16	17	3	13	11	12	17	19	**20**	**5**	**7**	17	19	20	4	16	**19**	**3**	**5**	15

Notice that the bold (sub)lines give the three parallel classes of a Pappus configuration within $AG(2,4)$ and hence within $PG(2,4)$.

7.9.2 Associated Graphs

DEFINITIONS

D14: The ***Menger graph*** of a geometry (P, L) has the point set P as its vertex set and all edges of the form $\{p_1, p_2\}$, for $p_1, p_2 \in P$, where p_1 and p_2 are collinear; i.e., $p_1, p_2 \in l$, for some $l \in L$.

D15: The ***Levi graph*** of a geometry (P, L) is the bipartite graph having $P \cup L$ as its vertex set and all edges of the form $\{p, l\}$, for $p \in P, l \in L$, and $p \in l$. (Similarly, Levi graphs can encode objects and blocks of a block design, or vertices and hyperedges of a hypergraph.)

D16: A (d, g)-***cage*** is a graph of minimum order among all d-regular graphs having girth g.

FACTS

F10: In general, every BIBD of order v has K_v as its Menger graph.

F11: A Levi graph represents its geometry uniquely, whereas two or more different geometries might have the same Menger graph.

F12: The Levi graph of an (r, k)-configuration has girth $g \geq 6$. The Levi graph of a symmetric configuration $(v)_k$ is k-regular.

F13: [Si66] There is an $(n+1, 6)$-cage of order $2(n^2+n+1)$ if and only if there exists a finite projective plane $\Pi(n)$.

F14: The Levi graph of $PG(2, n)$ is an $(n+1, 6)$-cage of order $2(n^2+n+1)$, for n a prime power.

F15: If the Menger graph of a $(v, b, r, k; 0, 1)$-BD is strongly regular, then the geometry is a partially balanced incomplete block design.

EXAMPLES

E13: In general, the Menger graph of n-point geometry is K_n, while the Levi graph of that geometry is homeomorphic to K_n (each edge of K_n is replaced by a path of length two).

E14: The projective plane $PG(2, 2)$ and the 7-point geometry are clearly different geometries, having 7 and 21 lines, respectively, yet both have Menger graph K_7.

E15: The Levi graph of $PG(2, 2)$ is the Heawood graph, the unique $(3, 6)$-cage. The Levi graph of 7-point geometry is obtained from K_7 by performing an elementary subdivision on each edge; that is, by replacing each edge with a path of length two.

E16: The Menger graph of a 3-configuration $(v, b, r, 3; 0, 1)$ is $2r$-regular of order v, and its edge set decomposes into b disjoint 3-cycles. For instance, the strongly-regular octahedral graph is the Menger graph for a $(6, 4, 2, 3; 0, 1)$-PBIBD called the "Pasch configuration". The shading in Figure 7.9.2 below depicts the C_3-decomposition.

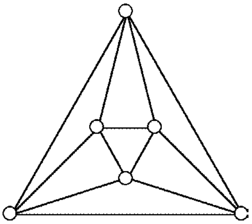


Figure 7.9.2 **The Pasch configuration.**

E17: The Menger graph for the geometry of Pappus is $K_{3(3)}$, the complement of the graph $3K_3$. The Menger graph for the geometry of Desargues is the complement of the Petersen graph. Since both Menger graphs are strongly regular, the corresponding incomplete block designs are both partially balanced.

7.9.3 Surface Models

DEFINITIONS

D17: A ***model for an axiom system*** Σ for a finite geometry (P, L) is an interpretation of the points and lines of Σ such that each interpreted axiom in Σ is a true statement about (P, L).

D18: A model of a geometry (P, L) is ***abstract*** if it specifies P as an abstract set of points and L as a collection of subsets of P. It is ***concrete*** if it represents P as a finite set of points of R^n, for some positive integer n, and L as a collection of locally one-dimensional subsets of R^n (often a geometric realization of a graph or of a hypergraph).

D19: An axiom system Σ is ***consistent*** if no contradictions can be derived from it.

D20: An axiom system Σ is ***independent*** if no axiom in Σ can be derived from the other axioms in Σ.

D21: An axiom system Σ is ***complete*** if every statement in the undefined and defined terms of Σ can either be proven true or proven false, using Σ.

D22: The ***genus of a finite geometry*** is the genus of its Levi graph.

FACTS

The first three facts below indicate the importance of models in the formal study of geometry. Informally, models have heuristic and pedagogical functions, and might even provide aesthetic pleasure.

F16: An axiom system Σ is consistent if there is a model for Σ.

F17: An axiom system Σ is independent if, for each axiom $\sigma \in \Sigma$, a model can be found satisfying $(\Sigma - \sigma) \wedge (\sim \sigma)$.

F18: An axiom system Σ is complete, if there is a unique (up to isomorphism) model for Σ.

F19: The genus of n-point geometry ($n > 2$) is $\lceil (n-3)(n-4)/12 \rceil$. In consideration of Example 13, this follows from [RiYo68].

F20: [FiWh00] The geometry of Pappus has genus 1. That of Desargues has genus 2. The Desargues model has a 3-fold rotational symmetry that fixes the point of perspectivity, the line of perspectivity, the two triangles of perspectivity, and nothing else.

F21: [Wh95] Surface models for $PG(2,q)$, q a prime power, depend upon the residue of $q(\text{mod } 3)$:

(a) If $q \equiv 2(\text{mod } 3)$, then $PG(2,q)$ has genus $1 + (q-2)(q^2+q+1)/3$; all the hyperregions are triangular.

(b) If $q \equiv 1(\text{mod } 3)$, then $PG(2,q)$ can be modelled on the surface of genus $1 + (q-1)(q^2+q+1)/3$, with q^2+q+1 hyperregions pentagonal and all others triangular.

(c) If $q \equiv 0(\text{mod } 3)$, then $PG(2,q)$ is conveniently modelled on an orientable pseudosurface of characteristic $(3-2q)(q^2+q+1)/3$, with q^2+q+1 hyperregions quadrilateral and all others triangular.

In each case, the group Z_{q^2+q+1} acts regularly, as a group of map automorphisms, on the point set, the line set, and on each orbit of the region set for the modified Levi graph imbedding.

F22: Topological models for $AG(2,q)$ are obtained by deletions from the above models for $PG(2,q)$.

F23: There is a toroidal symmetric 3-configuration $(v)_3$ on v points for all $v \geq 7$.

F24: [Wh02] There is a 3-configuration $(3n, n^2, n, 3; 0, 1)$-PBIBD with Menger graph $K_{n,n,n}$ and genus $(n-1)(n-2)/2$ for all $n > 1$. This generalizes the Pasch configuration.

F25: An imbedding of the Levi graph for a finite geometry on a closed orientable 2-manifold (a surface) can be readily modified to model the geometry (on the same surface) by an imbedded graph G having bichromatic dual: vertices model points, and boundaries of regions of one fixed color model lines. (The regions of the other color are hyperregions.) If the geometry is a 3-configuration, then G is its Menger graph. The process reverses, so that the Levi graph can be obtained from such an imbedding of G.

EXAMPLES

E18: An axiom system for the n-point geometry is as follows:

(i) There are exactly n points.

(ii) Two distinct points belong to a unique common line.

(iii) Each line consists of exactly two points.

This axiom system is consistent, since the complete graph K_n is a model. For $n = 4$, we have the following abstract model:

$$P = \{1,2,3,4\} \qquad L = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{2,4\},\{3,4\}\}$$

The graph in Figure 7.9.3 is a concrete model (which could be imbedded in R^2 or, reversing the Riemann stereographic projection, imbedded on the sphere in R^3).

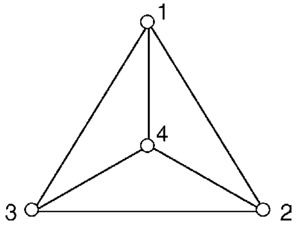


Figure 7.9.3 A concrete model for the 4-point geometry.

This axiom system is independent, for $n > 3$, since:

- The complete graph K_{n+1} shows that axiom (i) does not follow from axioms (ii) and (iii).
- The cycle graph C_n shows that axiom (ii) does not follow from axioms (i) and (iii).
- The path graph P_n (a one-line geometry) shows that axiom (iii) does not follow from axioms (i) and (ii).

This axiom system is complete, since all models are isomorphic to K_n. (The abstract and concrete models above for $n = 4$, for example, are isomorphic under the identity function on the point sets, inducing the identity function on the line sets.)

E19: The imbedded Menger graph of Example 16 of §7.9.2 can be readily modified to obtain a spherical imbedding of the corresponding Levi graph. Thus the Pasch configuration has genus 0.

E20: Conversely, the Levi graph for the Fano plane is the Heawood graph, the unique $(3, 6)$-cage, which has genus 1. The modification of a toroidal imbedding of the Heawood graph (see Fig 7.9.4a below) gives a bichromatic-dual imbedding of K_7, the Menger graph for the Fano plane (see Fig 7.9.4b below). The seven triangular regions of either color class model the lines of this geometry. Thus the Fano plane has genus one. All the defects of the traditional model for this geometry, displayed in §7.9.1 (where the lines correspond to the unshaded regions in Fig 7.9.4b), are now remedied.

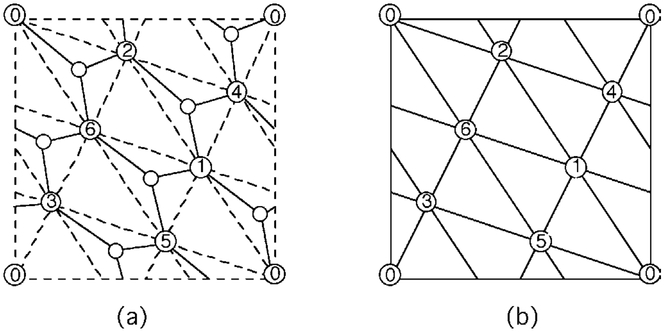


Figure 7.9.4 (a) Levi graph for the Fano plane and (b) Menger graph.

References

[BrRy49] R. H. Bruck and H. J. Ryser, The nonexistence of certain finite projective planes, *Canad. J. Math.* 1 (1949), 88–93.

[FiWh00] R. M. Figueroa-Centeno and A. T. White, Topological models for classical configurations, *J. of Statistical Planning and Inference* 86 (2000), 421–434.

[RiYo68] G. Ringel and J. W. T. Youngs, Solution of the Heawood map-coloring problem, *Proc. Nat. Acad. Sci. U.S.A.* 60 (1968), 438–445.

[Si66] R. R. Singleton, On minimal graphs of maximum even girth, *J. Combin. Theory* 1 (1966), 306–332.

[Wh95] A. T. White, Efficient imbeddings of finite projective planes, *Proc. London Math. Soc.* (3) 70 (1995), 33–55.

[Wh01] A. T. White, *Graphs of Groups on Surfaces: Interactions and Models*, North-Holland Mathematical Studies 188, 2001.

[Wh02] A. T. White, Modelling finite geometries on surfaces, *Discrete Mathematics* 244 (2002), 479–493.

GLOSSARY FOR CHAPTER 7

annulus: synonym for *cylinder*.

apex graph: a graph G in which there is a vertex v such that removing v from G results in a planar graph.

Archimedean solid: a semi-regular polyhedron; it has regular polygons as faces and the same configuration of faces at each vertex.

automorphism of a map: an isomorphism of the map onto itself.

axiom system: a list of axioms for a mathematical structure.

___, **complete**: an axiom system in which every well-formed statement can either be shown to be true or be shown to be false.

___, **consistent**: an axiom system having no contradictions.

___, **independent**: a system in which no axiom is derivable from the others.

___, **model for**: an interpretation of the undefined terms so that each interpreted axiom is true.

balanced incomplete block design (abbr. BIBD): a geometry of v points (each in r lines) and b lines (each containing k points) such that each pair of points belong to λ lines.

bar-amalgamation – of two disjoint graphs G and H: the result of running a new edge e between a vertex u of G and a vertex v of H; denoted $G_u *_e H_v$.

barycenter – of a face of a cellular graph imbedding: a point in the interior of the face, corresponding to the image of the center of the geometric polygon that the face represents.

base graph – of a voltage graph construction: the graph to whose edges the voltages are assigned.

base imbedding for an *imbedded voltage graph* $\langle G \to S, \alpha\rangle$: the imbedding $\langle G \to S\rangle$ of the *base graph* in the *base surface*.

base surface for an *imbedded voltage graph* $\langle G \to S, \alpha\rangle$: the surface S in which the voltage graph is imbedded.

BIBD: see balanced incomplete block design.

boundary – of a 2-manifold M: the subspace of those points in M that do not have neighborhoods homeomorphic to open disks; instead, their fundamental neighborhoods are homeomorphic to half-disks.

boundary-separating closed curve – in a region of a noncellular graph imbedding: a closed curve that separates that region so that at least one boundary component of the region lies on each side of the separation.

boundary-walk specification – of a polygonal complex: a list of the signed boundary walks of the faces.

bouquet B_n: the graph with one vertex and n self-loops.

branch point – in the codomain S of a branched covering $p : \tilde{S} \to S$: a point where branching occurs.

branch set – of a branched covering $p : \tilde{S} \to S$: the discrete subset $\tilde{B} \subset \tilde{S}$ to whose complement the restriction of the *branched covering* p is a *covering projection*.

branched covering, combinatorial – of a surface S with a cellularly imbedded graph G by a surface $\tilde{S}$ with a cellularly imbedded graph $\tilde{G}$: a face-to-face, edge-to-edge, vertex-to-vertex mapping that is topologically a branched covering with every branch point occurring in the interior of some face; exemplified by the natural projection associated with an imbedded *voltage graph.*

branched covering, topological: a continuous function $p : \tilde{S} \to S$ between surfaces, whose restriction to the complement $\tilde{S} - \tilde{B}$ of a discrete subset $\tilde{B} \subset \tilde{S}$ is a *covering projection.*

branched covering space: the domain of a *branched covering* $p : \tilde{S} \to S$ of a surface S.

(d, g)**-cage**: a graph of minimum order among all d-regular graphs of girth g.

canonical factors for an abelian group: the factors $\mathcal{Z}_{m_i}$ of the *canonical form.*

canonical form for an abelian group: the form $\mathcal{Z}_{m_1} \times \cdots \times \mathcal{Z}_{m_r}$, where $m_j | m_{j+1}$ for $j = 1 \ldots r$.

Cayley graph – for a group $\mathcal{A}$ and generating set X: the graph whose vertices are the elements of $\mathcal{A}$ and such that, for each element $a \in \mathcal{A}$ and each generator $x \in X$, there is a directed edge from a to ax.

Cayley map: an imbedding of a Cayley graph on a surface, possibly specified by a *rotation scheme.*

0-cell – of a polygonal complex: see *vertex of a polygonal complex.*

1-cell – of a polygonal complex: see *edge of a polygonal complex.*

2-cell – of a polygonal complex: see *face of a polygonal complex.*

2-cell imbedding – of a graph: see *cellular imbedding.*

cellular imbedding – of a graph G: an imbedding of G in which the interior of each face is homeomorphic to an open disk.

cellular imbedding – of a graph into a surface: a graph imbedding such that the interior of each face is an open disk.

centroid of a triangle: the point common to the three medians of the triangle.

cevian of a triangle: a line from a vertex of the triangle to the opposite side.

chiral map: a map that is symmetrical, but not regular.

chromatic number of a surface S: the least number of colors sufficient to properly color the faces (or vertices) of any map on S.

circular imbedding: an imbedding where each face is bounded by a simple cycle.

circumcenter of a triangle: the point common to the three perpendicular bisectors of the sides of the triangle.

closed surface: a surface that as a topological space is compact and without boundary.

closed under minors – of a graph class: a graph class $\mathcal{C}$ such that for every graph G in $\mathcal{C}$, all minors of G are also in $\mathcal{C}$.

closed-end ladder L_n: the graph obtained from the cartesian product $P_n \times K_2$ by doubling the edges $v_1 \times K_2$ and $v_n \times K_2$ at both ends of the path.

cobblestone path J_n: the graph obtained by doubling every edge of the path P_n.

complete set of forbidden minors – for a class $\mathcal{F}$ closed under minors: a set M of minimal forbidden minors such that for every graph G that is not in $\mathcal{F}$, there exists a graph in $\mathcal{M}$ that is a minor of G.

composition of a graph G with a graph H: the graph obtained from copies of H corresponding to all vertices of G, by adding all possible edges between two copies corresponding to an adjacency of G.

(r,k)-configuration: a geometry having every point in r lines, every line consisting of k points, and each pair of points in at most one line.

___, **symmetrical**: an (r,k)-configuration such that $r = k$.

congestion at an edge e of the host of a graph mapping $f : G \to H$: the cardinality $|f^{-1}(e)|$ of its preimage; terminology for modeling the *emulation of distributed computation.*

congestion of the mapping $f : G \to H$: the maximum *congestion on any edge*, taken over all edges of H; terminology for modeling the *emulation of distributed computation.*

congruent imbeddings: two imbeddings $f_1, f_2 : G \to S$ with $hf_1 = f_2\sigma$ for some surface homeomorphism $h : S \to S$ and some graph automorphism $\sigma : G \to G$.

connected sum – of two surfaces S and S': a surface obtained by excising the interior of a closed disk in each surface and then gluing the corresponding boundary curves; denoted by $S\#S'$.

consistent orientation – of a polygonal complex: orientation of the faces such that, within a union of oriented boundary walks, none of the edges is traversed twice in the same direction.

contractible closed curve – on a surface S: a simple closed curve C on S, such that the closure of one of the components of $S - C$ is a disk.

contractible edge – in a triangulation: an edge whose contraction does not create a multiple adjacency.

contractible to the triangulation $G \to S$: said of a triangulation that can be transformed into the given triangulation $G \to S$ by a sequence of edge contractions.

co-tree: the edge complement of a spanning tree of a graph.

covering or **covering projection, combinatorial**$_1$ – of a graph G by a graph $\tilde{G}$: an edge-to-edge, vertex-to-vertex mapping that is topologically a covering; exemplified by the natural projection associated with a *voltage graph.*

covering or **covering projection, combinatorial**$_2$ – of a surface S with a cellularly imbedded graph G by a surface $\tilde{S}$ with a cellularly imbedded graph $\tilde{G}$: a face-to-face, edge-to-edge, vertex-to-vertex mapping that is topologically a covering; exemplified by the natural projection associated with an imbedded *voltage graph.*

covering or **covering projection, topological**: a continuous function $p : \tilde{X} \to X$ between locally arcwise connected topological spaces, in which every point of the codomain X has an open neighborhood U such that each arc-component of $p^{-1}(U)$ is mapped homeomorphically onto U by p.

___, **regular** – onto a space X: a covering projection onto X, such that there exists a group of *covering transformations* that acts *freely* and *transitively* on it.

covering space: the domain of a *covering projection* $p : \tilde{X} \to X$.

___, **regular** – of a space X: the domain of a regular covering projection onto X.

covering transformation – for a covering projection $p : \tilde{X} \to X$: an autohomeomorphism h on $\tilde{X}$ such that $ph = p$.

covering with folds: a natural projection from the composition of a graph G with $\overline{K}_m$ to the graph G.

Coxeter complex: the barycentric subdivision of the tessellation $\{p, q\}$, formed by all mirrors of reflection symmetries.

Coxeter group (of rank 3): a group with presentation by three generators ρ_0, ρ_1, ρ_2 and the relations $\rho_0^2 = \rho_1^2 = \rho_2^2 = (\rho_0\rho_1)^p = (\rho_1\rho_2)^q = (\rho_2\rho_0)^r = 1$.

crosscap distribution – of a graph G: the sequence whose $j^{\underline{th}}$ entry is $\overline{\gamma}_j(G)$, starting with a (possibly empty) subsequence of zeroes, followed by the subsequence of the crosscap imbedding numbers, and then an infinite sequence of zeroes.

___, **polynomial** – of a graph G: the polynomial $\overline{I}_G(y) = \sum_{j=1}^{\infty} \overline{\gamma}_j(G) y^j$

crosscap – on a surface: a subspace of the surface that is homeomorphic to a Möbius band.

___, **number** – of a closed nonorientable surface S or of the sphere: the integer k such that S is homeomorphic to N_k; denoted $\tilde{\gamma}(S)$; 0 for the sphere.

crosscap number – of a graph: the minimum *crosscap number* of a surface in which the graph is imbeddable.

___, **range** – of a graph G: the integer interval $[\overline{\gamma}_{min}(G), \overline{\gamma}_{max}(G)]$.

___, **symmetric** of a group $\mathcal{A}$: the smallest number c such that $\mathcal{A}$ has a Cayley graph imbedded in a nonorientable surface of crosscap number c.

current – on a directed edge e: the value $\alpha(e)$ assigned to edge e by a *current assignment* in a group called the *current group*.

___, **assignment** – on a digraph $G = (V, E)$ imbedded in a surface S: a function α from edge set E to a group $\mathcal{B}$; used to specify a *derived digraph*; it must be stated explicitly that the algebraic values on the edges are currents, rather than voltages.

___, **group**: the group in which a *current assignment* takes its values; usually a finite group.

current graph: a pair $\langle G \to S, \alpha\rangle$ comprising an imbedded digraph and a *current assignment*; it specifies a graph imbedding.

cycle rank$_1$ – of a connected graph $G = (V, E)$: the number $|E| - |V| + 1$, which is the number of edges in a cotree of a spanning graph; denoted $\beta(G)$ (for Betti, an Italian mathematician).

cycle rank$_2$ – of a possibly non-connected graph $G = (V, E)$: the number $|E| - |V| + c$, where c is the number of components.

cylinder: a surface homeomorphic to the unit cylinder in $\mathbb{R}^3$, i.e., to

$$\{(x, y, z) \mid x^2 + y^2 = 1, 0 \leq z \leq 1\}$$

deficiency of a branch point – of order r in a regular n-sheeted branched covering: the number $n - n/r$.

deficiency $\xi(G)$ of a graph G: the minimum value of $\xi(G, T)$, over all spanning trees T of G.

deficiency $\xi(G, T)$ of a spanning tree T – in a graph G: the number of odd components of the co-tree $G - T$.

delta-Y transformation – in the theory of triangulations: see *ΔY-transformation* under *transformation.*

dense imbedding: an imbedding of "large" face-width, where "large" is relative to the context.

derived digraph – specified by a voltage graph $\langle G = (V, E), \alpha\rangle$: the covering graph G^α associated with the specified type of voltages, i.e., *permutation* or *regular.*

derived digraph, Σ_n-permutation – for a *voltage assignment* $\alpha : E(G) \to \Sigma_n$: the digraph $G^\alpha = (V^\alpha, E^\alpha)$, with $V^\alpha = V(G) \times \{1, ..., n\}$ and $E^\alpha = E(G) \times \{1, ..., n\}$; if the edge e joins vertex u to vertex v in G, then the edge $e_j = (e, j)$ joins vertex $u_j = (u, j)$ to the vertex $v_{\alpha(j)} = (v, \alpha(j))$.

derived digraph, regular – for a *voltage assignment* $\alpha : E(G) \to \mathcal{B}$: the digraph $G^\alpha = (V^\alpha, E^\alpha)$, with $V^\alpha = V(G) \times \mathcal{B}$ and $E^\alpha = E(G) \times \mathcal{B}$; if the edge e joins vertex u to vertex v in G, then the edge $e_j = (e, j)$ joins vertex $u_j = (u, b)$ to the vertex $v_{b\alpha(e)} = (v, b\alpha(e))$.

derived graph: formally, the result of deleting directions from a *derived digraph*; informally, the derived digraph itself is also called a derived graph.

derived imbedding$_1$ – for an *imbedded voltage graph*: the imbedding $G^\alpha \to S^\alpha$ of the derived graph into the derived surface; constructed as described in §7.4.

derived imbedding$_2$ – for a *current graph*: the derived imbedding for the *voltage graph* of which it is the dual.

derived surface for an *imbedded voltage graph*$\langle G \to S, \alpha\rangle$: the cellular 2-complex S^α that results from fitting to each closed walk in the set $\tilde{\Omega}$ of lifted boundary walks in the derived graph G^α a polygonal region (whose number of sides equals the length of that closed walk).

diagonal flip – in a triangulation: to switch the *diagonal* in a quadrilateral formed by two faces that meet on an edge (i.e., that edge is the "diagonal").

___, **$\mathcal{P}$-equivalent under**: two $\mathcal{P}$-triangulations that can be transformed into each other by a sequence of $\mathcal{P}$-preserving diagonal flips.

$\mathcal{P}$-preserving diagonal flip: a diagonal flip that preserves a specified property $\mathcal{P}$.

digon – in a graph imbedding: a two-sided face.

disk – closed, open: a topological space homeomorphic, respectively, to the closed or to the open unit disk.

dual, topological: a concept due to Poincaré associated with an involutory mathematical property.

___, **graph** – for a cellularly imbedded graph G in any closed surface S: the graph G^* whose vertices are the *barycenters* of the faces of the imbedding $G \to S$, such that through each edge $e \in E(G)$ there is an edge that joins the dual vertex in the region on one side of the edge to the dual vertex on the other side (a self-loop if a face meets itself on edge e).

___, **map** – of a graph imbedding $G \to S$: the map corresponding to the *dual imbedding.*

___, **imbedding** – of a graph imbedding $G \to S$: the imbedding $G^* \to S$ obtained while constructing the *dual graph.*

___, **of a current graph** $\langle G = (V, E) \to S, \alpha : E \to \mathcal{B}\rangle$: the imbedded voltage graph whose base imbedding $G^* \to S$, is dual to the imbedding $G \to S$ (which involves reversing the orientation from the primal imbedding surface, if S is orientable), such that for each primal directed edge $e \in E$, the dual edge e^* has voltage $\alpha^*(e^*) = \alpha(e)$.

ear: a path attached at its end-vertices to a graph; the name was inspired by some drawings in which such paths had the shape of human ears.

ear decomposition – of a graph G: a partition of the edge set of G into an ordered collection P_0, P_1, $\cdots$, P_r, such that P_0 is a simple cycle and P_i, $i \geq 1$, is a path with only its endpoints in common with $P_0 + \cdots + P_{i-1}$.

edge contraction$_1$ – in a graph: removing the edge e then identifying the two vertices u and v; topologically, the edge is shrunk homotopically to a point.

edge contraction$_2$ – in a triangulation: topologically shrinking an edge, and then excising the two degenerate faces (digons) that result.

edge-width – $ew(G)$ of an imbedded graph G: the length of the shortest cycle in the graph that is non-contractible in the surface.

elementary subdivision – of an edge: the operation of replacing the edge by a path of length two; a special case of the PL-topological concept of *barycentric subdivision*.

emulation of distributed computation: porting a distributed algorithm from the parallel computer (*guest*) for which it is designed to a computer (*host*) with a different parallel architecture; modeled by a graph mapping from a graph model for the guest to a graph model for the host.

essential curve – on a surface: a simple closed curve that is not contractible on the surface; that is, either it does not separate the surface, or if it separates, then neither side of the separation is a disk.

essential cycle – in an imbedded graph: a cycle that bounds no cellular region of the imbedding surface.

Euclidean space group: a group of isometries of the Euclidean plane.

euler characteristic$_1$ – of a closed surface S: the integer $\chi(S)$ defined by $\chi(S) = 2-2g$ if S is homeomorphic to g-torus S_g, and $\chi(S) = 2-k$, if S is homeomorphic to non-orientable surface with k crosscaps N_k.

euler characteristic – of a cellular imbedding of a graph $G = (V, E)$: the alternating sum $|V| - |E| + |F|$, where F is the set of faces.

euler genus $\bar{\gamma}(S)$ – of a surface S: twice the number of handles if S is orientable, and the number of crosscaps if S is non-orientable.

euler genus of a group $\mathcal{A}$: the minimum number d such that the group $\mathcal{A}$ has a Cayley graph that imbeds in a surface of Euler genus d.

___, **symmetric** of a group $\mathcal{A}$: the smallest number d such that the group $\mathcal{A}$ has a Cayley graph imbeddable in a surface of Euler genus d.

Euler's Formula – for a cellularly imbedded graph $G \to S$: the formula $v - e + f = 2 - \bar{\gamma}(S)$.

face – of a graph imbedding: a component of the complement of the image of the graph.

___, **boundary of**: the vertices and edges encountered while traversing the face boundary walk.

___, **boundary walk of**: the closed walk that encircles the face; it may have repeated edges and repeated vertices.

face of a polygonal complex: a polygon used in the construction of the polygonal complex, viewed as a subspace of that complex.

face-width $fw(G \to S)$ – of a graph imbedding $G \to S$: the minimum value of the number $|C \cap G|$, taken over all non-contractible cycles C in the surface S; also called the *representativity*;

fiber over an edge e – of a *voltage graph* $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$: in the derived graph G^α, the edge subset $\{e\} \times \mathcal{B} = \{e_b : b \in \mathcal{B}\}$.

fiber over a vertex v – of a *voltage graph* $\langle G = (V, E), \alpha : E \to \mathcal{B}\rangle$: in the derived graph G^α, the vertex subset $\{v\} \times \mathcal{B} = \{v_b : b \in \mathcal{B}\}$.

fiber over x – where x is a point in the codomain of a *topological covering projection* $p : \tilde{X} \to X$: the set $p^{-1}(x)$.

finite geometry: a geometry whose point set is finite.

flag: an ordered triple (F_0, F_1, F_2) of pairwise incident faces of a map of dimensions 0, 1, and 2, respectively.

free action – of a group on a graph: an automorphism group with no fixed vertices, except by the identity automorphism.

freely acting group $\mathcal{H}$ of *covering transformations* for a covering projection $p : \tilde{X} \to X$: a group such that no transformation except the identity has a fixed point in $\tilde{X}$.

fundamental polygon – for a closed surface S: a polygon whose edges are pairwise identified and pasted so that the resulting polygonal complex has only one face and so that it is homeomorphic to the surface S.

f-**vector** – for a graph map: see *vector*.

genus distribution polynomial – of a graph: the polynomial $I_G(x) = \sum_{j=0}^{\infty} \gamma_j(G) x^j$.

genus distribution sequence – of a graph G: the sequence whose $j^{\underline{th}}$ entry is $\gamma_j(G)$, starting with a (possibly empty) subsequence of zeroes, followed by the subsequence of the orientable imbedding numbers, and then an infinite sequence of zeroes.

genus of a surface S: the number of handles for an orientable surface (and sometimes, the number of crosscaps for a nonorientable surface); the surface S_g has genus g; denoted $\gamma(S)$.

genus of a graph: the smallest genus of any orientable *minimum genus.*

genus imbedding – of a graph: an imbedding of the graph in a surface of smallest possible genus; short for *minimum genus imbedding.*

genus of a group $\mathcal{A}$: the minimum number g such that $\mathcal{A}$ has a Cayley graph that imbeds in a surface of genus g.

___, **symmetric** $\mathcal{A}$: the smallest number g such that $\mathcal{A}$ has a Cayley graph symmetrically imbedded in an orientable surface of genus g.

genus range of a graph G: the integer interval $[\gamma_{min}(G), \gamma_{max}(G)]$.

geometry: a non-empty set of points and a non-empty collection of subsets of the point set.

graph-encoded map – often abbreviated GEM: a particular system for describing a map using colored graphs.

(p, q, r)-group A: a group with presentation

$$A = \langle x, y, z : x^2 = y^2 = z^2 = 1, (xy)^p = (yz)^q = (xz)^r = 1\rangle$$

___, **proper**: a (p, q, r) group such that the subgroup generated by xy and yz has index two.

group action on a surface: a subgroup of the homeomorphism group of the surface.

guest graph – for a graph mapping: the domain of the mapping; terminology used when modeling the *emulation of distributed computation*; see *host graph*.

half disk: a topological space homeomorphic to the unit half-disk.

hereditary property – under minors: a property such that whenever a graph has it, then so do all of its minors.

homeomorphic graphs: two graphs that become isomorphic after smoothing all their degree-2 vertices.

host graph for a graph mapping: the codomain; terminology used when modeling the *emulation of distributed computation*; see *guest graph*.

Hurwitz group: a $(2,3,7)^o$-group.

hypermap: a generalization of a graph imbedding to a hypergraph representation on a surface.

hyperregion – of an imbedding of a hypergraph in a surface: a component of the complement of the image of the hypergraph in the surface.

imbedded voltage graph: a pair $\langle G \to S, \alpha\rangle$, such that $\langle G, \alpha\rangle$ is a voltage graph such that S is a closed surface in which the graph G is (cellularly) imbedded.

imbedding – of a graph: an imbedding of the topological realization of the graph.

___, **2-cell**: see *cellular imbedding*.

___, **cellular**: an imbedding of a graph G on a surface S such that the components of $S \setminus G$ are open disks.

imbedding – of a topological space: an immersion which is (globally) one-to-one.

imbeddings, equivalent: two imbeddings $f_1, f_2 : G \to S$ with $hf_1 = f_2$ for some homeomorphism $h : S \to S$.

immersion – of a topological space: a continuous mapping that is locally one-to-one.

incenter of a triangle: the point common to the three internal angle bisectors of the triangle.

interior of a contractible cycle C on a surface S: the component of $S - C$ that is homeomorphic to the plane.

irreducible triangulation of N_2, crosscap-type: an irreducible triangulation on the Klein bottle that splits into two triangulations on the projective plane.

irreducible triangulation of N_2, handle-type: an irreducible triangulation of the Klein bottle that contains no separating cycle of length 3.

irreducible triangulation: a triangulation that has no contractible edge.

isomorphic triangulations: two triangulations on a surface such that there is an auto-homeomorphism on the surface mapping one skeleton onto the other.

isomorphism of maps: a homeomorphism of the respective surfaces that induces a graph isomorphism of the respective graphs.

isotopic triangulations: two triangulations on a surface one of which can be transformed continuously on the surface into the other.

kernel: an imbedded graph G such that for every proper minor H, there is a curve C with $\mu(H,C) < \mu(G,C)$.

Kirchhoff current law (KCL) – at a vertex v of a current graph: a possible condition, namely, that the net current at v is the group identity.

Kirchhoff current law (KCL) – on a current graph: a possible condition, namely, that KVL holds at every vertex.

Kirchhoff voltage law (KVL) – on a closed walk W in a voltage graph: a possible condition, namely, that the *net voltage* on W is the identity of the voltage group.

Kirchhoff voltage law (KVL) – on an imbedded voltage graph: a possible condition, namely, that KVL holds on every face boundary walk in the graph.

Klein bottle N_2: a closed nonorientable surface obtained by identifying the pairs of points $\{(x,y,-1),(x,-y,1)\}$ on the two boundary components of the cylinder $\{(x,y,z) \mid x^2+y^2=1, -1 \leq z \leq 1\}$; its crosscap number is 2.

large edge-width imbedding: an imbedding where the edge-width is strictly larger than the length of the longest facial cycle.

large-edge-width map: a map whose edge-width is greater than the number of edges in any face boundary.

Levi graph of a geometry: a graph whose edges join incident point/line pairs of that geometry.

LEW-imbedding: short for large edge-width imbedding.

lift of a walk W – in a *voltage graph* $\langle G,\alpha\rangle$: a walk $\tilde{W}$ in the *derived graph* that is mapped isomorphically onto W by the *natural projection.*

line$_1$: an element of a geometry; esp. an affine line in a real Euclidean space.

line$_2$: an edge of a graph.

link of a vertex v **in a triangulation**: the cycle around v through all its neighbors.

load at a vertex v in the host of a graph mapping $f : G \to H$: the cardinality $|f^{-1}(v)|$ of its preimage; terminology for modeling the *emulation of distributed computation.*

load of the mapping $f : G \to H$: the maximum *load at a vertex*, taken over all vertices of H; terminology for modeling the *emulation of distributed computation.*

2-manifold: a topological space in which each point has a neighborhood that is homeomorphic either to an open disk or to a half-disk.

map minor of M: a map $\overline{M}$ obtained from map M by deleting and/or contracting edges.

map: a cellular imbedding of a graph on a surface.

maximum crosscap imbedding – of a graph: a cellular imbedding into a closed nonorientable surface of maximum crosscap number.

maximum crosscap number of a graph G – also called the *maximum nonorientable genus*: the largest integer k such that the graph G has a cellular imbedding in the nonorientable surface N_k; denoted $\overline{\gamma}_{max}(G)$.

maximum crosscap number – of a graph: the maximum of the set of integers k such that G has a cellular imbedding in the nonorientable surface N_k; 0 if the graph is planar; denoted $\tilde{\gamma}_{\max}(G)$.

maximum genus – of a graph G: the largest integer g such that the graph G has a cellular imbedding in the orientable surface S_g; denoted $\gamma_{max}(G)$.

maximum genus imbedding – of a graph: an imbedding of the graph into a closed orientable surface of maximum genus.

medial graph – $M(G)$ of an imbedded graph G: an imbedded graph whose vertices are the edges of G and whose edges join two vertices corresponding to two consecutive edges in a face boundary of G.

Menger graph of a geometry: a graph whose edges join collinear points of a geometry.

minimal forbidden minor – for a class $\mathcal{F}$ of graphs closed under minors: a graph G that is not in $\mathcal{F}$, but such that every proper minor of G is in $\mathcal{F}$.

minimal triangulation: a simplicial polyhedral map such that the contraction of any edge results in a map that is no longer polyhedral.

minimum crosscap imbedding – of a graph: an imbedding into a closed nonorientable surface of minimum crosscap number; an imbedding in the sphere if possible.

minimum crosscap number of a graph G – also known as the *minimum nonorientable genus*: the smallest integer k such; 0 if the graph is planar; denoted $\overline{\gamma}_{min}(G)$.

minimum crosscap number – of a graph: the minimum of the set of integers k such that the graph is imbeddable in the nonorientable surface N_k; denoted $\tilde{\gamma}_{\min}(G)$ or $\tilde{\gamma}(G)$.

minimum genus – of a graph G: the minimum integer g such that the graph G has an imbedding into the orientable surface S_g of genus g; denoted $\gamma_{\min}(G)$ or $\gamma(G)$.

minimum genus imbedding – of a graph: an imbedding of the graph into a closed orientable surface of minimum possible genus.

minor of a graph G: a graph formed from G by a sequence of edge deletions and edge contractions.

Möbius band: a surface obtained from a 2-×-2 square $\{(x,y)|-1 \leq x \leq 1, -1 \leq y \leq 1\}$ by pasting the vertical sides together with the matching $(-1,x) \rightarrow (1,-x)$.

monogon – in a graph imbedding: a one-sided face.

mu-invariant $\mu(G,C)$ – for a cycle C in an imbedded graph G: half the minimum length of a closed walk W, taken over all walks W in the radial graph that are homotopic to C.

natural action of a group $\mathcal{A}$ on the Cayley graph $C(\mathcal{A},X)$: left multiplication by elements of $\mathcal{A}$.

natural projection for a voltage graph $\langle G=(V,E), \alpha : E \rightarrow \mathcal{B}\rangle$: the graph mapping $G^\alpha \rightarrow G$ comprising the vertex function $v_b \mapsto v$ and the edge function $e_b \mapsto e$. (Thus, the natural projection is given by "erasure of subscripts".)

natural projection for an imbedded voltage graph $\langle G \rightarrow S, \alpha\rangle$: the extension of the natural projection $p : G^\alpha \rightarrow G$ to the surfaces, so that it maps the center of each polygon f in the derived imbedding $G^\alpha \rightarrow S^\alpha$ to the center of the region of the imbedding $G \rightarrow S$ bounded by $p(bd(f))$.

near triangulation: a rooted map in which every nonroot face is a 3-gon.

Nebeský nu-invariant $\nu(G)$: an invariant used in calculating the crosscap number of a graph.

neighborly polyhedral map: a polyhedral map in which every pair of distinct vertices is joined by an edge.

k-nest – in an imbedding: a sequence $C_1, \ldots, C_k$ of disjoint contractible cycles such that C_i is in the interior of C_{i+1}.

net current at a vertex v of a current graph: for an abelian group, the sum of the inflowing currents; for a non-abelian group, the product of the inflowing currents in the cyclic order of the *rotation* at v.

net voltage on a walk in a voltage graph: for an abelian group, the sum of the voltages on the walk, taken in the order of traversal; for a non-abelian group, the product of the algebraic elements in its voltage sequence, in cyclic order (which is unique up to conjugacy).

noncontractible cycle – in an imbedded graph: a cycle that is noncontractible on the surface.

non-orientable 2-manifold: a 2-manifold that contains a subspace homeomorphic to the Möbius band.

non-orientable surface with k crosscaps N_k: a connected sum of k copies of the projective plane N_1.

nonrevisiting path: a path p in the graph of a map M such that the set $p \cap F$ is connected, for each face F of M.

nonseparating cycle – in an imbedded graph: a cycle whose removal separates the surface.

orientable 2-manifold: a 2-manifold which is not non-orientable.

orientation reversing curve – on a surface: a simple closed curve whose regular neighborhood is a Möbius band.

oriented boundary walk – of a face of a graph imbedding in an oriented surface: the closed walk in the 1-skeleton that results from traversing the face boundary in the direction of orientation.

oriented polygon: a polygon together with a direction (clockwise or counterclockwise) of traversal of its boundary, designated to be *preferred.*

oriented polygonal complex: polygonal complex together with a consistent orientation.

orthocenter of a triangle: the point common to the three altitudes of the triangle.

overlap matrix – for a general rotation system ρ of a graph G and a spanning tree T: the matrix $M_{\rho,T} = [m_{i,j}]$ whose entries are given for all pairs of edges e_i, e_j of the cotree $G - T$ by

$$m_{i,j} = \begin{cases} 1 & \text{if } i \neq j \text{ and } \mathit{pure}\,(\rho)\,\big|_{T+e_i+e_j} \text{ is nonplanar} \\ 1 & \text{if } i = j \text{ and edge } i \text{ is twisted} \\ 0 & \text{otherwise} \end{cases}$$

The notation $\mathit{pure}(\rho)|_{T+e_i+e_j}$ means the restriction of the underlying pure part of the rotation system ρ to the subgraph $T + e_i + e_j$.

panel – of a triangulation: a 3-cycle in the skeleton that always bounds a face in any imbedding of the skelton graph in that same surface.

panel structure – of a triangulation: the composite structure of a triangulation and its panels.

panel structures, equivalent: two panel structures whose skeletons with all panels inserted form homeomorphic 2-complexes.

parallel lines – of a geometry: lines with no point in common.

partially balanced incomplete block system (abbr. PBIBD): a geometry of v points (each in r lines) and b lines (each containing k points), together with a strongly regular graph (whose vertices are the points of the geometry) such that two non-adjacent points belong to λ_1 lines and two adjacent points belong to λ_2 lines.

pasting topological spaces X and Y – along homeomorphic subspaces: obtaining a new topological space from the original ones by identifying the points of the homeomorphic subspaces under a homeomorphism.

permutation scheme: a particular system for describing a map using a pair of permutations.

pinched open disk: a topological space obtained from several copies of open disks by identifying their respective centers to a single vertex.

planar graph: a graph whose minimum genus is 0.

planarizing collection of cycles: a set $C_1, \ldots, C_g$ of cycles in an imbedded graph such that cutting along all of the C_i simultaneously yields a connected graph imbedded in the plane.

planarizing curve – for a nonplanar region of a noncellular graph imbedding: a separating closed curve such that all of the boundary components lie to one side of the separation and all of the genus lies to the other.

point$_1$: a point of Euclidean space or a topological space.

point$_2$: a vertex of a graph.

polygonal complex: roughly, a topological space obtained from a set of oriented polygons by pasting some of these polygons to each other (and to themselves) along their sides.

polyhedral imbedding: an imbedding such that the intersection of any two face boundaries is either empty or a path.

polyhedral map: a map M whose face boundaries are cycles, and such that any two distinct face boundaries are either disjoint or meet in either a single edge or vertex.

___, **weakly neighborly**: a polyhedral map for which every pair of vertices is contained on a face.

preferred direction – of the traversal of a polygon boundary: a chosen direction of traversal of sides of the polygon (clockwise or counterclockwise).

projective plane N_1: a closed surface obtained from the closed unit disk by identifying pairs of boundary points that are diametrically opposite relative to the center of the disk.

projective plane: the nonorientable surface of genus 1.

2-pseudomanifold: a topological space in which each point has a neighborhood that is homeomorphic either to an open disk, to a half-disk, or to a pinched disk.

pseudosurface: a 2-pseudomanifold, usually assumed to be connected.

punctured surface: a surface with one boundary component.

quadrangulation: a graph imbedding all of whose faces are 4-sided.

quadrilateral: a 4-sided face of a graph imbedding.

radial graph $R(G)$ of an imbedded graph G: an imbedded bipartite graph whose vertices are the vertices and faces of G, and whose edges join incident elements.

ramification point of a covering: see *branch point*.

realization – of a map: an imbedding of the map into Euclidean space $\mathbf{E}^d$ such that each face is a plane convex polygon and adjacent faces are not coplanar.

regular map: a map whose automorphism group acts transitively on the set of *flags*.

representativity $\rho(G)$ – of an imbedded graph G: same as the face-width of the imbedding.

rooted imbedding: an imbedding with a distinguished vertex v, an edge e incident with v, and a face f incident with e.

rooted map: a map in which a flag has been distinguished.

rotation (global) – on a graph: an assignment of a rotation at each vertex.

rotation at a vertex: a cyclic permutation on the set of half-edges incident to the vertex.

rotation scheme: a purely combinatorial description of an imbedding of a graph G on a surface, by giving a *rotation* at each vertex of G.

semicellular graph imbedding – of a graph G: an imbedding $G \to S$ whose regions are planar, but which may have more than one boundary component.

separating closed curve – on a surface: a simple closed curve the excision of which splits the surface into two components.

p-**sequence** – of a polyhedral map: the sequence $\{p_i\}$, where p_i is the number of i-gonal faces.

v-**sequence** – of a polyhedral map: the sequence $\{v_i\}$, where v_i is the number of vertices of degree i.

signed boundary walk – of a face of a polygonal complex: the list of the signed edges that occur on an oriented boundary walk of that face.

signed edge – in a polygonal complex: the occurrence of an oriented edge or of its reverse edge within a walk in the 1-skeleton of the polygonal complex.

similar imbeddings: two imbeddings such that one can be changed into the other by a sequence of $Y\Delta$- and ΔY-transformations and the taking of geometric duals.

simple map: a map in which each vertex has degree 3.

simplicial map: a map where each face boundary is a 3-cycle.

skeleton, or **1-skeleton** – of a polygonal complex: the graph consisting of the vertices and edges of the polygonal complex.

skew polyhedron: a realization of a polyhedral map in $\mathbb{R}^d$, for $d > 3$.

skew vertex – in a triangulation: a vertex whose skeleton has at least two different cycles that contain all of its neighbors.

smoothing – a degree-2 vertex: an operation that removes a degree-2 vertex v then adds a new edge between the two neighbors of v.

sphere S_0: a surface homeomorphic to the standard sphere $\{(x, y, z) \mid x^2 + y^2 + z^2 = 1\}$ in $\mathbb{R}^3$.

standard triangulation on the sphere: an n-vertex triangulation on the sphere whose skeleton is isomorphic to the join $P_{n-2} + K_2$.

star neighborhood of a vertex v in a triangulation: the wheel obtained by joining the link of v to v.

Steiner triple system: a *balanced incomplete block design* with $k = 3$ and $\lambda = 1$.

strong symmetric genus of a group $\mathcal{A}$: the smallest number g such that the group $\mathcal{A}$ has a Cayley graph with a strongly symmetric imbedding in an orientable surface of genus g.

strongly cellular imbedding – of a graph: an imbedding such that the closure of each face is a closed disk.

strongly noncellular graph imbedding $G \to S$: an imbedding with at least one nonplanar region.

strongly noncontractible closed curve – in a region of a noncellular graph imbedding: a curve such that cutting it open reduces the genus of the region.

strongly symmetric imbedding of a Cayley graph $C(\mathcal{A}, X)$ in an orientable surface S: an imbedding such that the natural action of $\mathcal{A}$ on $C(\mathcal{A}, X)$ extends to an orientation-preserving action on the surface S.

surface minor – of an imbedded graph: another imbedded graph in the same surface formed by a sequence of edge deletions and contractions in the surface.

surface with k holes: a surface obtained by removing the interiors of k disjoint disks from a closed surface.

surface: a 2-manifold, usually assumed to be connected, compact, and without boundary, unless otherwise declared.

symmetric imbedding of a Cayley graph $C(\mathcal{A}, X)$ in a surface S: an imbedding such that the natural action of $\mathcal{A}$ on $C(\mathcal{A}, X)$ extends to the surface S.

symmetrical map: a map with at most two orbits under the action of the automorphism group on the set of flags.

symmetry group of an imbedding $f : G \to S$: the subgroup of automorphisms that are symmetries of an imbedding f; denoted Sym(f).

symmetry of an imbedding $f : G \to S$: an automorphism $\sigma \in \mathrm{Aut}(G)$ such that $hf = f\sigma$ for some homeomorphism $h : S \to S$.

tessellation $\{p, q\}$: the classical tiling of the sphere, Euclidean plane, or hyperbolic plane into p-gons, of which q are incident at each vertex.

topological realization – of a graph: a topological space obtained from the graph by first assigning to each of its edges a closed interval and then identifying endpoints of intervals according to the coincidences of the corresponding endpoints of edges of the graph.

g-**torus** S_g: a connected sum of g copies of a torus; this surface is usually called the *orientable surface of genus g*.

torus S_1: the orientable surface of genus 1; a closed surface obtained by rotating a circle $\{(x, y, z) \mid (x-2)^2 + y^2 = 1, z = 0\}$ around the y-axis.

total imbedding distribution – of a graph G: the bivariate polynomial

$$\ddot{I}_G(x, y) = I_G(x) + \overline{I}_G(y) = \sum_{j=0}^{\infty} \gamma_j(G) x^j + \sum_{j=1}^{\infty} \overline{\gamma}_j(G) y^j$$

ΔY-transformation – in the theory of triangulations: the operation of deleting the three edges joining three mutually adjacent vertices and inserting a new vertex with new edges to all three vertices; the inverse of a $Y\Delta$-transformation.

$Y\Delta$-transformation – in the theory of triangulations: a graph formed by deleting a vertex v of degree three and adding in a new 3-cycle incident with its neighbors.

transitively acting group $\mathcal{H}$ – of *covering transformations* for a *covering projection* $p : \tilde{X} \to X$: a group whose restriction to every fiber is a transitive permutation group.

triangle group: a group of isometries generated by rotation about the vertices of a triangle with angles $\pi/p, \pi/q, \pi/r$; the symmetry group of the *tessellation* of type $\{p, q\}$.

___, **full**: a group generated by the reflections in the sides of a triangle with angles $\pi/p, \pi/q, \pi/r$.

triangle$_1$: a 3-sided polygon, a figure in plane geometry; see *centroid, cevian, circumcenter, incenter, orthocenter.*

triangle$_2$: a 3-sided face of a graph imbedding.

triangle$_3$: a 3-cycle of a graph.

triangular imbedding: an imbedding that imbeds a graph with all faces 3-sided.

triangulates a surface – a possible graph property: having a triangular imbedding in some surface.

triangulation of a surface: a simplicial map where each face boundary is a 3-cycle.

___, **Catalan**: a triangulation on a surface-with-boundary whose boundary includes all vertices.

___, **clean**: a triangulation such that every 3-cycle in the skeleton bounds a face.

___, **eulerian**: a triangulation with each vertex of even degree.

___, **frozen**: a triangulation such that no edge can be flipped without giving the skeleton a double adjacency.

___, **k-irreducible**: a triangulation such that each edge is contained in an essential cycle of length at least k.

___, **minimal clean**: a clean triangulation minimal with respect to edge contractions.

___, **minimal** – of a surface: a triangulation on the surface having the fewest vertices.

___, **k-minimal**: the same as a k-irreducible triangulation.

___, **pseudo-minimal**: a triangulation such that no sequence of diagonal flips transforms it into one having a vertex of degree 3.

___, **tight**: a triangulation $G \to S$ such that, for any partition of $V(G)$ into three nonempty subsets V_1, V_2 and V_3, there is a face $v_1v_2v_3 \in F(G \to S)$ with $v_i \in V_i$.

___, **untight**: a triangulation that is not tight.

type-$\{p, q\}$ map: a map with p edges incident with each vertex and q edges incident with each face.

underlying cellular imbedding – of a semicellular graph imbedding: the imbedding obtained by cutting each non-cell region open along a maximal family of boundary-separating closed curves and capping the holes with disks.

underlying semicellular imbedding – of a strongly noncellular graph imbedding $G \to S$: the imbedding obtained by cutting each non-cell region open along a maximal family of boundary-separating closed curves and then capping the holes with disks.

unimodal sequence $\{a_m\}$: a sequence such that there exists at least one integer M such that

$$a_{m-1} \leq a_m \text{ for all } m \leq M \quad \text{and} \quad a_m \geq a_{m+1} \text{ for all } m \geq M$$

uniquely imbeddable on a surface S – a possible graph property: having a unique imbedding on the surface (up to a suitable equivalence).

unit disk – closed, open: respective subsets $\{(x,y) \mid x^2+y^2 \leq 1\}$ and $\{(x,y) \mid x^2+y^2 < 1\}$ of Euclidean plane together with the inherited Euclidean topology.

unit half-disk: a subset $\{(x,y) \mid x \geq 0, x^2+y^2 < 1\}$ of Euclidean plane together with the inherited Euclidean topology.

upper-imbeddable graph: a graph G whose maximum genus is equal to $\lfloor \beta(G)/2 \rfloor$, where $\beta(G)$ is the cycle rank of G.

f**-vector** – for a graph map: the triple (f_0, f_1, f_2) where f_i is the number of i-dimensional faces of the map.

vertex – of a polygonal complex: the image of arbitrarily many polygon corners that have been pasted together when building the polygonal complex.

vertex-amalgamation – of two disjoint graphs G and H: the result of identifying a vertex u of G and a vertex v of H. Notation: $G_u * H_v$.

vertex-face graph: same as the radial graph.

vertex-face incidence graph: same as the radial graph.

vertex splitting: an operation on a map inverse to *edge contraction* – a single vertex is replaced by two vertices joined by an edge.

vertex-transitive action of a group of automorphisms on a graph: a group such that for any pair of vertices, there is an automorphism taking one vertex to the other.

vertex-transitive map: a map whose automorphism group acts transitively on the set of vertices.

voltage – on a directed edge e: the value $\alpha(e)$ assigned to e by a *voltage assignment.*

voltage assignment – on a digraph $G = (V, E)$: a function α from edge-set E to a group $\mathcal{B}$; used to specify a *derived digraph.*

voltage graph: a pair $\langle G, \alpha \rangle$, where G is a digraph and $\alpha : E_G \to \mathcal{B}$ is a *voltage assignment*; an algebraic specification of a *derived graph.*

voltage group: the group in which a *voltage assignment* $\alpha : E_G \to \mathcal{B}$ takes its values.

voltage sequence on a walk $W = v_0, e_1, v_1, e_2, ..., e_n, v_n$ in a voltage graph $\langle G, \alpha \rangle$: the sequence of voltages $a_1, ..., a_n$ encountered, where $a_j = \alpha(e_j)$ or $\alpha(e_j)^{-1}$, respectively, depending on whether edge e_j is traversed in the forward or backward direction.

walk in a voltage graph $\langle G, \alpha \rangle$: a walk in G as if it were undirected, so that some of its edge-steps may proceed in opposite direction from the assigned direction on the edge it traverses.

k**-walk**: a spanning walk that visits no vertex more than k times.

weakly neighborly polyhedral map: see *polyhedral map.*

wheel-neighborhood, having a – a possible property of a vertex v: the property that any two face boundaries containing v intersect in a path.

Xuong tree T in a graph G: a spanning tree whose deficiency is equal to the deficiency of the graph G.

$Y\Delta Y$**-equivalent graphs**: two graphs such that one can be changed into the other by a sequence of $Y\Delta$- and ΔY-transformations.

$Y\Delta$**-transformation**: see *transformation*.

Chapter 8

ANALYTIC GRAPH THEORY

8.1 EXTREMAL GRAPH THEORY

Bela Bollobás, University of Memphis
Trinity College, Cambridge, UK

Vladimir Nikiforov, University of Memphis

8.2 RANDOM GRAPHS

Nicholas Wormald, University of Waterloo, Canada

8.3 RAMSEY GRAPH THEORY

Ralph Faudree, University of Memphis

8.4 THE PROBABILISTIC METHOD

Alan Frieze, Carnegie Mellon University

GLOSSARY

8.1 EXTREMAL GRAPH THEORY

Bela Bollobás, University of Memphis
Trinity College, Cambridge, UK

Vladimir Nikiforov, University of Memphis

8.1.1 Turán-Type Problems
8.1.2 The Number of Complete Graphs
8.1.3 Erdős-Stone Theorem and Its Extensions
8.1.4 Zarankiewicz Problem and Related Questions
8.1.5 Paths and Trees
8.1.6 Circumference
8.1.7 Hamiltonian Cycles
8.1.8 Cycle Lengths
8.1.9 Szemerédi's Uniformity Lemma
8.1.10 Asymptotic Enumeration
8.1.11 Graph Minors
8.1.12 Ramsey-Turán Problems
References

Introduction

Extremal graph theory is concerned with inequalities among functions of graph invariants and the structures that demonstrate that these inequalities are best possible. Accordingly, in its wide sense, it encompasses most of graph theory. Nevertheless, there is a clearly identifiable body of core extremal results: in this all-too-brief review we shall present a selection of narrowly interpreted extremal results.

Since by now there are many thousands of papers on extremal graph theory, no short survey of extremal graph theory has any hope of being complete. There is no doubt that the selection of topics in this survey, in which we shall concentrate on the basic graph invariants such as size, maximal and minimal degrees, connectivity, number of r-cliques, and independence number, strongly reflects the tastes and preferences of its authors.

NOTATIONS AND CONVENTIONS

CONVENTION Unless explicitly stated, all graphs are assumed to be defined on the vertex set $[n] = \{1, 2, \ldots, n\}$.

CONVENTION $G(n)$ stands for a graph with n vertices and $G(n, m)$ stands for a graph with n vertices ("of *order* n") and m edges ("of *size* m"). Thus, the statement

> "...if in $G = G(n)$ with $n \geq 3$, every vertex has degree at least $n/2$, then G is Hamiltonian..."

means that every graph of order $n > 3$ and minimal degree $n/2$ is Hamiltonian.

NOTATION: $v(G)$ and $e(G)$ denote the numbers of vertices and edges in a graph G.

NOTATION: $\Delta(G)$ and $\delta(G)$ stand for the maximal and minimal degrees of G.

NOTATION: If u is a vertex of a graph G, then $\Gamma_G(u)$ is its set of neighbors, and $d_G(u) = |\Gamma_G(u)|$ is its degree. We use $d(u)$ and $\Gamma(u)$ instead of $d_G(u)$ and $\Gamma_G(u)$ when it is clear which graph G is intended.

NOTATION: If the graphs G_1 and G_2 are disjoint graphs, then $G_1 + G_2$ denotes the *join* of G_1 and G_2 (see Subsection 1.1.3), that is, the union of G_1 and G_2 together with some new edges joining every vertex of G_1 to every vertex of G_2.

NOTATION: We denote by $k_s(G)$ the number of copies of the complete graph K_s in the graph G.

DEFINITIONS

D1: Given graphs H and G, we say that G is ***H*-free** if G has no subgraph isomorphic to H.

D2: An ***s*-clique** is a complete subgraph on s vertices. Thus, $k_s(G)$ equals the number of s-cliques of the graph G.

D3: Given a graph G, the ***clique number*** $\omega(G)$ of G is the order of its largest clique.

D4: The ***independence number*** $\alpha(G)$ of G is the clique number of its complement.

D5: A graph is ***k*-connected** if the deletion of fewer than k of its vertices leaves it connected. The vertex connectivity $\kappa(G)$ of a graph G is the maximal number k such that G is k-connected.

EXAMPLES

Let us illustrate the difference between the wide and narrow interpretations of extremal graph theory with two examples.

E1: Let $f(n)$ be the minimal number of triangles needed to cover all the edges of a complete graph with n vertices. Then

$$f(n) \leq \binom{n}{2}/3$$

and equality holds iff there is a Steiner triple system of order n. This assertion is not really a result in extremal graph theory, but leads to *design theory.*

E2: Consider the statement that "every 2-connected graph with n vertices and minimal degree k contains a cycle with at least min$\{2k, n\}$ vertices". This is clearly a result in extremal graph theory with the narrow interpretation: it expresses a relationship involving the number of vertices, the minimal degree, the connectivity, and the circumference (maximal length of a cycle).

8.1.1 Turán-Type Problems

The quintessential problem in extremal graph theory is the following question due to Turán: given $3 \leq r \leq n$, what is the maximal size graph $G(n)$ that does not contain K_r? Equivalently, what is the maximal size of a K_r-free $G(n)$?

DEFINITION

The Turán problem has been substantially generalized.

D6: Let $\{F_n\}_{n=1}^{\infty}$ be a sequence of families of graphs, and let $\Phi(n, F_n)$ be the set of graphs $G(n)$ that are H-free for every $H \in F_n$.

- The function $ex(n, F_n)$ of n is called the ***extremal function*** of the sequence $\{F_n\}$.
- The graphs $G \in \Phi(n, F_n)$ for which $e(G) = ex(n, F_n)$ are called ***extremal graphs***.
- In this context, the families $\{F_n\}_{n=1}^{\infty}$ are called ***forbidden graphs***.
- The aim to find or estimate $ex(n, F_n)$ and to determine the extremal graphs is called a ***Turán type problem*** for the families of forbidden graphs $\{F_n\}_{n=1}^{\infty}$.

Turán's Theorem and Its Extensions

The fundamental theorem of Turán has been the driving force of extremal graph theory for more than six decades.

DEFINITIONS

D7: Let $n \geq r \geq 2$ be integers. The ***Turán graph*** $T_r(n)$ is the complete r-partite graph whose classes are as nearly equal as possible.

D8: The ***Turán number*** $t_r(n)$ is the size of the Turán graph $T_r(n)$.

FACTS

F1: If $n = rs + t$ $(0 \leq t \leq r - 1)$, then $T_r(n)$ has t classes of cardinality $\lceil n/r \rceil$ and $r - t$ classes of cardinality $\lfloor n/r \rfloor$. Therefore,

$$t_2(n) = \left\lfloor \frac{n^2}{4} \right\rfloor, \quad t_r(n) = \frac{r-1}{2r}\left(n^2 - t^2\right) + \binom{t}{2}$$

$$\frac{r-1}{2r}n^2 \;\geq\; t_r(n) \;\geq\; \frac{r-1}{2r}n^2 - \frac{n}{4}$$

F2: **Mantel** [Man07] If a graph $G = G(n)$ is K_3-free, then $e(G) \leq \lfloor n^2/4 \rfloor$.

F3: **Turán's theorem** [Tur41] If a graph $G = G(n)$ is K_{r+1}-free, then $e(G) \leq t_r(n)$; and if $e(G) = t_r(n)$, then $G = T_r(n)$.

F4: **Zykov** [Zyk49], **Erdős** [Erd62a] If a graph $G = G(n)$ is K_{r+1}-free, then $k_s(G) \leq k_s(T_r(n))$ for every $s = 2, \ldots, r$; and if $k_s(G) = k_s(T_r(n))$ for some s such that $2 \leq s \leq r$, then $G = T_r(n)$.

F5: **Erdős** [Erd70] For every K_{r+1}-free graph G, there exists an r-partite graph H with $V(H) = V(G)$, such that $d_G(u) \leq d_H(u)$ for every $u \in V(G)$. If G is not a complete r-partite graph, then H may be chosen so that $d_G(u) < d_H(u)$ for some $u \in V(G)$.

F6: **Khadžiivanov** [Kha77], **Fisher and Ryan** [FiRa92] If the graph G is K_{r+1}-free, then for every $s = 1, \ldots, r - 1$ we have

$$\left(\frac{k_s(G)}{\binom{r}{s}}\right)^{1/s} \geq \left(\frac{k_{s+1}(G)}{\binom{r}{s+1}}\right)^{1/(s+1)}$$

F7: **Motzkin-Straus inequality** [MoSt65] If a graph $G = G(n)$ is K_{r+1}-free and $x_1, \ldots, x_n$ are nonnegative numbers then

$$\sum_{v \in V(G)} \sum_{u \in \Gamma(v)} x_u x_v \quad \leq \quad \frac{r-1}{r} \left(\sum_{v \in V(G)} x_v \right)^2$$

F8: **Bomze inequality** [Bom97] If the graph $G = G(n)$ is K_{r+1}-free and if the numbers $x_1, \ldots, x_n$ are nonnegative and not all zero, then

$$\sum_{v \in V(G)} \sum_{u \in \Gamma(v)} x_u x_v + \frac{1}{2} \sum_{u \in V(G)} x_u^2 \quad \leq \quad \left(1 - \frac{1}{2r}\right) \left(\sum_{v \in V(G)} x_v \right)^2 \qquad (1)$$

Let $x_{u_1}, \ldots, x_{u_q}$ be the nonzero numbers in $\{x_1, \ldots, x_n\}$. Equality in (1) is attained if and only if $q = r$, $x_{u_1} = \cdots = x_{u_r}$, and the vertices $u_1, \ldots, u_r$ are an r-clique in G.

Structural Properties of the Graphs $G(n, t_r(n) + 1)$

Turán's theorem guarantees that every $G(n, t_r(n) + 1)$ contains a K_{r+1}, but further investigations revealed a lot more properties of such graphs. We present below three topics of considerable interest.

NOTATIONS

NOTATION: $K_r(s_1, \ldots, s_r)$ denotes the complete r-partite graph with classes of size $s_1, \ldots, s_r$ respectively. The graph $K_r^+(s_1, \ldots, s_r)$ is obtained from $K_r(s_1, \ldots, s_r)$ by adding an edge to the first specified class, i.e., of order s_1.

NOTATION: For all natural numbers n, m, r let $\delta(n, m, r)$ denote the maximal value such that every graph $G(n, m)$ has an r-clique R with $\sum \{d(u) : u \in R\} \geq \delta(n, m, r)$.

FACTS

F9: **Bollobás and Thomason** [BoTh81], **Erdős and Sós** [ErSo83] For $r \geq 3$ every $G(n, t_r(n) + 1)$ has a vertex u with degree $d(u) > \left(1 - 1/r - (1 + \sqrt{r})^{-1}\right) n$ and such that $\Gamma(u)$ induces more than $t_{r-1}(d(u))$ edges.

F10: **Bondy** [Bon83a], [Bon83b] For $r \geq 3$ every $G(n, t_r(n) + 1)$ has a vertex u of maximal degree such that $\Gamma(u)$ induces more than $t_{r-1}(d(u))$ edges.

F11: **Bollobás** [Bol99] Let $G = G(n, t_r(n) + a)$ where $a \geq 0$. Let u be a vertex of maximal degree $d(u) = n - k$. Then $e(G[\Gamma(u)]) \geq t_r(d(u))$; and the inequality is strict unless $k = \lfloor n/r \rfloor$, $d(u) = \lceil (r-1)n/r \rceil$, the set $V \backslash \Gamma(u)$ is independent, and every vertex of $\Gamma(u)$ is joined to every vertex of $V - \Gamma(u)$.

F12: **Erdős** [Erd63] For every $\varepsilon > 0$ there exists a number $c = c(\varepsilon) > 0$ such that every $G(n, \lfloor n^2/4 \rfloor + 1)$ contains a $K_2^+(c \log n, n^{1-\varepsilon})$.

F13: **Erdős and Simonovits** [ErSi73] For every q and n sufficiently large, every $G(n, t_r(n) + 1)$ contains a $K_r^+(q, \ldots, q)$.

F14: **Edwards** [Edw77], [Edw78] It $3 \leq r \leq 8$, $n > r^2$ and $m \geq t_r(n)$, then

$$\delta(n, m, r) \geq \frac{2rm}{n}$$

F15: **Faudree** [Fau92] If $r \geq 3$, $n > r^2(r-1)/4$, and $m \geq t_r(n)$, then

$$\delta(n, m, r) \geq \frac{2rm}{n} \tag{2}$$

REMARKS

R1: In a sense, Fact 13 is best possible, since if H is a fixed graph that occurs in any graph $G = G(n, t_r(n) + 1)$ then $H \subset K_r^+(q, \ldots, q)$ if q is sufficiently large. However, Fact 12 suggests that extensions are still possible, although we are not aware of any such extension.

R2: Bollobás and Nikiforov showed that Fact 15 holds for every n.

R3: The result in Fact 16 confirms a conjecture of Bollobás and Erdős [BoEr76], and it is essentially best possible, since if G is regular, then equality holds in (2). On the other hand, if $m \leq t_{r-1}(n)$, then $\delta(n, m, r) = 0$. It is a difficult open question to determine $\delta(n, m, r)$ for $t_{r-1}(n) < m < t_r(n)$ (see, e.g., [CEV88]).

Books and Generalized Books

The study of books was initiated by Erdős in 1962 [Erd62b] and has attracted much effort since then. Nevertheless, the Turán problems about books, except for the simplest case, are largely open.

DEFINITION

D9: For $q \geq 1$, $r \geq 1$ an ***r*-book** is the graph $B_q^{(r)}$ consisting of q distinct $(r+1)$-cliques, sharing a common r-clique.

- The value q is called the ***size*** of the r-book; we write $bk^{(r)}(G)$ for the size of the largest r-book in a graph G.
- We call 2-books simply ***books*** and write $bk(G)$ for $bk^{(2)}(G)$.

FACTS

F16: **Dirac** [Dir63] Every $G = G(n, t_r(n)+1)$ contains a K_{r+2} with one edge removed.

F17: **Edwards** [EdMS], **Khadžiivanov, and Nikiforov** [KhNi79]

$$bk\left(G\left(n, \left\lfloor\frac{n^2}{4}\right\rfloor + 1\right)\right) > \frac{n}{6}$$

and this inequality is essentially best possible in view of the following graph. Let $n = 6k$. Partition $[n]$ into 6 sets A_{11}, A_{12}, A_{13}, A_{21}, A_{22}, A_{23} with $|A_{11}| = |A_{12}| = |A_{13}| = k-1$ and $|A_{21}| = |A_{22}| = |A_{23}| = k+1$. For $1 \leq j < k \leq 3$ join every vertex of A_{ij} to every vertex of A_{ik}, and for $j = 1, 2, 3$ join every vertex of A_{1j} to every vertex of A_{2j}. The size of the resulting graph is greater than $\lfloor n^2/4 \rfloor + 1$, and its booksize is $n/6 + 1$.

F18: **Erdős, Faudree, and Rousseau** [EFR94] If $m > (r-1)n^2/2r$ and if n is sufficiently large, then

$$bk^{(r)}(G(n,m)) \geq \frac{3r-4}{8r(r+1)}n$$

For $q \geq r > 2$, $s > 0$ define the graph G with $V(G) = [r] \times [q] \times [s]$ and join two vertices (x_1, y_1, z_1) and (x_2, y_2, z_2) if and only if $x_1 \neq x_2$ and $y_1 \neq y_2$. Setting

$$n = rqs \quad \text{and} \quad m = \left(1 - \frac{1}{q}\right)\left(1 - \frac{1}{r}\right)\frac{n^2}{2}$$

we have $G = G(n,m)$, and for $1 < k < r$,

$$bk^{(r)}(G) = \left(1 - \frac{k}{r}\right)\left(1 - \frac{k}{q}\right)n$$

F19: **Erdős, Faudree and Györi** [EFG95] There exists a number $c > 1/6$ such that if $G = G(n)$ and $\delta(G) > n/2$, then $bk(G) > cn$.

Vertex-Disjoint Cliques

The next two results are milestones in extremal graph theory. In particular, the proof of Hajnal-Szemerédi theorem, although somewhat simplified by Bollobás in [Bol78], is still very difficult.

DEFINITION

NOTATION: The union of s vertex-disjoint copies of a graph G is denoted by sG.

D10: Let H and G be graphs such that $v(G) = kv(H)$. If $kH \subset G$ then G is said to have an ***H*-factor**.

FACTS

F20: **Corrádi-Hajnal theorem** [CoHa63] Let n, k be natural numbers with $n \leq 3k$ and $s = \lfloor n/k \rfloor$, and let $t = k - (n - ks)$. If $G = G(n)$ and $\delta(G) \geq 2k$, then G contains k vertex-disjoint cycles of length at most $s + 1$, and t of them are of length at most s. In particular, if $\delta(G) \geq 2n/3$, then G contains $\lfloor n/3 \rfloor K_3$.

F21: **Hajnal-Szemerédi theorem** [HaSz70] If $G = G(n)$ and $\Delta(G) \leq r$, then $V(G)$ can be partitioned into $r + 1$ independent sets such that the sizes of any two sets differ by at most 1.

8.1.2 The Number of Complete Graphs

An exciting and difficult problem is to determine min $k_s(G)$ for a given value of $k_r(G)$. In spite of the few illuminating results to be presented below, the general problem remains largely unsolved.

FACTS

F22: **Rademacher** [Erd62b]

$$k_3\left(G\left(n, \left\lfloor\frac{n^2}{4}\right\rfloor + 1\right)\right) \ \geq \ \left\lfloor\frac{n}{2}\right\rfloor$$

and this inequality is best possible.

F23: **Lovász and Simonovits** [LoSi83] If $l \leq \lfloor n/2 \rfloor$, then

$$k_3\left(G\left(n, \left\lfloor\frac{n^2}{4}\right\rfloor + l\right)\right) \ \geq \ l\left\lfloor\frac{n}{2}\right\rfloor$$

and this inequality is best possible.

F24: Let $0 \leq l \leq n/2r$ and suppose the graph G is obtained by adding l disjoint edges to one of the larger classes of the Turán graph $T_r(n)$. Then $k_{r+1}(G)$ is given by

$$k_{r+1}(G) = f_r(n, l) = l \prod_{i=0}^{r-2} \left\lfloor\frac{n+i}{r}\right\rfloor$$

F25: **Erdős** [Erd69] For every r there exist $c = c(r) > 0$ and $n_0 = n_0(r)$, such that if $n > n_0$ and $0 < l < cn$, then

$$k_{r+1}(G(n, t_r(n) + l)) \geq f_r(n, l)$$

and this inequality is best possible.

F26: **Fisher** [Fis89] If $G = G(n, m)$ and $n^2/4 \leq m \leq n^2/3$ then

$$k_3(G) \geq \frac{9nm - 2n^3 - 2(n^2 - 3m)^{3/2}}{27}$$

and this is best possible up to a term of order $O(n^2)$.

F27: **Nordhaus and Stewart** [NoSt63], **Moon and Moser** [MoMo62] If $G = G(n)$ and $k_s(G) > 0$ then

$$(s^2 - 1)\frac{k_{s+1}(G)}{k_s(G)} \geq s^2 \frac{k_s(G)}{k_{s-1}(G)} - n$$

F28: **Bollobás** [Bol76] Suppose $2 \leq s < r \leq n$. Let the function $\phi(x)$ be defined in the interval $\left[0, \binom{n}{s}\right]$ such that for every $q = s, \ldots, n$,

(i) $\phi(k_s(T_q(n))) = k_r(T_q(n))$;

(ii) ϕ is linear in the interval $[k_s(T_{q-1}(n)), k_s(T_q(n))]$.

Then

$$k_r(G) \geq \phi(k_s(G))$$

8.1.3 Erdős-Stone Theorem and Its Extensions

The fundamental theorem of Erdős and Stone has attracted the attention of researchers for more than 50 years; no doubt this will continue in the future. The theorem can be viewed as a considerable extension of Turán's theorem: slightly more than $t_r(n)$ edges in a graph of order n guarantees not only a K_{r+1} but a $K_{r+1}(q)$ for q fairly large. Equivalently, the Erdős-Stone theorem solves asymptotically the Turán problem for a fixed family of forbidden graphs.

NOTATION

NOTATION: For $\varepsilon > 0$ and natural $2 \leq r \leq n$, let $g(n, r, \varepsilon)$ denote the maximal number q such that every $G(n, \lceil (1 - 1/r + \varepsilon)n^2 \rceil)$ contains a $K_{r+1}(q)$ for n sufficiently large.

FACTS

F29: **Erdős-Stone theorem** [ErSt46] For $\varepsilon > 0$ and $2 \leq r \leq n$, the function $g(n, r, \varepsilon)$ tends to infinity when n tends to infinity. Since $T_r(n)$ is r-chromatic, this implies the result of Erdős and Simonovits [ErSi66] that if $F = \{F_1, \ldots, F_k\}$ is a fixed family of graphs and $r + 1 = \min_i \chi(F_i) \geq 2$, then

$$ex(n, F) = \frac{r-1}{2r} n^2 + o(n^2)$$

F30: **Bollobás and Erdős** [BoEr73] There exist constants $c_1, c_2 > 0$ such that

$$c_1 \log n \leq g(n, r, \varepsilon) \leq c_2 \log n.$$

F31: **Bollobás, Erdős and Simonovits** [BES76] There exists $\alpha > 0$ such that if $0 < \varepsilon < 1/r$ then

$$g(n, r, \varepsilon) \geq \frac{\alpha \log n}{r \log \frac{1}{\varepsilon}}.$$

There exists $\varepsilon_r > 0$ such that if $0 < \varepsilon < \varepsilon_r$ then

$$g(n, r, \varepsilon) \leq 3 \frac{\log n}{\log \frac{1}{\varepsilon}}.$$

F32: **Chvátal and Szemerédi** [ChSz83]

$$g(n, r, \varepsilon) \geq \frac{\log n}{500 \log \frac{1}{\varepsilon}}$$

F33: **Bollobás and Kohayakawa** [BoKo94] There exists an absolute constant $\alpha > 0$ such that if

$$r \geq 2, \quad 0 < \gamma < 1, \quad 0 < \varepsilon < 1/r$$

then every graph G of sufficiently large order n with $e(G) \geq (1 - 1/r + \varepsilon)n^2$ contains a $K_{r+1}(s, m, \ldots, m, l)$ such that

$$s \geq \alpha(1-\gamma)\frac{\log n}{r \log \frac{1}{\varepsilon}}, \quad m \geq \alpha(1-\gamma)\frac{\log n}{\log r}, \quad l \geq \alpha \varepsilon^{1+\gamma/2} n^{\gamma}$$

F34: **Ishigami** [Ish02] There exists an absolute constant $\beta > 0$ such that if

$$r \geq 2, \quad 0 < \gamma < 1, \quad \text{and} \quad 0 < \varepsilon < 1/r$$

then every graph G of sufficiently large order n with $e(G) \geq (1 - 1/r + \varepsilon)n^2$ contains a $K_{r+1}(s, m, \ldots, m, l)$ such that

$$s \geq \beta(1-\gamma)\frac{\log n}{\log \frac{1}{\varepsilon}}, \quad m \geq \beta(1-\gamma)\frac{\log n}{\log r}, \quad \text{and} \quad l \geq n^{\gamma}$$

The Structure of Extremal Graphs

The structure of the extremal graphs is fairly well understood in case of a fixed family of forbidden graphs. Moreover, the stability theorems of Erdős and Simonovits give useful information about the structure of a graph without forbidden subgraphs, provided the size is close to the maximum.

FACTS

F35: *Stability theorem.* **Simonovits** [Sim68] For every r there is some $c = c(n)$ such that if $l \leq cn$ then every K_{r+1}-free graph $G(n, t_r(n) - l)$ is r-chromatic.

F36: *Stability theorem.* **Erdős** [Erd68], **Simonovits** [Sim68] Let H be a graph with $\chi(F) = r + 1 \geq 3$. For every $\varepsilon > 0$ there exists $\delta > 0$ such that if G is a H-free graph of order n and $e(G) > ((r-1)/2r - \delta)n^2$ then there is a $K_r(n_1, \ldots, n_r)$ with $n_1 + \ldots + n_r = n$ that can be obtained from G by changing less than εn^2 edges of G.

F37: **Erdős** [Erd68], **Simonovits** [Sim68] Let $F = \{F_1, \ldots, F_k\}$ be a fixed graph family with $r + 1 = \min_i \chi(F_i) \geq 3$ and suppose that F_1 has a $(r+1)$-coloring in which one of the color classes contains t vertices. Then

$$ex(n, F) \;=\; \left(\frac{(r-1)}{2r}\right) n^2 + O(n^{2-1/t})$$

If $G \in \Phi(n, F)$ is such that $e(G) = ex(n, F)$, then $\delta(G) = (1 - 1/r)n + o(n)$ and all of the following hold:

(i) the vertices of G can be partitioned into r classes each of size $n/r + o(n)$;

(ii) each vertex is joined to at most as many vertices of its own class as to any other class. For every $\varepsilon > 0$ the number of vertices joined to at least εn vertices of their own class is $o(n)$;

(iii) there are $O(n^{2-1/t})$ edges joining vertices of the same class.

F38: **Simonovits** [Sim68] Let n and s be fixed integers. If n is sufficiently large then every graph $G = G(n)$ with

$$e(G) \;\geq\; t_r(n-s-1) + (s-1)(n-s-1) + \binom{s-1}{2}$$

contains sK_{r+1}, unless $G = K_{s-1} + T_r(n-s-1)$.

8.1.4 Zarankiewicz Problem and Related Questions

The problem of Zarankiewicz is the counterpart of Turán's theorem for bipartite graphs; this problem has turned out to be extremely difficult — it seems that even today we are very far from a satisfactory solution.

NOTATION

NOTATION: Let $z(m,n,s,t)$ denote the largest size of an n-by-m bipartite graph not containing the complete biparatite graph $K_2(s,t)$, and set $z(n,t) = z(n,n,t,t)$.

FACTS

F39: $2ex(n, \{K_2(s,t)\}) \;\le\; z(m,n,s,t) \;\le\; ex(2n, \{K_2(s,t)\})$

F40: **Kövary, Sós and Turán** [KST54] If $2 \le s \le m$ and $2 \le t \le n$, then

$$z(m,n,s,t) \le (s-1)^{1/t}(n-t+1)m^{1-1/t} + (t-1)m$$

and

$$z(n,t) \le (t-1)^{1/t}n^{2-1/t} + O(n)$$

F41: **Reiman** [Rei58]

(i) $z(n,2) \le (n/2)(1+\sqrt{4n-3})$;

(ii) for every $n = q^2+q+1$, where q is a power of a prime,

$$z(n,2) \le \frac{n}{2}(1+\sqrt{4n-3}) = (q-1)(q^2+q+1)$$

(iii) $\lim_{n\to\infty} z(n,2)n^{-3/2} = 1$.

F42: **Erdős, Rényi and Sós** [ERS66], **Brown** [Bro66] Let q be a power of a prime. Then for the cycle $C_4 = K(2,2)$, we have

$$\frac{1}{2}q(q+1)^2 \le ex(q^2+q+1, \{C_4\}) \le \frac{1}{2}q(q+1)^2 + \frac{q+1}{2} \qquad (3)$$

and

$$\lim_{n\to\infty} \frac{ex(n,\{C_4\})}{n^{3/2}} = \frac{1}{2}$$

The ***Erdős-Rényi graph*** giving the lower bound in (3) has for vertices the q^2+q+1 points of the projective plane $PG(2,q)$ over the field of order q, and two points (x_1,y_1,z_1) and (x_2,y_2,z_2) are joined if and only if $x_1x_2 + y_1y_2 + z_1z_2 = 0$.

F43: **Füredi** [Fur83] For every natural number q,

$$ex(q^2+q+1, \{C_4\}) \le \frac{1}{2}q(q+1)^2$$

and if q is a power of a prime, then

$$ex(q^2+q+1, \{C_4\}) = \frac{1}{2}q(q+1)^2$$

$\mathbf{K_r}$-Free Graphs with Large Minimal Degree

In 1973 Erdős and his collaborators initiated the study of K_r-free graphs with large minimal degree. It turned out that under certain conditions, the chromatic number of such graphs is bounded, and as later investigations showed, their structure is well determined. Despite intensive efforts the general questions remain open, the most challenging of which is the conjecture that every K_3-free $G = G(n)$ with $\delta(G) > n/3$ is at most 4-chromatic.

DEFINITIONS

D11: We say that a graph G is ***homomorphic*** to a graph H if there exists a mapping $f : V(G) \to V(H)$ such that $(u, v) \in E(G)$ implies that $(f(u), f(v)) \in E(H)$.

D12: The ***kth power of a cycle*** C_n is a graph G with $V(G) = [n]$ and $(i, j) \in E(G)$ if and only if $|i - j| = 1, 2, \dots, k \bmod n$.

D13: The ***square of a cycle*** is its second power.

EXAMPLES

E3: ***Mycielski graphs*** [Myc55] Define the sequence of graphs $M_1, M_2, \dots$ as follows: set $M_1 = K_2$; suppose that M_{s-1} is already defined, and let $V(M_{s-1}) = [n]$. Set $V(M_s) = [2n + 1]$, and let $E(M_s)$ be the union

$$E(M_{s-1}) \cup \{(i, j + n) : (i, j) \in E(M_{s-1})\} \cup \{(2n + 1, i) : n < i \leq 2n\}$$

For every i, the graph M_i is K_3-free and $\chi(M_i) = i$. In particular, the graph M_2 is C_5; the graph M_3 is also known as the ***Grötzsch graph***.

E4: ***Andrásfai graphs*** [And62] Set $A_1 = K_2$ and for every $i \geq 2$ let A_i be the complement of the $(i - 1)$th power of C_{3i-1}. For every i, the graph A_i is K_3-free and $\chi(M_i) = 3$. In particular, the graph A_2 is C_5; the graph A_3 is also known as the ***Möbius ladder***.

FACTS

F44: **Andrásfai, Erdős and Sós** [AES74] If a graph $G = G(n)$ is K_{r+1}-free with minimal degree

$$\delta(G) \geq \left(1 - \frac{3}{3r - 1}\right) n$$

then G is r-chromatic.

F45: **Erdős-Hajnal-Simonovits graphs** [ErSi73] If $\varepsilon > 0$, $h > 2$ and n is sufficiently large, then there exists a K_3-free graph $G(n)$ with $\delta(G) \geq (1/3 - \varepsilon)n$ and $\chi(G) \geq h$.

F46: **Häggkvist** [Hag82] Every K_3-free graph $G = G(n)$ with $\delta(G) > 3n/8$ is homomorphic to $A_2 = C_5$ and so satisfies the inequality $\chi(G) \leq 3$.

F47: **Häggkvist** [Hag82] For every natural number k, there exists a 4-chromatic, K_3-free, $10k$-regular graph of order $29k$.

F48: **Jin** [Jin95] Every K_3-free $G = G(n)$ with $\delta(G) > 10n/29$ is homomorphic to the graph A_9 and so satisfies the inequality $\chi(G) \leq 3$.

F49: **Chen, Jin and Koh** [CJK97] Every K_3-free 3-chromatic $G = G(n)$ with $\delta(G) > n/3$ is homomorphic to some graph A_i. If $\chi(G) \geq 4$, then $M_3 \subset G$.

F50: **Brandt** [Bra00] If $G(n)$ is d-regular maximal K_3-free graph with $d > n/3$, then $\chi(G) \leq 4$.

F51: **Häggkvist and Jin** [HaJi98] Let $G = G(n)$ be K_3-free and C_5-free. If $\delta(G) > n/4$, then G is homomorphic to C_7. The bound $n/4$ is best possible, as there is a K_3-free, C_5-free, $3k$-regular $G(12k)$ that is not homomorphic to C_7.

8.1.5 Paths and Trees

One of the most famous unsolved problems in extremal graph theory is the **Erdős-Sós conjecture**:

Every graph $G(n, \lfloor (k-1)n/2 \rfloor + 1)$ contains all trees of order k.

The conjecture is true for many types of trees and under special conditions for the graph, but the general case remains open.

NOTATION

NOTATION: P_k denotes the path of order k.

FACTS

F52: **Erdős-Gallai theorem** [ErGa59] Every $G = G(n)$ with $e(G) > (n-1)k/2$ contains a P_{k+1}. If $n = q(k-1)+1$, then there exists a graph $G = G(n, (n-1)k/2)$ containing no P_{k+1}.

F53: **Faudree and Schelp** [FaSc75] If $n = kt + r$ and $0 \leq r < t$, then

$$ex(n, P_{k+1}) = t\binom{k}{2} + \binom{r}{2}$$

and the extremal graphs are known.

F54: **Brandt and Dobson** [BrDo96] Every graph $G(n, \lfloor (k-1)n/2 \rfloor + 1)$ of girth at least 5 contains all trees of order k.

F55: **Saclé and Woźniak** [SaWo97] Every C_4-free graph $G(n, \lfloor (k-1)n/2 \rfloor + 1)$ contains all trees of order k.

F56: **Wang, Li and Liu** [WLL00] Every graph $G(n, \lfloor (k-1)n/2 \rfloor + 1)$ whose edge-complement is of girth at least 5 contains all trees of order k.

F57: **Dobson** [Dob02] Every graph $G(n, \lfloor (k-1)n/2 \rfloor + 1)$ whose edge-complement is $K(2,4)$-free contains all trees of order k.

8.1.6 Circumference

DEFINITIONS

D14: The ***circumference*** of a graph G is the length of its largest cycle. It is denoted by $c(G)$.

D15: The ***girth*** of a graph G is the length of its smallest cycle.

NOTATIONS

NOTATION: The set of cycle lengths of a graph G is denoted by $C(G)$.

NOTATION: $ec(G)$ is the largest even number in $C(G)$ and $oc(G)$ is the largest odd number in $C(G)$.

FACTS

F58: **Erdős and Gallai** [ErGa59] If $2 \leq k \leq n$ then

$$c(G(n, \lfloor (k-1)(n-1)/2 \rfloor + 1)) \geq k$$

F59: **Bollobás and Häggkvist** [BoHa90] If $G = G(n)$ and $\delta(G) \geq n/k$, then $c(G) \geq \lceil n/(k-1) \rceil$ and this inequality is best possible.

F60: **Egawa and Miyamoto** [EgMi89] If $G = G(n)$ and if $d(u) + d(v) \geq \lceil 2n/k \rceil$ whenever u and v are two nonajacent vertices, then $c(G) \geq \lceil n/(k-1) \rceil$ and this inequality is best possible.

F61: **Dirac** [Dir52] If $G = G(n)$ is 2-connected with $\delta(G) = \delta \leq n/2$, then $c(G) \geq 2\delta$.

F62: **Voss and Zuluaga** [VoZu77] If $G = G(n)$ is a 2-connected, nonbipartite graph with $\delta(G) = \delta \leq n/2$, then

$$oc(G) \geq 2\delta - 1, \quad ec(G) \geq 2\delta$$

8.1.7 Hamiltonian Cycles

The theory of Hamiltonian graphs is one of the most popular areas of graph theory. Here we present several well-known results with an “extremal” flavor.

DEFINITIONS

D16: A graph $G = G(n)$ is said to be ***Hamiltonian*** if $n \in C(G)$.

D17: The ***closure*** of a graph $G(n)$ is obtained by successively joining every two nonadjacent vertices u and v with $d(u) + d(v) \geq n$.

FACTS

F63: **Dirac's theorem** [Dir52] If $G = G(n)$, $n \geq 3$, and $\delta(G) \geq n/2$, then G is Hamiltonian.

F64: **Shi** [Shi92], **Bollobás and Brightwell** [BoBr93] Let $G = G(n)$, and let S be the set of vertices of degree at least $n/2$. If $|S| \geq 3$, then there is a cycle in G that includes every vertex of S.

F65: **Ore's theorem** [Ore60] If $G = G(n)$ with $n \geq 3$, and if $d(u)+d(v) \geq n$ whenever u and v are two nonajacent vertices, then G is Hamiltonian.

F66: **Pósa's Theorem** [Pos62] Let $G = G(n)$ with $n \geq 3$. If for every k with $1 \leq k < (n-1)/2$, the number of vertices of G of degree not exceeding k is less than k, and for odd n the number of vertices of degree $(n-1)/2$ does not exceed $(n-1)/2$, then G is Hamiltonian.

F67: **Closure Lemma of Bondy and Chvátal** [BoCv76] A graph is Hamiltonian if and only if its closure is Hamiltonian.

F68: **Chvátal's theorem** [Chv72] Let $G = G(n)$ with $n \geq 3$ and with vertex degrees $d(1) \leq \ldots \leq d(n)$. If for every $k \leq (n-1)/2$ either $d(k) > k$ or $d(n-k) \geq n-k$, then G is Hamiltonian.

F69: **Chvátal-Erdős theorem** [ChEr72] If $\alpha(G) \leq \kappa(G)$, then G is Hamiltonian.

F70: **Fan and Häggkvist** [FaHa94] If $G = G(n)$ and $\delta(G) \geq 5n/7$, then G contains the square of C_n.

8.1.8 Cycle Lengths

Erdős proposed the sum $\sum\{1/r : r \in C(G)\}$ as a measure of the wealth of cycle lengths in a graph G. He stated a conjecture that led to the following two results.

FACTS

F71: **Gyárfás, Komlós, and Szemerédi** [GKS84] There exists a number $c > 0$ such that for every graph $G = G(n, m)$ we have

$$\sum\{1/r : r \in C(G)\} \geq c\log(2m/n)$$

F72: **Gyárfás, Prömel, Szemerédi, and Voigt** [GPSV85] If k is sufficiently large and $2m \geq (1 + 1/k)n$, then

$$\sum\{1/r : r \in C(G)\} \geq (300k \log k)^{-1}$$

Cycles of Consecutive Lengths

In this section we present several sufficient conditions in terms of the size and minimal degree for the existence of large intervals in the set $C(G)$ of cycle lengths.

FACTS

F73: **Bondy-Simonovits theorem** [BoSi74] Every graph $G(n, \lfloor 100kn^{1/k} \rfloor + 1)$ contains the cycle C_{2l} for $k \leq 2l \leq kn^{1/k}$.

F74: **Verstraëte** [Ver00] Every graph $G(n, \lfloor 8(k-1)n^{1/k} \rfloor + 1)$ contains the cycle C_{2l} for $k \leq 2l \leq kn^{1/k}$.

F75: **Genghua Fan** [Fan02] If G is a graph with $\delta(G) \geq 3k$, then G contains $k+1$ cycles $C_0, C_1, \ldots, C_k$ such that

$$k+1 < |C_0| < \ldots < |C_k|, \quad |C_i| - |C_{i-1}| = 2, (1 \leq i \leq k)$$

and $|C_k| - |C_{k-1}| \leq 2$.

F76: **Gould, Haxell, and Scott** [GHS02] For every $c > 0$ there exists a constant $k = k(c)$ such that if $G = G(n)$ and $\delta(G)cn$, then G contains a cycle of order t for every even $t \in [4, ec(G) - k]$ and for every odd $t \in [k, oc(G) - k]$.

Pancyclicity and Weak Pancyclicity

In 1971 Bondy introduced the concept of pancyclicity that soon became a topic of intensive study. We present below only few of the known results.

DEFINITIONS

D18: A graph $G = G(n)$ is called ***weakly pancyclic*** if $C(G)$ is an interval.

D19: A graph $G = G(n)$ is called ***pancyclic*** if $C(G) = [3, n]$.

FACTS

F77: **Bondy** [Bon71] If $G = G(n, \lfloor n^2/4 \rfloor)$ is Hamiltonian, then G is pancyclic unless $G = K(\lfloor n/2 \rfloor, \lceil n/2 \rceil)$.

F78: **Bondy** [Bon71] If $G = G(n, \lfloor n^2/4 \rfloor + 1)$, then $c(G) \geq \lfloor (n+3)/2 \rfloor$ and G is weakly pancyclic.

F79: **Amar, Flandrin, Fournier, and Germa** [AFFG83] If $n \geq 102$, $G = G(n)$ is Hamiltonian and $\delta(G) > 2n/5$, then G is pancyclic.

F80: **Shi** [Shi86] If $n > 50$, $G = G(n)$ is Hamiltonian and for every two nonadjacent vertices u and v, $d(u) + d(v) > 4n/5$, then G is pancyclic.

F81: **Brandt, Faudree and Goddard** [BFG98] If $\delta(G) \geq n/4 + 250$, then the graph G is weakly pancyclic unless the order of the shortest odd cycle of G is 7, in which case $C(G) = \{4, 6, 7, \ldots, c(G)\}$.

F82: **Brandt, Faudree, and Goddard** [BFG98] If G is a 2-connected nonbipartite graph of sufficiently large order n with $\delta(G) > 2n/7$, then G is weakly pancyclic.

F83: **Brandt** [Bra97] Every $G(n, \lfloor (n-1)^2/4 \rfloor + 2)$ is weakly pancyclic or bipartite.

F84: **Bollobás and Thomason** [BoTh99] Every graph $G(n, \lfloor n^2/4 \rfloor - n + 59)$ is weakly pancyclic or bipartite.

8.1.9 Szemerédi's Uniformity Lemma

The Uniformity Lemma of Szemerédi, whose power and versatility could hardly be overemphasized, is one of the most remarkable tools in discrete mathematics. Loosely stated, it guarantees that every dense graph has some finite rough structure, which, surprisingly often, is the basis of successful attacks on difficult combinatorial problems. The Blow-up Lemma of Komlós and Sárközy and Szemerédi, a close relative of the Uniformity Lemma, has been used to solve a number of difficult graph embedding conjectures. For comprehensive surveys of this area see [KoSi96], [Kom99], and [Kom00].

DEFINITIONS

D20: Let $a > 0$. A ***tower of*** a ***of length*** k is the function

$$a^{a^{\cdot^{\cdot^{a}}}}$$

where the exponentiation is done k times.

D21: A bipartite graph with classes A, B is called an ***ε-uniform pair*** if for every pair of vertex subsets $X \subset A$ and $Y \subset B$ with $|X| > \varepsilon\,|A|$ and $|Y| > \varepsilon\,|B|$ we have

$$\left|\frac{E(X,Y)}{|X|\,|Y|} - \frac{E(A,B)}{|A|\,|B|}\right| < \varepsilon$$

D22: A bipartite graph with classes A, B is called an ***(ε,δ)-super-uniform pair*** if it is ε-uniform and

$$d(u) \geq \delta\,|B|\,, \ldots, d(v) \geq \delta\,|A|$$

whenever every $u \in A$, $v \in B$.

FACTS

F85: **Szemerédi's Uniformity Lemma** [Sze76] For every $\varepsilon > 0$ there exist numbers $n_0 = n_0(\varepsilon)$ and $k_0 = k_0(\varepsilon)$ such that for every graph G of order $n > n_0$ there is a partition $V(G) = V_0 \cup V_1 \cup \ldots \cup V_k$ satisfying these criteria:

(i) $k \leq k_0(\varepsilon)$;

(ii) $|V_0| < \varepsilon n$, $|V_1| = \ldots = |V_k|$;

(iii) all but εk^2 pairs (V_i, V_j) are ε-uniform.

F86: The function $k_0(\varepsilon)$ in Szemerédi's Uniformity Lemma is bounded from above by a tower of 2s of length ε^{-5}.

F87: **Gowers' bound** [Gow98] There exist constants $\varepsilon_0 > 0$ and $c > 0$ such that for $0 < \varepsilon \leq \varepsilon_0$, there is a graph G whose vertices cannot be partitioned according to criteria (i)-(iii) of Szemerédi's Uniformity Lemma unless k is as large as a tower of 2s of length $c\varepsilon^{-1/16}$.

F88: **Blow-up Lemma, Komlós, Sárközy, and Szemerédi** [KSS97] Fix a graph R with $V(R) = [r]$. For every $\delta, \Delta > 0$ there exists $\varepsilon > 0$ such that the following holds. Fix a natural n and let $V_1, \ldots, V_r$ be r disjoint sets of size n. Define the graphs $R(n)$ and G as follows:

(i) Set $V(R(n)) = \cup V_i$ and for every $(i, j) \in E(R)$, place all edges between V_i and V_j.

(ii) Set $V(G)) = \cup V_i$ and for every $(i, j) \in E(R)$, place an (ε, δ)-super-uniform pair between V_i and V_j.

If $H \subset R(n)$ and $\Delta(H) \leq \Delta$ then $H \subset G$.

Applications of the Uniformity and Blow-up lemmas

The Uniformity Lemma and the Blow-up Lemma are powerful tools in graph theory. We present below only four of their applications but, in fact, many results described in other sections are also obtained applying these two lemmas.

FACTS

F89: **Komlós, Sárközy, and Szemerédi** [KSS98] For every $\varepsilon > 0$ and natural number k there exists a number $n_0 = n_0(k)$, such that if

$$n > n_0, \quad G = G(n), \quad \text{and} \quad \delta(G) \geq \left(1 - \frac{1}{k+1}\right) n$$

then the graph G contains the kth power of a Hamiltonian cycle.

F90: **Alon and Yuster** [AlYu96] For every $\varepsilon > 0$ and natural number h there exists a number $n_0 = n_0(\varepsilon, h)$, such that if H is a graph of order h with $\chi(H) = k$ and

$$n > n_0, \quad G = G(hn), \quad \text{and} \quad \delta(G) \geq \left(1 - \frac{1}{k} + \varepsilon\right) hn$$

then the graph G contains a H-factor.

F91: **Komlós, Sárközy, and Szemerédi** [KSS01] Let H be a graph of order h and $\chi(H) = k$. There exist numbers $c = c(H)$ and $n_0 = n_0(H)$ such that if

$$n > n_0, \quad G = G(hn), \quad \text{and} \quad \delta(G) \geq \left(1 - \frac{1}{k}\right) hn + c$$

then G contains a H-factor.

F92: **Komlós, Sárközy, and Szemerédi** [KSS01a] For every $\varepsilon > 0$, there exist numbers c and n_0 such that for $n > n_0$, every graph $G = G(n)$ with $\delta(G) > (1/2 + \varepsilon) n$ contains every tree T of order n such that $\Delta(T) < cn/\log n$.

8.1.10 Asymptotic Enumeration

An intriguing question is how many graphs with given properties are there. For certain natural properties like "G is K_r-free" or "G has no induced subgraph isomorphic to H" satisfactory answers have been obtained.

DEFINITIONS

D23: A ***graph property*** is a graph family closed under isomorphism.

D24: A graph property P is called ***monotone*** if $G \in P$ implies $H \in P$ for every subgraph of G.

D25: A graph property P is called ***hereditary*** if $G \in P$ implies $H \in P$ for every induced subgraph of G.

D26: For any graph property P, set $P^n = \{G : G \in P, v(G) = n\}$. The ***logarithmic density*** of P^n is the value

$$c_n(P) = (\log_2 |P^n|)/\binom{n}{2}$$

D27: Let $0 \leq s \leq r$ be integers. A graph H is called (r, s)***-colorable*** if its vertices can be colored in r colors, so that the vertices colored with the ith color are a clique for $1 \leq i \leq s$, and an independent set otherwise.

D28: The ***coloring number*** $r(P)$ of a hereditary property P is the largest integer such that for some s, the family P contains every (r, s)-colorable graph.

NOTATION

NOTATION: Given a hereditary property P, let $ex_{ind}(n, P)$ denote the maximal number of edges in a graph $G_0 = G(n)$ for which there is a graph $G_1 = G(n)$ with $V(G_1) = V(G_0)$ and $E(G_1) \cap E(G_0) = \emptyset$, so that every graph G with $G_1 \subseteq G \subseteq G_0 \cup G_1$ belongs to P^n.

FACTS

F93: **Erdős, Kleitman, and Rothschild** [EKR73] The number k_n of the K_{r+1}-free graphs of order n is given asymptotically by

$$\log_2 k_n = \left(1 - \frac{1}{r} + o(1)\right) n^2$$

F94: **Erdős, Frankl, and Rödl** [EFR86] Let H be a graph with $\chi(H) = r + 1$. The number h_n of the H-free graphs of order n is given asymptotically by

$$\log_2 h_n = \left(1 - \frac{1}{r} + o(1)\right) n^2$$

F95: **Kolaitis, Prömel, and Rothschild** [KPR87] For every n let k_n be the number of K_{r+1}-free graphs $G(n)$ and h_n the number of r-chromatic graphs $G(n)$. Then

$$\lim_{n \to \infty} \frac{k_n}{h_n} = 1$$

F96: A property P is monotone if and only if there exists some sequence of graphs $F_1, F_2, \ldots$ such that P is the collection of graphs having no subgraph isomorphic to an F_i.

F97: A property P is hereditary if and only if there exists some sequence of graphs $F_1, F_2, \ldots$ such that P is the collection of graphs having no induced subgraph isomorphic to an F_i. The coloring number of $r(P)$ is exactly the maximal r such that for some $0 \leq s \leq r$ no F_i is (r, s)-colorable.

F98: **Prömel and Steger** [PrSt91] There exist numbers c_0 and c_1 such that the number t_n of $G(n)$ with no induced C_4 is given by

$$t_n = (c_r + o(1))2^{n^2/4+n-(\log n)/2}$$

where $r = 0, 1$, $r = n \bmod 2$.

F99: **Prömel and Steger** [PrSt92], [PrSt93] Fix a graph H, and let P be the hereditary property "G has no induced subgraph isomorphic to H". Then

$$\lim_{n\to\infty} ex_{ind}(n, P)\binom{n}{2}^{-1} = 1 - \frac{1}{r(P)}$$

F100: **Alekseev** [Ale92], **Bollobás and Thomason** [BoTh95] Let P be a hereditary property. Then

$$1 = c_1(P) \;\geq\; \ldots \;\geq\; c_n(P) \;\geq\; \ldots$$

and the limit

$$c(P) = \lim_{n\to\infty} c_n(P)$$

exists.

F101: **Scheinerman and Zito** [ScZi94] For every hereditary property P, one of the following is true:

(i) for n sufficiently large $|P^n|$ is identically $0, 1$ or 2;

(ii) $|P^n| = \Theta(1)n^k$ for some integer $k \geq 1$;

(iii) for some $c_2 \geq c_1 > 0$, $c_1^n \leq |P^n| \leq c_2^n$;

(iv) for some $c > 0$, $|P^n| \geq n^{cn}$.

F102: **Bollobás and Thomason** [BoTh97] For every hereditary property P,

$$c(P) \;=\; \lim_{n\to\infty} ex_{ind}(n, P)\binom{n}{2}^{-1} \;=\; 1 - \frac{1}{r(P)}$$

F103: **Balogh, Bollobás, and Weinreich** [BBW00], [BBW01], [BBW02] For every hereditary property P one of the following is true:

(i) there exists a collection of polynomials $\{p_i(n)\}_{i=0}^k$ such that for n sufficiently large $|P^n| = \sum_{i=0}^k p_i(n)i^n$;

(ii) for some integer $k > 1$, $|P^n| = n^{(1-1/k+o(1))n}$;

(iii) $n^{(1+o(1))n} \leq |P^n| \leq n^{o(n^2)}$;

(iv) for some integer $k > 1$, $|P^n| = n^{(1-1/k+o(1))n^2/2}$.

8.1.11 Graph Minors

The study of graph minors was initially motivated by the conjecture of Hadwiger that every r-chromatic graph has K_r as a minor. However, from the extremal point of view, minors happen to be of their own fascinating interest.

DEFINITIONS

D29: Let G and H be graphs. We say that H is a ***minor*** of G, and we write

$$G \succ H$$

if there are disjoint sets $W(u)$, $u \in V(H)$, such that $W(u)$ induces a connected graph in G, and for every $(u,v) \in E(G)$, there is an edge between $W(u)$ and $W(v)$.

D30: Let $\mu(H)$ be the minimal number μ such that $e(G) \geq \mu v(G)$ implies that $G \succ H$.

FACTS

F104: **Mader** [Mad67], [Mad68]

$$\mu(K_r) \leq 8r \log_2 r$$

F105: **Bollobás, Catlin, and Erdős** [BCE80], **de la Vega** [Fer83] For some $C > 0$,

$$\mu(K_r) \geq Cr\sqrt{\log r}$$

F106: **Kostochka** [Kos82], [Kos84], **Thomason** [Tho84]

$$\mu(K_r) = O(r\sqrt{\log r})$$

F107: **Thomason** [Tho01] There is an explicit constant $\alpha = 0.319\ldots$ such that

$$\mu(K_r) = (\alpha + o(1))r\sqrt{\log r}$$

F108: **Myers and Thomason** [MyTh02] Given a graph H of order n, set

$$\gamma(H) = \min_w \frac{1}{n} \sum_{u \in V(H)} w(u) \quad \text{with} \quad \sum_{(u,v)\in E(H)} n^{-w(u)w(v)} = n$$

where $w(u)$ are nonnegative real numbers assigned to the vertices of H. Then

$$\mu(H) = (\alpha\gamma(H) + o(1))r\sqrt{\log r}$$

8.1.12 Ramsey-Turán Problems

Ramsey-Turán problems are in fact Turán-type problems with with restriction on the independence number. For a comprehensive survey of this topic see [SiSo01]; we present below only some of the highlights of the area.

NOTATIONS

NOTATION: Let $F_1, \ldots, F_s$ be fixed graphs. Let $RT_s(n, F_1, \ldots, F_s, f(n))$ denote the maximal size of a graph $G(n)$ with $\alpha(G) \leq f(n)$ whose edges can be colored in s colors so that there is no F_i in the i^{th} color. We write $RT(n, F_1, \ldots, F_s, f(n))$ instead of $RT_s(n, F_1, \ldots, F_s, f(n))$ when s is understood.

NOTATION: Let $R(s)$ is the maximal number R such that one can color the edges of the complete graph K_R in s colors, so that there is no monochromatic triangle and so that each star is colored in at most $(r-1)$ colors.

FACTS

F109: **Erdős graph** [Erd61] For every k there exists $\varepsilon > 0$ such that if n is sufficiently large there exists a graph $F_{n,k} = G(n)$ with girth $g(G) > k$ and independence number $\alpha(G) < n^{1-\varepsilon}$.

F110: **Erdős and Sós** [ErSo70]

$$RT(n, K_{2r+1}, o(n)) = \frac{r-1}{2r} n^2 + o(n^2)$$

The lower bound comes from the following graph: take the Turán graph $T_r(n)$ and add to each of its classes a copy of $F_{s,3}$, where s is the size of the class.

F111: **Bollobás and Erdős** [BoEr76a] For every $\varepsilon > 0$ and n sufficiently large, there exists a K_4-free graph $BE_n = G(n)$ with $\alpha(BE_n) \leq \varepsilon n$ and $|d(u) - n/4| < \varepsilon n$ for every $u \in V(BE_n)$. Thus,

$$RT(n, K_4, o(n)) \geq \frac{1}{8} n^2 + o(n^2)$$

F112: **Szemerédi** [Sze72]

$$RT(n, K_4, o(n)) \leq \frac{1}{8} n^2 + o(n^2)$$

F113: **Erdős, Hajnal, Sós, and Szemerédi** [EHSS83]

$$RT\left(n, 2r, o(n)\right) = \frac{3r-5}{6r-4} n^2 + o(n^2)$$

To prove the lower bound consider the following graph: for $l = \lceil 4n/(3r-2) \rceil$ take $BE_l + T_{r-l}(n-l)$, and add to each of the parts of $T_{r-l}(n-l)$ a copy of $F_{s,3}$, where s is the size of the part.

F114: **Erdős and Sós** [ErSo70]

$$RT(n, K_3, K_3, o(n)) = \frac{1}{4} n^2 + o(n^2)$$

F115: **Erdős, Hajnal, Sós, and Szemerédi** [EHSS83]

$$RT_s(n, K_3, \ldots, K_3, o(n)) = \frac{R(s)-1}{2R(s)} n^2 + o(n^2)$$

F116: **Erdos, Hajnal, Simonovits, Sós, and Szemerédi** [EHSSS93]

$$RT_s(n, K_3, K_4, o(n)) = \frac{1}{2}\left(1 - \frac{1}{3}\right) n^2 + o(n^2)$$

$$RT_s(n, K_3, K_5, o(n)) = \frac{1}{2}\left(1 - \frac{1}{5}\right) n^2 + o(n^2)$$

If p and q are odd integers then

$$RT_s(n, C_p, C_q, o(n)) = \frac{1}{4} n^2 + o(n^2)$$

References

[AES74] B. Andrásfai, P. Erdős and V. T. Sós, On the connection between chromatic number, maximal clique and minimal degree of a graph, *Discrete Math.* **8** (1974), 205–218.

[AFFG83] D. Amar, E. Flandrin, I. Fournier and A. Germa, Pancyclism in Hamiltonian graphs, *Discrete Math.* **89** (1991), 111–131.

[Ale92] V. E. Alekseev, Range of values of entropy of hereditary classes of graphs (Russian), *Diskret. Mat.* **4** (1992), 148–157.

[AlYu96] N. Alon and R. Yuster, H-factors in dense graphs, *J. Combin. Theory, Ser. B* **66** (1996), 269–282.

[And62] B. Andrásfai, Über ein Extremalproblem der Graphentheorie (German), *Acta Math. Acad. Sci. Hungar.* **13** (1962), 443–455.

[BBW00] J. Balogh, B. Bollobás and D. Weinreich, The speed of hereditary properties of graphs. *J. Combin. Theory, Ser. B* **79** (2000), 131–156.

[BBW01] J. Balogh, B. Bollobás and D. Weinreich, The penultimate rate of growth for graph properties, *European J. Combin.* **22** (2001), 277–289.

[BBW02] J. Balogh, B. Bollobás and D. Weinreich, Measures on monotone properties of graphs, *Discrete Appl. Math.* **116** (2002), 17–36.

[BCE80] B. Bollobás, P. Catlin and P. Erdős, Hadwiger's conjecture is true for almost every graph, *European J. Combin.* **1** (1980), 195–199.

[BES76] B. Bollobás, P. Erdős and M. Simonovits, On the structure of edge graphs II. *J. London Math. Soc.* **12** (1976), 219–224.

[BFG98] S. Brandt, R. Faudree and W. Goddard, Weakly pancyclic graphs, *J. Graph Theory* **27** (1998), 141–176.

[BoBr93] B. Bollobás and G. Brightwell, Cycles through specified vertices, *Combinatorica* **13** (1993), 147–155.

[BoEr73] B. Bollobás and P. Erdős, On the structure of edge graphs, *J. London Math. Soc.* **5** (1973), 317–321.

[BoEr76] B. Bollobás and P. Erdős, On a Ramsey-Turán type problem, *J. Combin. Theory, Ser. B* **21** (1976), 166–168.

[BoEr76a] B. Bollobás and P. Erdős, Unsolved problems, *Proc. Fifth Brit. Comb. Conf. (Univ. Aberdeen, Aberdeen, 1975)*, Winnipeg, Util. Math. Publ., 678–680.

[BoHa90] B. Bollobás and R. Häggkvist, The circumference of a graph with a given minimal degree, pp. 97–104 in *A tribute to Paul Erdös*, Cambridge University Press, 1990.

[BoKo94] B. Bollobás and Y. Kohayakawa, An extension of the Erdös-Stone theorem, *Combinatorica* **14** (1994), 279–286.

[Bol78] B. Bollobás, *Extremal Graph Theory*, Academic Press, 1978.

[Bol76] B. Bollobás, On complete subgraphs of different orders, *Math. Proc. Cambridge Philos. Soc.* **79** (1976), 19–24.

[Bol98] B. Bollobás, *Modern Graph Theory,* Graduate Texts in Mathematics, **184,** Springer-Verlag, 1998.

[Bol99] B. Bollobás, Turán's theorem and maximal degrees, *J. Combin. Theory, Ser. B* **75** (1999), 160–164.

[Bom97] I. Bomze, Evolution towards the maximum clique, *J. Global Optim.* **10** (1997), 143–164.

[Bon71] J. A. Bondy, Pancyclic graphs I, *J. Comb. Theory, Ser. B* **11** (1971), 80–84.

[Bon83a] J. A. Bondy, Large dense neighbourhoods and Turán's theorem, *J. Combin. Theory, Ser B* **34** (1983), 109–111.

[Bon83b] J. A. Bondy, Erratum: Large dense neighbourhoods and Turán's theorem, *J. Combin. Theory, Ser B* **35** (1983), 80.

[Bon95] J. A. Bondy, Basic graph theory: paths and circuits, *Handbook of Combinatorics*, Vol. **1**, Elsevier, Amsterdam, 1995, pp. 3–110.

[BoSi74] J. A. Bondy and M. Simonovits, Cycles of even length in graphs, *J. Combin. Theory, Ser B* **16** (1974), 97–105.

[BoTh81] B. Bollobás and A. Thomason, Dense neighbourhoods and Turán's theorem, *J. Combin. Theory Ser. B* **31** (1981), 111–114.

[BoTh85] B. Bollobás and A. Thomason, Random graphs of small order, *Ann. Discrete Math.,* Vol 28, pp 47–97, North-Holland, Amsterdam, 1985.

[BoTh95] B. Bollobás and A. Thomason, Projections of bodies and hereditary properties of hypergraphs, *J. London Math. Soc.* **27** (1995), 417–424.

[BoTh97] B. Bollobás and A. Thomason, Hereditary and monotone properties of graphs, *The mathematics of Paul Erdös, II,* pp 70–78, *Algorithms Comb.* **14**, Springer, Berlin, 1997.

[BoTh99] B. Bollobás, A. Thomason, Weakly pancyclic graphs, *J. Combin. Theory, Ser B* **77** (1999), 121–137.

[Bra97] S. Brandt, A sufficient condition for all short cycles, *Discrete Appl. Math.* **79** (1997), 63–66.

[Bra00] S. Brandt, A 4-colour problem for dense triangle-free graphs, *Discrete Math.* **251** (2002), 33–46.

[BrDo96] S. Brandt and E. Dobson, The Erdős-Sós conjecture for graphs of girth 5, *Discrete Math.* **150** (1996), 411–414.

[Bro66] W. G. Brown, On graphs that do not contain a Thomsen graph, *Canad. Math. Bull.* **9** (1966), 281–285.

[CEV88] L. Caccetta, P. Erdős and K. Vijayan, Graphs with unavoidable subgraphs with large degrees, *J. Graph Theory* **12** (1988), 17–27.

[ChEr72] V. Chvátal and P. Erdős, A note on Hamiltonian circuits, *Discrete Math.* **2** (1972), 111–113.

[ChSz83] V. Chvátal and E. Szemerédi, Notes on the Erdös-Stone theorem, *Annals of Discrete Math.* **17** (1983), 207-214.

[Chv72] V. Chvátal, On Hamilton's ideals, *J. Combin. Theory, Ser B* **12** (1972), 163–168.

[CJK97] C. C. Chen, G. P. Jin and K. M. Koh, Triangle-free graphs with large degree, *Combin. Probab. Comput.* **6** (1997), 381–396.

[CoHa63] K. Corradi and A. Hajnal, On the maximal number of independent circuits in a graph, *Acta Math. Acad. Sci. Hungar.* **14** (1963), 423–439.

[Dir52] G. Dirac, Some theorems on abstract graphs, *Proc. London Math. Soc.* (3) **2**, (1952). 69–81.

[Dir63] G. Dirac, Extension of Turán's theorem on graphs, *Acta Math. Acad. Sci. Hungar* **14** (1963), 417–422.

[Dob02] E. Dobson, Constructing trees in graphs whose complement has no $K_{2,s}$, *Combin. Probab. Comput.* **11** (2002), 343–347.

[Edw77] C. Edwards, The largest vertex degree sum for a triangle in a graph, *Bull. Lond. Math. Soc.*, **9** (1977), 203–208.

[Edw78] C. Edwards, Complete subgraphs with largest sum of vertex degrees, *Colloq. Math. Soc. János Bolyai* **18**, North-Holland, Amsterdam-New York, 1978, pp. 293–306.

[EdMS] C. Edwards, A lower bound for the largest number of triangles with a common edge, Manuscript.

[EFR86] P. Erdős, P. Frankl and V. Rödl, The asymptotic number of graphs not containing a fixed subgraph and a problem for hypergraphs having no exponent, *Graphs Combin.* **2** (1986), 113–121.

[EFR94] P. Erdős, R. Faudree and C. Rousseau, Extremal problems and generalized degrees, *Discrete Math.* **127** (1994), 139–152.

[EFG95] P. Erdős, R. Faudree and E. Györi, On the book size of graphs with large minimal degree, *Studia Sci. Math. Hungar.* **30** (1995), 25–46.

[EgMi89] Y. Egawa and T. Miyamoto, The longest cycles in a graph G with minimum degree at least $|G|/k$, *J. Combin. Theory, Ser B* **46** (1989), 356–362.

[EKR73] P. Erdős, D. J. Kleitman and B. L. Rothschild, Asymptotic enumeration of K_n-free graphs, *Colloquio Internazionale sulle Teorie Combinatorie (Rome, 1973), Tomo II*, pp. 19–27. *Atti dei Convegni Lincei, No. 17, Accad. Naz. Lincei*, Rome, 1976.

[EHSS83] P. Erdős, A. Hajnal, V. Sós and E. Szemerédi, More results on Ramsey-Turán type problems, *Combinatorica* **3** (1983), 69–81.

[EHSSS93] P. Erdős, A. Hajnal, M. Simonovits, V. Sós and E. Szemerédi, Turán-Ramsey theorems and simple asymptotically extremal structures, *Combinatorica* **13** (1993), 31–56.

[Erd61] P. Erdős, Graph theory and probability II, *Canad. J. Math.* **13** (1961), 346–352.

[Erd62a] P. Erdős, On the number of complete subgraphs contained in certain graphs, *Publ. Math. Inst. Hung. Acad. Sci.* **VII**, Ser. A3 (1962), 459-464.

[Erd62b] P. Erdős, On a theorem of Rademacher-Turán, *Illinois J. Math.* **6** (1962), 122–127.

[Erd63] P. Erdős, On the structure of linear graphs, *Israel J. Math.* **1** (1963), 156–160.

[Erd66] P. Erdős, Some recent results on extremal problems in graph theory (results) pp. 117–130 in *Theory of Graphs (Internat. Sympos., Rome, 1966)*, Gordon and Breach, New York; Dunod, Paris.

[Erd68] P. Erdős, On some new inequalities concerning extremal properties of graphs, pp. 77–81 in *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, Academic Press, 1968.

[Erd69] P. Erdős, On the number of complete subgraphs and circuits contained in graphs, *Časopis Pěst. Mat.* **94** (1969), 290–296.

[Erd70] P. Erdős, On the graph theorem of Turán (Hungarian), *Mat. Lapok* **21** (1970), 249–251.

[ErGa59] P. Erdős and T. Gallai, On maximal paths and circuits of graphs, *Acta Math. Acad. Sci. Hungar.* **10** (1959), 337–356.

[ERS66] P. Erdős, A. Rényi and V. Sós, On a problem of graph theory, *Studia Sci. Math. Hungar.* **1** (1966), 215–235.

[ErSi66] P. Erdős and M. Simonovits, A limit theorem in graph theory, *Studia Sci. Math. Hungar* **1** (1966), 51–57.

[ErSi73] P. Erdős and M. Simonovits, On a valence problem in extremal graph theory, *Discrete Math.*, **5** (1973), 323–334.

[ErSo70] P. Erdős and V. T. Sós, Some remarks on Ramsey's and Turán's theorem, pp. 395–404 in *Combinatorial theory and its applications, II (Proc. Colloq., Balatonfüred, 1969)*, North-Holland, Amsterdam, 1970.

[ErSo83] P. Erdős and V. T. Sós, On a generalization of Turán's graph theorem, pp. 181–185 in *Studies in Pure Mathematics*, Birkhäuser, 1983.

[ErSt46] P. Erdős and A. H. Stone, On the structure of linear graphs, *Bull. Amer. Math. Soc.* **52** (1946), 1087–1091.

[Fan02] G. Fan, Distribution of cycle lengths in graphs, *J. Combin. Theory, Ser B* **84** (2002), 187–202.

[FaHa94] G. Fan and R. Häggkvist, The square of a Hamiltonian cycle, *SIAM J. Discrete Math.* **7** (1994), 203–212.

[FaSc75] R. Faudree and R.H. Schelp, Path-path Ramsey-type numbers for the complete bipartite graph, *J. Combin. Theory, Ser. B* **19** (1975), 161–173.

[Fau92] R. Faudree, Complete subgraphs with large degree sums, *J. Graph Theory* **16** (1992), 327–334.

[Fer83] W. Fernandez de la Vega, On the maximum density of graphs which have no subcontraction to K_s, *Discrete Math.* **46** (1983), 109–110.

[FiRa92] D. Fisher and J. Ryan, Bounds on the number of complete subgraphs, *Discrete Math.* **103** (1992), 313–320.

[Fis89] D. C. Fisher, Lower bounds on the number of triangles in a graph, *J. Graph Theory* **13** (1989), 505–512.

[Fur83] Z. Füredi, Graphs without quadrilaterals, *J. Combin. Theory, Ser B* **34** (1983), 187–190.

[GKS84] A. Gyárfás, J. Komlós and E. Szemerédi, On the distribution of cycle lengths in graphs, *J. Graph Theory* **8** (1984), 441–462.

[GHS02] R. Gould, P. Haxell and A. Scott, A note on cycle lengths in graphs, *Graphs Combin.* **18** (2002), 491–498.

[GPSV85] A. Gyárfás, H. Prömel, E. Szemerédi and B. Voigt, On the sum of the reciprocals of cycle lengths in sparse graphs, *Combinatorica* **5** (1985), 41–52.

[Gow98] W. Gowers, Lower bounds of tower type for Szemerédi's uniformity lemma, *Geom. Funct. Anal.* **7** (1997), 322–337.

[Hag82] R. Häggkvist, Odd cycles of specified length in nonbipartite graphs, pp. 89–99 in *Graph theory (Cambridge, 1981)*, North-Holland Math. Stud., **62**, North-Holland, 1982.

[HaJi98] R. Häggkvist and G. Jin, Graphs with odd girth at least seven and high minimal degree, *Graphs Comb.* **14** (1998), 351–362.

[HaSz70] A. Hajnal and E. Szemerédi, Proof of a conjecture of P. Erdös, pp. 601–623 in *Combinatorial Theory and Its Applications, II (Proc. Colloq., Balatonfüred, 1969)*, North-Holland, 1970.

[Jin95] G. P. Jin, Triangle-free four-chromatic graphs, *Discrete Math.* **145** (1995), 151–170.

[Ish02] Y. Ishigami, Proof of a conjecture of Bollobás and Kohayakawa on the Erdös-Stone theorem, *J. Combin. Theory, Ser B* **85** (2002), 222–254.

[Kha77] N. Khadžiivanov, Inequalities for graphs (Russian), *C. R. Acad. Sci. Bul.* **30** (1977), 793–796.

[Kom99] J. Komlós, The blow-up lemma, *Combin. Probab. Comput.* **8** (1999), 161–176.

[Kom00] J. Komlós, Tiling Turán theorems, *Combinatorica* **20** (2000), 203–218.

[Kos82] A. Kostochka, The minimum Hadwiger number for graphs with a given mean degree of vertices (Russian), *Metody Diskret. Analiz.* **38** (1982), 37–58.

[Kos84] A. Kostochka, A lower bound of the Hadwiger number of graphs by their average degree, *Combinatorica* **4** (1984), 307–316.

[KoSi96] J. Komlós and M. Simonovits, Szemerédi's regularity lemma and its applications in graph theory, *Combinatorics, Paul Erdős is Eighty*, Vol. **2** (Keszthely, 1993), pp. 295–352, Bolyai Soc. Math. Stud., 2, János Bolyai Math. Soc., Budapest, 1996.

[KPR87] G. Kolaitis, H. Prömel, and B. Rothschild, K_{l+1}-free graphs: asymptotic structure and a $0-1$ law, *Trans. Amer. Math. Soc.* **303** (1987), 637–671.

[KSS97] J. Komlós, G. Sárközy, and E. Szemerédi, Blow-up lemma, *Combinatorica* **17** (1997), 109–123.

[KSS98] J. Komlós, G. Sárközy, and E. Szemerédi, Proof of the Seymour conjecture for large graphs, *Ann. Combin.* **2** (1998), 43–60.

[KSS01] J. Komlós, G. Sárközy, and E. Szemerédi, Proof of the Alon-Yuster conjecture, *Discrete Math.* **235** (2001), 255–269.

[KSS01a] J. Komlós, G. Sárközy and E. Szemerédi, Spanning trees in dense graphs, *Combin. Probab. Comput.* **10** (2001), 397–416.

[LoSi83] L. Lovász and M. Simonovits, On the number of complete subgraphs of a graph II, *Studies in pure mathematics,* pp. 459–495, Birkhäuser, 1983.

[Mad67] W. Mader, Homomorphieeigenschaften und mittlere Kantendichte von Graphen (German), *Math. Ann.* **174** (1967), 265–268.

[Mad68] W. Mader, Homomorphiesätze für Graphen (German), *Math. Ann.* **178** (1968), 154–168.

[Man07] W. Mantel, Problem 28, soln. by H. Gouventak, W. Mantel, J. Teixeira de Mattes, F. Schuh and W. A. Wythoff, *Wiskundige Opgaven* **10** (1907), 60–61.

[MoMo62] J. Moon and L. Moser, On a problem of Turán, *Magyar Tud. Akad. Mat. Kutató Int. Közl.* 7 (1962), 283–286.

[MoSt65] T. Motzkin and E. Straus, Maxima for graphs and a new proof of a theorem of Turán, *Canad. J. Math.*, **17** (1965), 533–540.

[Myc55] J. Mycielski, Sur le coloriage des graphs (French), *Colloq. Math.* **3**, (1955), 161–162.

[MyTh02] J. Myers and A. Thomason, The extremal function for noncomplete minors, preprint.

[NoSt63] E. Nordhaus and B. Stewart, Triangles in an ordinary graph, *Canad. J. Math.* **15** (1963), 33–41.

[Ore60] O. Ore, Note on Hamilton circuits, *Amer. Math. Monthly* **67** (1960), 55.

[Pos62] L. Pósa, A theorem concerning Hamilton lines, *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **7** (1962), 225–226.

[PrSt91] H. Prömel and A. Steger, Excluding induced subgraphs: quadrilaterals, *Random Structures Algorithms* **2** (1991), 55–71.

[PrSt92] H. Prömel and A. Steger, Excluding induced subgraphs III. A general asymptotic, *Random Structures Algorithms* **3** (1992), 19–31.

[PrSt93] H. Prömel and A. Steger, Excluding induced subgraphs II. Extremal graphs *Discrete Appl. Math.* **44** (1993), 283–294.

[Rei58] I. Reiman, Über ein Problem von K. Zarankiewicz (German), *Acta. Math. Acad. Sci. Hungar.* **9** (1958), 269–273.

[SaWo97] J.-F. Saclé, M. Woźniak, The Erdős-Sós conjecture for graphs without C_4, *J. Combin. Theory, Ser B* **70** (1997), 367–372.

[Shi86] R. Shi, Ore-type conditions for pancyclism of Hamiltonian graphs (Chinese), *J. Systems Sci. Math. Sci.* **11** (1991), 79–90.

[Shi92] R. Shi, 2-neighborhoods and Hamiltonian conditions, *J. Graph Theory* **16** (1992), 267–271.

[Sim68] M. Simonovits, A method for solving extremal problems in graph theory, stability problems, pp. 279–319 in *Theory of Graphs (Proc. Colloq., Tihany, 1966)*, Academic Press, 1968.

[Sim74] M. Simonovits, Extermal graph problems with symmetrical extremal graphs. Additional chromatic conditions, *Discrete Math.* **7** (1974), 349–376.

[Sim83] M. Simonovits, Extremal graph theory, pp. 161–200 in *Selected Topics in Graph Theory* (ed. L. W. Beineke and R. J. Wilson), vol. 2, Academic Press, 1983.

[Sze72] E. Szemerédi, On graphs containing no complete subgraph with 4 vertices (Hungarian), *Mat. Lapok* **23** (1972), 113–116.

[Tho84] A. Thomason, An extremal function for contractions of graphs, *Math. Proc. Cambridge Philos. Soc.* **95** (1984), 261–265.

[Tho01] A. Thomason, The extremal function for complete minors, *J. Combin. Theory Ser. B* **81** (2001), 318–338.

[Tur41] P. Turán, On an extremal problem in graph theory (Hugarian), *Mat. és Fiz. Lapok* **48** (1941), 436–452.

[Ver00] J. Verstraëte, On arithmetic progressions of cycle lengths in graphs, *Combin. Probab. Comput.* **9** (2000), 369–373.

[VoZu77] H.-J. Voss and C. Zuluaga, Maximale gerade und ungerade Kreise in Graphen I. (German) *Wiss. Z. Techn. Hochsch. Ilmenau* **23** (1977), 57–70.

[WLL00] M. Wang, G. Li and A. Liu, A result of Erdös-Sós conjecture, *Ars Combin.* **55** (2000), 123–127.

[Zyk49] A. A. Zykov, On some properties of linear complexes (Russian), *Mat. Sbornik N.S.* **24(66)** (1949), 163–188.

8.2 RANDOM GRAPHS

Nicholas Wormald, University of Waterloo, Canada*

8.2.1 Random Graph Models
8.2.2 Threshold Functions
8.2.3 Small Subgraphs and the Degree Sequence
8.2.4 The Phase Transition
8.2.5 Many More Properties of Random Graphs
8.2.6 Random Regular Graphs
8.2.7 Other Random Graph Models
8.2.8 Random Graph Processes
References

Introduction

The field of random graphs came into its own with papers of Erdős and Rényi in 1959–61. Earlier it had surfaced mainly in probabilistic proofs, where facts about random graphs were used to prove the existence of graphs with desired properties. Then, progressively, many interesting features of the random graphs themselves were discovered. For instance, a large random graph can be relied upon (in a sense to be made precise) to have diameter 2 and to contain any arbitrary fixed subgraph. Consequently, it will contain large complete subgraphs, and thus will have large genus and will not be k-colorable for small k. A number of models of random graphs are commonly studied variations arising from other areas such as communication networks. Several monographs have been devoted to the subject ([Bo01], [Pa85], [Ko99], [JaLuRu00], [Sp01]).

8.2.1 Random Graph Models

NOTATION: For §8.2, let n be a positive integer and let p be a real number, $0 \le p \le 1$, and $q = 1 - p$. An n-vertex simple graph G has vertex set $V = [n] = \{1, \ldots, n\}$. The number of edges in the complete graph K_n is $N = \binom{n}{2}$.

NOTATION: Probability and expectation are denoted by $\mathbf{P}$ and $\mathbf{E}$, respectively, and variance is denoted by $\mathbf{Var}$.

DEFINITIONS

Here we consider the two most common random graph models.

D1: For $0 \le p \le 1$, the ***binomial*** (or ***Bernoulli***) ***random graph***, denoted by $\mathcal{G}(n, p)$, is a probability space whose underlying set is the set of n-vertex graphs. The probability function is determined by specifying that the edges of K_n occur independently with probability p each. Equivalently, the probability of any given graph with m edges is defined to be

$$p^m q^{N-m}$$

D2: For $0 \leq m \leq N$, the ***uniform*** (or ***Erdős-Rényi***) ***random graph***, denoted by $\mathcal{G}(n,m)$, is the uniform probability space on those graphs with exactly n vertices and m edges. Thus, the probability of any n-vertex m-edge graph is

$$\binom{N}{m}^{-1}$$

TERMINOLOGY NOTE: To denote that G is a random graph with the probability distribution of $\mathcal{G}(n,p)$, we write $G \in \mathcal{G}(n,p)$; alternatively we write $\mathcal{G}(n,p)$ for such G.

D3: An ***event*** is a subset of the graphs in whichever model is under discussion. Given a graph probability space, every graph property Q defines an event in a natural way, being the set of graphs with property Q. This event is also denoted by Q.

D4: A graph property Q is ***increasing*** if a graph G has property Q whenever one of its spanning subgraphs has Q.

D5: A graph property Q is ***convex*** if G has property Q whenever $G_1 \subseteq G \subseteq G_2$ for some G_1, G_2 both having property Q and with the same vertex sets as G.

D6: ***Decreasing properties*** are the complements of increasing properties.

D7: ***Monotone properties*** are either increasing or decreasing.

EXAMPLES

E1: Let Q be the graph property "is complete". Then for $\mathcal{G}(n,p)$, we have

$$\mathbf{P}(Q) = p^N$$

E2: Let Q be the property "vertex 1 is isolated". For the Bernoulli graph $\mathcal{G}(n,p)$, we have

$$\mathbf{P}(Q) = q^{n-1}$$

For the Erdős-Rényi graph $\mathcal{G}(n,m)$, we have

$$\mathbf{P}(Q) = \frac{\binom{N-n+1}{m}}{\binom{N}{m}}$$

E3: The properties in the two previous examples are monotone and hence convex, as are the properties "G has a subgraph in $\mathcal{F}$" for any family of graphs $\mathcal{F}$, and "G has minimum degree k" for any k.

E4: The property "G has diameter exactly 2" is neither increasing nor decreasing, but is convex, whereas "G has a vertex of degree exactly 2" is not convex.

REMARK

R1: The independence of edge occurrence in $\mathcal{G}(n,p)$ tends to simplify calculations of probability.

Asymptotics

Most of the interest in random graphs lies in the asymptotic behavior as $n \to \infty$, with $p = p(n)$ and $m = m(n)$ functions of n.

DEFINITIONS

D8: An event A_n holds ***asymptotically almost surely*** (a.a.s.) if $\mathbf{P}(A_n) \to 1$ as $n \to \infty$. This applies to events A_n defined on any sequence of probability spaces indexed by n, such as the Bernoulli random graph $\mathcal{G}(n,p)$ with $p = p(n)$ a function of n, or the Erdős-Rényi random graph $\mathcal{G}(n,m)$ with $m = m(n)$.

NOTATION: Suppose that $|f| < \phi g$, for some functions $f(n)$, $g(n)$, and $\phi(n)$.

- If $\phi(n)$ is bounded, then we write $f = O(g)$.
- If $\phi \to 0$ as $n \to \infty$, then we write $f = o(g)$ or alternatively, $f \ll g$ or $g \gg f$.
- If $f = O(g)$ and $g = O(f)$, then we write $f = \Theta(g)$.
- If $f(n) = (1 + o(1))g(n)$, then we write $f \sim g$.

CONVENTION: The appearance of $o(g)$ in a formula denotes a function f for which $f = o(g)$, and the same convention applies to $O(g)$ and $\Theta(g)$.

NOTATION: If S is a statement about a sequence of random variables involving any of these notations, rather than an event, we write "a.a.s. S" to mean that all inequalities $|f| < \phi g$ that are implicit in S hold a.a.s.

D9: Let $X_1, X_2, \ldots$ be random variables and $\lambda \geq 0$ constant. We say that $X = X_n$ is ***asymptotically Poisson with mean*** λ if

$$\mathbf{P}(X = k) = \frac{e^{-\lambda}\lambda^k}{k!} + o(1)$$

for all fixed integers $k \geq 0$, as $n \to \infty$. This also applies if $\lambda = \lambda(n)$ is a bounded function of n.

D10: For a random variable X with $0 < \mathbf{Var}X < \infty$, the ***standardized variable*** is

$$\hat{X} = (X - \mathbf{E}X)/\sqrt{\mathbf{Var}X}$$

D11: We say that $X = X_n$ is ***asymptotically normal*** if for all fixed values of a,

$$\mathbf{P}(\hat{X} \leq a) = o(1) + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} e^{-x^2/2}\, dx \quad \text{as } n \to \infty$$

TERMINOLOGY NOTE: Elsewhere, the notations *a.e.* (almost every), *whp* (with high probability), or *a.s.* (almost surely) are sometimes used instead of a.a.s.

TERMINOLOGY NOTE: Our definition of $o(g)$ is nonstandard, equivalent to the usual definition, but also accommodating the a.a.s. versions. For instance, [JaLuRu00] use $f = O_C(g)$, $f = \Theta_C(g)$ and $f = o_p(g)$ for a.a.s. $f = O(g)$, a.a.s. $f = \Theta(g)$ and a.a.s. $f = o(g)$, respectively.

FACTS

F1: (See [Bo01] and [JaLuRu00].) Let Q be a graph property and $0 \leq p = p(n) \leq 1$ such that $pqN \to \infty$. If $m = m(n)$ is a positive integer function, define $x = x(m,n)$ by $m = pN + x\sqrt{pqN}$.

(i) If Q is a.a.s. true in $\mathcal{G}(n,m)$ whenever x is bounded, then Q is a.a.s. true in $\mathcal{G}(n,p)$.

(ii) The converse of (i) is true if Q is convex.

F2: (Following from [Bo79] for example.) Let $0 < p < 1$, let H be a fixed graph, and F an induced subgraph of H. Then a.a.s. for $G \in \mathcal{G}(n,p)$, every isomorphism of F with an induced subgraph of G extends to an isomorphism of H with an induced subgraph of G. It follows that for k fixed,

(i) a.a.s. every vertex in G is in a complete subgraph of size k and in an independent set of size k,

(ii) a.a.s. G has diameter 2 and is k-connected; moreover, deleting any k vertices from G leaves a graph of diameter 2.

REMARKS

R2: Fact 1 shows that the two models under consideration share many properties. For instance, the properties in Fact 2 hold also in $\mathcal{G}(n,m)$ with $m \sim cn^2$, any constant $0 < c < 1/2$. For this reason, we usually limit ourselves to stating properties of just one model, when the corresponding property holds in the other model by Fact 1.

R3: The topics of random graphs, and asymptotic enumeration of graphs, are intimately intertwined; many results in either area owe their existence to techniques from the other, especially with the model $\mathcal{G}(n,m)$. For example, one may derive the asymptotic number of graphs with n vertices, m edges and no triangles, by multiplying $\binom{N}{m}$ by the probability that $G \in \mathcal{G}(n,m)$ has no triangles. Such results can be equally appealing when stated in either form. For uniformity we state only the random graph form here.

8.2.2 Threshold Functions

DEFINITIONS

D12: A ***threshold function*** for a property Q in $\mathcal{G}(n,p)$ is a function $f(n)$ such that for $G \in \mathcal{G}(n,p)$, with $p = p(n)$,

$$\mathbf{P}(G \text{ has } Q) \to \begin{cases} 0 & \text{if } p = o(f) \\ 1 & \text{if } f = o(p) \end{cases}$$

or alternatively, such that this is true with 0 and 1 interchanged.

D13: A threshold function for a property Q is ***sharp*** if for every fixed $\epsilon > 0$,

$$\mathbf{P}(G \text{ has } Q) \to \begin{cases} 0 & \text{if } p < (1-\epsilon)f, \\ 1 & \text{if } p > (1+\epsilon)f \end{cases}$$

(or the same with 0 and 1 interchanged).

FACTS

F3: [BoTh87] Every monotone property has a threshold function in $\mathcal{G}(n,p)$.

F4: [FrKa96] Every monotone property with a threshold function f in $\mathcal{G}(n,p)$ such that $\log(1/f) = o(\log n)$ has a sharp threshold function.

F5: [AcFr99] For fixed $k \geq 3$, the property of being k-colorable has a sharp threshold function in $\mathcal{G}(n,p)$.

F6: [ShSp88] Let A be a property expressible in the first-order theory of graphs, that is, using variables to represent vertices, using the equality and adjacency relations, and the usual Boolean connectives, and the quantifiers $\forall$, $\exists$. For any irrational α, $0 < \alpha < 1$, in $\mathcal{G}(n,p)$ with $p = n^{-\alpha}$, A is either a.a.s. true or a.a.s. false.

F7: [ErRé60] For $G \in \mathcal{G}(n,p)$ and fixed $k \geq 1$, $(\log n)/n$ is a sharp threshold function for the minimum vertex degree of G being at least k.

F8: [BoFr85] For $G \in \mathcal{G}(n,p)$ and fixed $k \geq 1$, $(\log n)/n$ is a sharp threshold function for the property that G has k edge-disjoint Hamilton cycles.

Strengthened versions of Facts 7 and 8 are given later.

REMARKS

R4: For a given monotone property, all sharp thresholds are clearly asymptotically equal.

R5: Statements analogous to Facts 3 and 4 also hold in more general probability spaces concerning random subsets of a set (with an extra symmetry condition, in the case of Fact 4).

R6: Threshold functions are known for many properties, and there are many properties for which more accurate information is known than the mere existence of a sharp threshold.

R7: Fact 6 is an example of a zero-one law in random graphs; see [JaLuRu00] and [Sp01] for much more on this topic.

R8: Facts 7 and 8 together imply that $(\log n)/n$ is a sharp threshold function for G being k-edge-connected [ErRé61], having a matching which either is perfect (if n is even) or meets all but one vertex (if n is odd) [ErRé66], having a Hamilton cycle [KoSz83].

R9: Threshold functions are also defined in $\mathcal{G}(n,m)$: f is a threshold function for a property Q if $\mathbf{P}(G \text{ has } Q) \to 0$ for $m = o(f)$ and $\mathbf{P}(G \text{ has } Q) \to 1$ for $f = o(m)$ (or with 0 and 1 interchanged). Sharp threshold functions are then defined in the obvious way. From Fact 1, $\frac{1}{2}n \log n$ is a threshold function in $\mathcal{G}(n,m)$ for the properties mentioned in Facts 7 and 8, and is, moreover, sharp.

8.2.3 Small Subgraphs and the Degree Sequence

DEFINITIONS

D14: The ***maximum density*** of a graph G is

$$\mu(G) = \max\left\{ \frac{|E(F)|}{|V(F)|} \; : \; F \subseteq G, \; |V(F)| > 0 \right\}$$

D15: A graph G is ***strictly balanced*** if its maximum density is achieved uniquely by $F = G$.

FACTS ABOUT SMALL SUBGRAPHS

NOTATION: Here H is a fixed graph with at least one edge, and X_H ($\tilde{X}_H$) denotes the number of subgraphs of $G \in \mathcal{G}(n,p)$ ($G \in \mathcal{G}(n,m)$, respectively) isomorphic to H.

F9: [Bo81] Let H be strictly balanced with k vertices and $j \geq 2$ edges, and automorphism group of order a. Let $c > 0$ be fixed and $p = cn^{-k/j}$. Then X_H is asymptotically Poisson with mean c^j/a.

F10: (See [Bo01].) For arbitrary H, a threshold function for $\{X_H > 0\}$ is $n^{-1/\mu(H)}$.

F11: [Ru88] If $|E(H)| \geq 1$ then the distribution of X_H is asymptotically normal if and only if $np^{\mu(H)} \to \infty$ and $n^2(1-p) \to \infty$.

F12: [Ja94] If $|E(H)| > 1$, $m \gg \sqrt{n}$, $N - m \gg \sqrt{n}$ and $ns(m/N)^{\mu(H)} \to \infty$, then $\tilde{X}_H$ is asymptotically normal.

F13: [JaLuRu90] For every $p = p(n) < 1$, $e^{-\Psi_H/(1-p)} \leq \mathbf{P}(X_H = 0) \leq e^{-\Theta(\Psi_H)}$ where $\Psi_H = \min\{\mathbf{E}(X_F) \, : \, F \subseteq H, \, |E(F)| > 0\}$.

F14: [OsPrTa03], [PrSt96] Let $P(n,m) = \mathbf{P}(\mathcal{G}(n,m)$ is bipartite$)$. For all $\epsilon > 0$,

$$\mathbf{P}(\tilde{X}_{K_3} = 0) \sim P(n,m) \quad \text{if} \quad m \geq (1+\epsilon)\frac{\sqrt{3}}{4} n^{3/2}\sqrt{\log n}$$

and $\mathbf{P}(\tilde{X}_{K_3} = 0) \gg P(n,m)$ if $n/2 \leq m \leq (1-\epsilon)\frac{\sqrt{3}}{4}n^{3/2}\sqrt{\log n}$.

F15: [Wo96] For $p = o(n^{-2/3})$,

$$\mathbf{P}(X_{K_3} = 0) \sim e^{-\frac{1}{6}p^3n^3 + \frac{1}{4}p^5n^4 - \frac{7}{12}p^7n^5}$$

For $d = m/N = o(n^{-2/3})$,

$$\mathbf{P}(\tilde{X}_{K_3} = 0) \sim e^{-\frac{1}{6}d^3n^3}$$

F16: [PrSt92] For $G \in \mathcal{G}(n, \frac{1}{2})$, $\mathbf{P}(X_H = 0) \sim \Pr(G$ is $k-$colorable) iff H has chromatic number $k+1$ but contains a color critical edge (i.e., an edge whose omission from H reduces the chromatic number to k).

FACTS ABOUT THE DEGREE SEQUENCE

NOTATION: The number of vertices of degree k in a random graph is denoted D_k, and $d_1 \geq \cdots \geq d_n$ is a descending ordering of the degrees of the vertices.

F17: [ErRé61] Let k be a fixed natural number, x a fixed real, and $m = m(n) = \frac{1}{2}n(\log n + k \log\log n + x + o(1))$. In $\mathcal{G}(n,m)$, $\mathbf{P}(d_n = k+1) \to e^{-e^{-x}/k!}$, and a.a.s. $d_n = k$ or $k+1$. (Note that d_n is the minimum vertex degree.)

F18: [Bo01] Let $k = k(n)$ be a natural number, and for fixed $\epsilon > 0$ let $\epsilon n^{-3/2} \leq p = p(n) \leq 1 - \epsilon n^{-3/2}$. If $\mathbf{E}D_k \to c$ then in $\mathcal{G}(n,p)$, the random variable D_k is asymptotically Poisson with mean c. (Note that $\mathbf{E}D_k = n\binom{n-1}{k}p^k(1-p)^{n-k}$.)

F19: [BaHoJa92] If $k = k(n)$ and either (i) $np \to 0$ and $k \geq 2$, or (ii) np is bounded away from 0 and $(np)^{-1/2}|k - np| \to \infty$, then the random variable D_k is asymptotically Poisson in $\mathcal{G}(n,p)$, in the sense that the total variation distance between $D_k = D_k(n)$ and a Poisson random variable $Z = Z(n)$ tends to 0 as $n \to \infty$. (In some cases $\mathbf{E}Z(n) \neq O(1)$.)

F20: [Bo01] Suppose that $p(1-p)n \gg (\log n)^3$ and y is a fixed real number. Then for every fixed m, in $\mathcal{G}(n,p)$

$$\lim_{n\to\infty} \mathbf{P}(d_m < f(n,p,y)) = e^{-e^{-y}} \sum_{k=0}^{m-1} e^{-ky}/k!$$

where $f(n,p,y) = pn + \sqrt{2p(1-p)n\log n}\Big(1 - \frac{\log\log n}{4\log n} + \frac{y-\log(2\sqrt{\pi})}{2\log n}\Big)$.

F21: [Bo01] The random graph $\mathcal{G}(n,p)$ a.a.s. has a unique vertex of maximum degree and a unique vertex of minimum degree iff $np(1-p) \gg \log n$ iff $\mathcal{G}(n,p)$ a.a.s. has a unique vertex of maximum degree or a unique vertex of minimum degree.

F22: [McWo97, Theorem 2.6] Let $\mathcal{B}_p(n)$ be a sequence of n independent binomial variables, each Binom$(n-1,p)$ (which is the distribution of the degree of any given vertex in $\mathcal{G}(n,p)$). Consider the degree sequences $\mathcal{D}_p(n)$ of $\mathcal{G}(n,p)$, and $\mathcal{D}_m(n)$ of $\mathcal{G}(n,m)$. Let A be any event defined on sequences, and assume that either $\log n/n^2 \ll p(1-p) \ll n^{-1/2}$ or $\liminf p(1-p)\log n > 2/3$.

(i) If $m = pN$ is always an integer, then the probabilities of the event A in the two models $\mathcal{D}_m(n)$, and $\mathcal{B}_p(n)$ restricted to sequences with sum $2m$, differ by $o(1)$.

(ii) Choose p' from the normal distribution with mean p and variance $p(1-p)/(2N)$, truncated to the unit interval $(0,1)$. Then the probabilities of the event A in the two models $\mathcal{D}_p(n)$, and $\mathcal{B}_{p'}(n)$ restricted to sequences with even sum, differ by $o(1)$.

REMARKS

R10: On small subgraphs, similar results on the number of induced copies of H were obtained by Janson (see [JaŁuRu00, Chapter 6]). Here there is usually a second (disappearance) threshold. For the number of copies vertex-disjoint from all other copies, again there is a second disappearance threshold, with significant pioneering results in [Su90]; see also the discussion in [JaŁuRu00]. Threshold results on covering every vertex by a copy of H were given by Spencer [Sp90], as part of more general results on extending all partial embeddings of k vertices of H. For threshold results on the property that every coloring of the edges of $\mathcal{G}(n,p)$ contains a monochromatic copy of a given graph G, see [RöRu95].

R11: On probabilities in the tail of the distribution of X_H and $\tilde{X}_H$, exponentially small upper bounds have been obtained but are not always sharp ([Vu01],[JaRu02]). There are a number of other papers estimating the probability of nonexistence of a given subgraph, or equivalently results on the number of graphs which do not contain the subgraph, e.g. [PrSt96a].

R12: Fact 17 (and a number of similar ones) are stated with limiting probability strictly between 0 and 1, covering the whole range. For monotone properties, this implies the results for limiting probabilities equal to 0 and 1. For example, from Fact 17 it follows that with $m = \frac{1}{2}n(\log n + k\log\log n + x)$,

$$\mathbf{P}(d_n \geq k+1) \to 1 \text{ for } x \to \infty \quad \text{and} \quad \mathbf{P}(d_n \geq k+1) \to 0 \text{ for } x \to -\infty$$

R13: Many properties of the degree sequences $\mathcal{D}_p(n)$ and $\mathcal{D}_m(n)$ of the random graphs $\mathcal{G}(n,p)$ and $\mathcal{G}(n,m)$, respectively, follow from Fact 22. It is conjectured in [McWo97] that the restriction on p can be relaxed to simply $p(1-p) \gg \log n/n^2$, which covers all p of any interest whatsoever.

8.2.4 The Phase Transition

Erdős and Rényi initiated the study of the random graph $\mathcal{G}(n,m)$ as an evolving object, growing from a sparse, disconnected graph for small m to a highly connected graph for large m, finally a complete graph when $m = N$. The biggest issue in this study has been the ***phase transition*** at $m \sim \frac{1}{2}n$, where increasing m by $o(n)$ can change the size of the largest component from a.a.s. $O(\log n)$ to a.a.s. nearly a constant times n. Here we state properties of $\mathcal{G}(n,m)$ or $\mathcal{G}(n,p)$ in a large neighborhood of this phenomenon. Most of these translate from one model to the other by Fact 1. We begin with a simple statement shown by Erdős and Rényi, the tripartite nature of which leads to the term ***double jump***.

F23: [ErRé60] Fix $c > 0$, and for $c > 1$ define $b = b(c)$ so as to satisfy $b + e^{-bc} = 1$ (see Figure 8.2.1). Let L denote the number of vertices in the largest component (called the *giant*) in $G \in \mathcal{G}(n,m)$ where $m = \lfloor cn/2 \rfloor$. Then a.a.s.

$$L = \begin{cases} O(\log n) & \text{if } c < 1, \\ \Theta(n^{2/3}) & \text{if } c = 1, \\ (b + o(1))n & \text{if } c > 1. \end{cases}$$

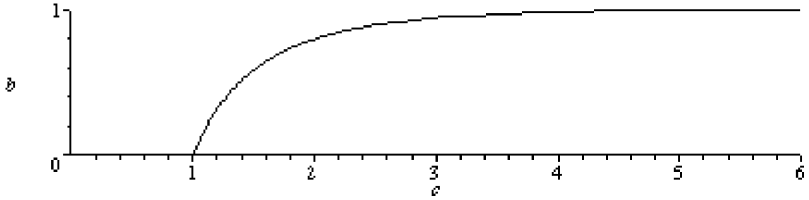


Figure 8.2.1 **The growth of the giant.**

More precise examination by Bollobás began a revelation of details on how the phase transition takes place. The phase transition was eventually shown to have width of the order of $n^{2/3}$. So we call $m = n/2 + O(n^{2/3})$ the critical phase, before that is subcritical and after is supercritical. Retrospectively, the significance of the "double jump" is mainly historical, as it manifests itself only when requiring m to be a fixed constant times n.

DEFINITIONS

D16: The ***excess of a graph*** with n vertices and m edges is $m - n$.

D17: A connected graph is ***complex*** if its excess is at least 1 (so there are at least two cycles).

D18: The ***k-core of a graph*** is the largest subgraph with all its vertex degrees at least k.

FACTS

For many of these, specific bounds on the error terms are known but unstated here.

Throughout the phase transition

F24: [Ja93] For any $m(n)$ and any $k \ll n^{2/3}$, the random graph $G(n,m)$ a.a.s has no complex component with fewer than k vertices.

F25: [Bo01, Theorem 5.15] For any $p(n)$ and any $k \gg n^{2/3}$, the random graph $G(n,p)$ a.a.s. contains no component which is a tree of order at least k.

F26: [ErRé60] If $0 < c \neq 1$ is fixed and $m \sim cn/2$, then the size of the largest tree component in $\mathcal{G}(n,m)$ is a.a.s. $(a+o(1))\log n$ where $a = a(c) = 1/(c-1-\log c)$.

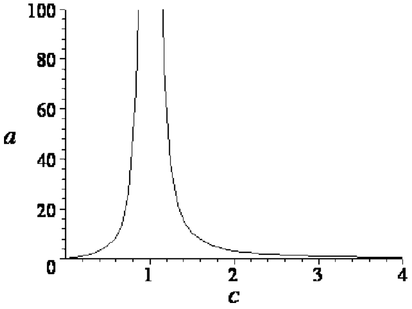


Figure 8.2.2 **The size of the largest tree component.**

Subcritical phase: $n/2 - m \gg n^{2/3}$

F27: [ErRé60] If $0 < c < 1$ is fixed and $m \sim cn/2$, then the probability that $\mathcal{G}(n,m)$ is a forest is asymptotic to $e^{c/2+c^2/4}\sqrt{1-c}$, and the expected total number of vertices belonging to cycles tends towards $c^3/(2-2c)$.

F28: (See [JaLuRu00].) Let $r \geq 1$ be fixed and $n^{2/3} \ll s \ll n$. In $\mathcal{G}(n, n/2-s)$, the r largest components are a.a.s. trees of order $(1/2+o(1))(n/s)^2\log(s^3/n^2)$.

F29: [JaLuRu00] For $m < n/2$, the probability that $\mathcal{G}(n,m)$ contains a complex component is less than $2n^2/(n-2m)^3$. (Note that this tends to 0 in the subcritical phase.)

F30: [Bo01 Corollary 5.8] For $p = c/n$, $0 < c < 1$, and any $\omega = \omega(n) \to \infty$, $G \in \mathcal{G}(n,p)$ a.a.s. has at most ω vertices in unicyclic components. Consequently, the length of the longest cycle is a.a.s. at most ω.

F31: [JaLuRu00 Section 5.4] Let $n^{2/3} \ll s \ll n$. For any $\omega = \omega(n) \to \infty$, the length ℓ of the longest cycle in $\mathcal{G}(n, n/2-s)$ a.a.s. satisfies $\ell/\omega < n/s < \ell\omega$.

Critical phase: $m = n/2 + O(n^{2/3})$

F32: [JaLuRu00 Section 5.5] Let $m = n/2 + O(n^{2/3})$ and let r_a be the number of components of $\mathcal{G}(n,m)$ with excess a. For any $\omega = \omega(n) \to \infty$, a.a.s. $\sum_{a\geq 1} ar_a < \omega$, and the total number of vertices in complex components of $\mathcal{G}(n,m)$ is at most $\omega n^{2/3}$.

F33: [JaKnLuPi93 Theorem 5] Let $m = n/2 + O(n^{1/3})$ and fix $q \geq 1$. The probability that in $\mathcal{G}(n,m)$ there are exactly r_i components of excess i for $1 \leq i \leq q$ and none of greater excess is

$$\left(\frac{4}{3}\right)^r \sqrt{\frac{2}{3}}\, \frac{c_1^{r_1}}{r_1!}\, \frac{c_2^{r_2}}{r_2!} \cdots \frac{c_q^{r_q}}{r_q!}\, \frac{r!}{(2r)!} + O(n^{-1/3})$$

where $r = r_1+2r_2+\cdots+qr_q$ and the c_j are (easily computed) constants. The probability that there are no components of excess 2 or more is $\sqrt{\frac{2}{3}}\cosh\sqrt{\frac{5}{18}}+O(n^{-1/3}) \approx 0.9325$.

F34: [LuPiWi94], see also [Al97] Let $m = n/2 + cn^{2/3}$ where c is constant. If (S_i, E_i) gives the size and excess of the ith component of $\mathcal{G}(n, m)$, listed so that S_i is nonincreasing with i, then the random sequence $(n^{-2/3}S_1, E_1), (n^{-2/3}S_2, E_2), \ldots$ converges in distribution to some random sequence $(X_{11}, X_{12}), (X_{21}, X_{22}), \ldots$ as $n \to \infty$.

F35: (See [JaLuRu00 Section 5.5].) If $m = n + O(n^{2/3})$ and $\omega = \omega(n) \to \infty$ then the length ℓ of the longest cycle in $\mathcal{G}(n, m)$ a.a.s. satisfies $\ell/\omega < n^{1/3} < \ell\omega$.

Supercritical phase: $m - n/2 \gg n^{2/3}$

F36: [Bo84 and Lu90] Define b so as to satisfy $b + e^{-2bm/n} = 1$. If $m - n/2 \gg n^{2/3}$ then in $\mathcal{G}(n, m)$, a.a.s. there is a complex component with $bn + O(n^{2/3})$ vertices, while every other component is not complex and has less than $n^{2/3}$ vertices.

F37: [Lu91] Let $n^{2/3} \ll s \ll n$. The longest cycle in $\mathcal{G}(n, n/2 + s)$ a.a.s. has length between $(16/3 + o(1))s^2/n$ and $(7.496 + o(1))s^2/n$.

F38: [Lu91] For any $\omega = \omega(n) \to \infty$, the length ℓ of the longest cycle in $\mathcal{G}(n, n/2 + s)$ outside the largest component a.a.s. satisfies $\ell/\omega < n/s < \ell\omega$, as does the length of the shortest cycle in the largest component.

F39: [Lu90], [JaKnLuPi93], [PiWo03] Define b as in Fact 36, $c = 2m/n$ and $t = c - cb$. By Fact 36 we may assume the largest component is unique. Let Y_1 denote the number of vertices in the 2-core of the largest component of $\mathcal{G}(n, m)$, Y_2 the number of vertices in the largest component not in the 2-core, and Y_3 the excess of the largest component. For $m = O(n)$ with $m - n/2 \gg n^{2/3}$, a.a.s. $Y_1 \sim (1-t)bn$, $Y_2 \sim tbn$ and $Y_3 \sim b(c+t-2)n/2$. Furthermore, each of the three variables is asymptotically normally distributed, and so are the numbers $Y_1 + Y_2$ (a.a.s. $\sim bn$) of vertices and $Y_1 + Y_2 + Y_3$ (a.a.s. $\sim b(c+t)n/2$) of edges in the largest component.

REMARKS

R14: Let G' denote the graph obtained by deleting the largest component from $\mathcal{G}(n, m)$ (or, if there is more than one largest component, delete all of them). From Fact 39, it follows that for $m = O(n)$ and $m - n/2 \gg n^{2/3}$, G' a.a.s. has any particular property which is a.a.s. true for $\mathcal{G}(n', m')$ as $n' \to \infty$ when $m' \sim \frac{1}{2}tn'$. (See Theorem 5.24 in [JaLuRu00] for a more precise statement.) Here $te^{-t} = ce^{-c}$, $t \in (0, 1)$.

R15: Other interesting facts about the phase transition can be seen by viewing the random graph as a process (see Section 8.2.8).

8.2.5 Many More Properties of Random Graphs

NOTATION: The connectivity of a graph G is denoted $\kappa(G)$, the edge connectivity $\lambda(G)$, the minimum vertex degree $\delta(G)$, the independence number $\alpha(G)$, and the chromatic number $\chi(G)$.

FACTS ON CONNECTIVITY, DIAMETER, MATCHINGS, CYCLES AND PATHS

The next fact implies the threshold for the property of being k-connected, and this coincides with having minimum degree k.

F40: [ErRé61] For fixed $k \geq 0$ and $m = m(n) = \frac{1}{2}n(\log n + k \log\log n + x + o(1))$,

$$\mathbf{P}(\kappa(\mathcal{G}(n,m)) = k) \to 1 - e^{-e^{-x}/k!} \quad \text{and} \quad \text{a.a.s. } \kappa(\mathcal{G}(n,m)) = k \text{ or } k+1$$

F41: [BoTh85] For $G \in \mathcal{G}(n,p)$ and any p, a.a.s. $\kappa(G) = \lambda(G) = \delta(G)$.

F42: [Bo81a] Let p be fixed, $0 < p < 1$, and $t = \lfloor n^{1/7} \rfloor$. For $G \in \mathcal{G}(n,p)$ let $G_0 = G$ and $G_i = G_{i-1} - v_i$ $(1 \leq i \leq t)$, where v_i is any member of the set S_i of vertices of minimum degree in G_{i-1}. Then a.a.s. $|S_i| = 1$, so v_i is uniquely determined, for $1 \leq i \leq t$. Also, a.a.s. $\kappa(G_i) = \delta(G_i)$ for all i, and $\delta(G_{i+1}) > \delta(G_i) + t$ for $0 \leq i < t$.

F43: [Bo81b] For constant $c > 0$ and $d = d(n) \geq 2$ an integer, we define p by $p^d n^{d-1} = \log(n^2/c)$. If $pn \gg (\log n)^3$, then for $G \in \mathcal{G}(n,p)$, a.a.s. the diameter of G is either d or $d+1$, and the probability it is d tends towards $e^{-c/2}$. Corollaries of this are:

if $p^2 n - 2\log n \to \infty$ and $n^2(1-p) \to \infty$ then a.a.s. $\mathcal{G}(n,p)$ has diameter 2;

if $m < N$ and $2m^2/n^3 - \log n \to \infty$ then a.a.s. $G(n,m)$ has diameter 2.

Facts 44 and 45 discuss properties for which the obvious necessary condition, minimum degree at least 1, has the same threshold (see Fact 17). The first is the classic special case of the second.

F44: [ErRé66] For $p = p(n) = (\log n + x + o(1))/n$, the probability that $G \in \mathcal{G}(n,p)$ has a perfect matching tends to $e^{-e^{-x}}$ as $n \to \infty$ with n restricted to the even integers.

F45: [ŁuRu91] Let T be a tree with at $t \geq 2$ vertices. For $p = p(n) = (\log n + x + o(1))/n$, the probability that $G \in \mathcal{G}(n,p)$ has a T-factor (i.e., a spanning subgraph each of whose components is isomorphic to T) tends to $e^{-e^{-x}}$ as $n \to \infty$ with n restricted to the integers divisible by t.

Fact 46 concerns a property for which the obvious necessary condition, minimum degree at least 2, has the same threshold. (See also Remark 9 and Fact 72.)

F46: [KoSz83] For $p = p(n) = (\log n + \log\log n + x + o(1))/n$, the probability that $G \in \mathcal{G}(n,p)$ has a Hamilton cycle tends to $e^{-e^{-x}}$ as $n \to \infty$.

F47: [AjKoSz81] Let $f(c)$ be the supremum of all β such that $\mathcal{G}(n, p = c/n)$ a.a.s. contains a path of length at least βn. Then $f(c) > 0$ for $c > 1$, and $\lim_{c\to\infty} f(c) = 1$.

FACTS ON INDEPENDENT SETS AND CHROMATIC NUMBER

F48: [BoEr76], [Ma76] Let $c < 1$ and suppose that $n^{-\delta} \ll p = p(n) < c$ for all $\delta > 0$. For fixed $\epsilon > 0$ the independence number $\alpha(G)$ of $G \in \mathcal{G}(n,p)$ a.a.s. satisfies $r_1 \leq \alpha(G) \leq r_2$ where

$$r_i = \lfloor 2\log_b n - 2\log_b \log_b np + 2\log_b(e/2) + 1 + (-1)^i \epsilon/p \rfloor,$$

$b = 1/(1-p)$.

F49: [Fr90] Let $\epsilon > 0$ and $r_i = \lfloor 2p^{-1}(\log np - \log\log np + \log(e/2) + (-1)^i\epsilon) \rfloor$. For some constant C_ϵ, the independence number $\alpha(G)$ of $G \in \mathcal{G}(n,p)$ a.a.s. satisfies the inequality $r_1 \leq \alpha(G) \leq r_2$ provided that $C_\epsilon/n < p = p(n) < \log^{-2} n$.

F50: [Bo88], [Lu91] Let $c < 1$ be constant. If $1/n \ll p = p(n) < c$, then for $G \in \mathcal{G}(n,p)$ a.a.s. $\chi(G) \sim -n \log \frac{1}{1-p}/(2 \log np)$.

F51: [Al93], [Kr00] Let $\epsilon > 0$. If $p(n) \geq n^{-1/4+\epsilon}$, then a.a.s. for $G \in \mathcal{G}(n,p)$, we have $\chi(G) \sim \chi_l(G)$, where $\chi_l(G)$ is the choice number (or list-chromatic number) of G.

F52: [PiSpWo96] For fixed $k \geq 3$, the existence of a k-core in $\mathcal{G}(n,p)$ has a sharp threshold function $p = c_k/n$ where $c_k = \inf_{\mu>0}\{\mu e^{\mu}/f(\mu,k)\}$, $f(\mu,k) = \sum_{i=k-1}^{\infty} \mu^i/i!$. (The k-core of a graph is the maximum subgraph of minimum degree at least k.)

F53: [Mo96] Let c_k be as above and $d_k = \sup\{d : \chi(G(n,d/n) \leq k\}$. Then $c_k \neq d_k$ for $k \geq 4$. (From Fact 52, we have $c_k \leq d_k$, since a graph with no k-core can be $(k-1)$-colored.)

FACTS ON PLANARITY, GENUS AND CROSSING NUMBER

F54: [ErRé60] For planarity in $\mathcal{G}(n,m)$, $m = n/2$ is a sharp threshold function. More precisely [LuPiWi94], there is a function f such that $0 \leq f(x) \leq 1$, $f(x) \to 0$ as $x \to -\infty$, $f(x) \to 1$ as $x \to \infty$, and $\mathbf{P}(\mathcal{G}(n,m)$ is planar$) \to f(c)$ for $m = n/2 + cn^{2/3}$ (where c is constant).

F55: [ArGr95] If $p^2(1-p)^2 \geq 8(\log n)^4/n$, the genus of $\mathcal{G}(n,p)$ is a.a.s. $(1+o(1))\frac{1}{12}pn^2$.

F56: [RöTh96] For every integer $i \geq 1$, if $n^{-i/(i+1)} \ll p \ll n^{-(i-1)/i}$, the genus of $\mathcal{G}(n,p)$ is a.a.s. $(1+o(1))(i/4(i+2))pn^2$.

F57: [SpTó02] The expected value of the crossing number of $G \in \mathcal{G}(n,p)$ (i.e., the minimum number of crossing points in a drawing of G in the plane, no three edges crossing at the same point) is $o(p^2n^4)$ if $p \sim 1/n$ and $\Theta(p^2n^4)$ if $p = c/n$ for fixed $c > 1$.

FACTS ON EIGENVALUES, AUTOMORPHISMS AND UNLABELLED GRAPHS

F58: [FüKo81] Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ be the eigenvalues of the adjacency matrix of G and let $p(n) = c$. Then a.a.s. in $\mathcal{G}(n,p)$,

$$\lambda_1 \sim np \quad \text{and} \quad \max_{2 \leq i \leq n} |\lambda_i| = 2\sqrt{pqn} + O(n^{1/3}\log n)$$

F59: [Wr70] The expected number of automorphisms of $G \in \mathcal{G}(n,m)$ tends to 1 iff $\min(m, N-m)/n - (\log n)/2 \to \infty$.

The next fact follows from this and explains why random unlabeled graphs are rarely studied: for many purposes they have the same properties as random labeled graphs.

F60: [Bo01] Let Q be a property of graphs of order n, $0 < c < 1$, and suppose that $m = m(n)$ is such that $\min(m, N-m)/n - (\log n)/2 \to \infty$. Let $G \in \mathcal{G}(n,m)$ and choose an unlabelled graph H on n vertices and m edges uniformly at random. Then $\mathbf{P}(G$ has $Q) \to c$ iff $\mathbf{P}(H$ has $Q) \to c$.

8.2.6 Random Regular Graphs

There is an abundance of interesting models of random graphs besides $\mathcal{G}(n,p)$ and $\mathcal{G}(n,m)$. One of the most common is a restriction of $\mathcal{G}(n,m)$, to graphs with specified degree sequence. Random regular graphs are a special case of this.

NOTATION: For $d \geq 0$ and dn even, $\mathcal{G}_{n,d}$ is the probability space containing just the d-regular graphs on n vertices, all being equiprobable.

FACTS

The next fact comes easily from the enumeration formulae of Bender and Canfield [BeCa78] and the model of random regular graphs given by Bollobás [Bo80]. (See the survey [Wo99] for example.)

F61: For fixed d and any fixed graph F with more edges than vertices, a random regular graph $G \in \mathcal{G}_{n,d}$ a.a.s. contains no subgraph isomorphic to F.

F62: [Bo80], [Wo81] For $d \geq 0$ and $k \geq 3$ fixed, the number of cycles of length k in a graph in $G \in \mathcal{G}_{n,d}$ is asymptotically Poisson with mean $(d-1)^k/2k$.

F63: [KrSuVuWo01], [CoFrRe02] For $3 \leq d = d(n) \leq n-4$, a random regular graph $G \in \mathcal{G}_{n,d}$ is a.a.s. d-connected.

F64: [BoFe82] Fix $d \geq 3$ and $\epsilon > 0$. The diameter $D(G)$ of $G \in \mathcal{G}_{n,d}$ a.a.s. satisfies the inequalities

$$1 + \lfloor \log_{d-1} n \rfloor + \left\lfloor \log_{d-1}\left(\frac{d-2}{6d}\log n\right)\right\rfloor \ \leq\ D(G)\ \leq\ 1 + \lceil \log_{d-1}((2+\epsilon)dn\log n)\rceil$$

F65: [McWo84] If $3 \leq d = d(n) = o(\sqrt{n})$, then the expected number of automorphisms of $G \in \mathcal{G}_{n,d}$ tends to 1 as $n \to \infty$. (c.f. Fact 59.)

F66: [KiSuVu02] If $3 \leq d = d(n) \leq n-4$, then $G \in \mathcal{G}_{n,d}$ a.a.s. has only the trivial automorphism.

F67: [KrSuVuWo01], [CoFrReRi02] If $1 \ll d = d(n) < 0.9n$, then for $G \in \mathcal{G}_{n,d}$, a.a.s. $\alpha(G) \sim 2\log d/\log(n/(n-d))$ and $\chi(G) \sim n/\alpha(G)$.

F68: [RoWo94] For fixed $d \geq 3$, a random regular graph $G \in \mathcal{G}_{n,d}$ a.a.s. has a Hamilton cycle and, for odd d, a.a.s. has edge chromatic number equal to d. Indeed, [KiWo01] for fixed $d \geq 3$, $G \in \mathcal{G}_{n,d}$ a.a.s. has a partition of its edge set into the edges of $\frac{d}{2}$ Hamilton cycles (for d even), or $\frac{d-1}{2}$ Hamilton cycles and a perfect matching (for d odd).

F69: [KrSuVuWo01], [CoFrRe02] For $3 \leq d = d(n) \leq n-1$, a random regular graph $G \in \mathcal{G}_{n,d}$ is a.a.s. hamiltonian.

F70: [Mc81] For fixed $d \geq 2$ and $|X| \leq 2\sqrt{d-1}$, the proportion of eigenvalues of the adjacency matrix of $\mathcal{G}_{n,d}$ which are at most x is

$$\frac{d\sqrt{4(d-1)-x^2}}{2\pi(d^2-x^2)} + o(1)$$

F71: [Fr91] For d even and fixed, the second-largest eigenvalue (in absolute value) of the adjacency matrix of $\mathcal{G}_{n,d}$ is a.a.s. at most $2\sqrt{2d-1} + 2\log d + o(1)$.

F72: [BrFrSuUp99, Lemma 16] The second-largest eigenvalue (in absolute value) of the adjacency matrix of $\mathcal{G}_{n,d}$ is a.a.s. $O(\sqrt{d})$ provided $d = o(\sqrt{n})$.

REMARKS

R16: There are interesting relationships between $\mathcal{G}_{n,d}$ and other random graph models, expressed in terms of contiguity (see [Wo99] or [JaLuRu00]).

R17: The behavior of the size of the largest independent set, smallest dominating set, and related functions of $\mathcal{G}_{n,d}$ is not very well determined (see the bounds in [Wo99]).

R18: There are many results on random graphs with given degree sequences, usually obtained by the same methods as for random regular graphs. For instance, random graphs with given degree sequence, all elements of which lie between 3 and $d \leq n^{0.02}$, a.a.s. have connectivity equal to minimum degree [Lu92]. Properties of the emerging giant component in random graphs with given degree sequences are studied in [MoRe98].

8.2.7 Other Random Graph Models

Many other random graph models, and closely related probabilistic models, have received much attention. Not attempting a complete or balanced treatment, we give a relevant pointer for some of the main models, either to a recent or significant result or to a major source of information: random trees [BaBoDeFlGaGo02], [MeMo98] (these are especially relevant to the average case analysis of many algorithms); random hypergraphs [FrJa95], [KrVu01]; random digraphs [Ka90]; random subgraphs of the cube [Ri00]; superposition models Wo99], [GrJaKiWo02]; k-in, k-out models [CoFr94]; random mappings [Ko86]; random instances of k-SAT [BoBoChKiWi01]; random graphs with independent edges but of unequal probabilities [LuSh95]; random maps (graphs embedded on surfaces) [GaWo00].

8.2.8 Random Graph Processes

A random graph process is a family of random graphs indexed by time. Some random graph processes are useful in proofs of facts about standard random graph models, but the following one in particular has been studied because of the interesting interpretation and extensions of threshold results which it enables.

DEFINITIONS

D19: The ***standard random graph process*** $\tilde{\mathcal{G}}(n)$ begins with no edges and adds new edges one at a time, each selected uniformly at random from those not already present. (Formally, $\tilde{\mathcal{G}}(n)$ is a Markov chain. Its state at time m (when it has m edges) is clearly equivalent to the random graph $\mathcal{G}(n, m)$, and so it is commonly represented as the sequence $\{\mathcal{G}(n, m)\}_{m \geq 0}$.)

D20: The ***hitting time*** of a graph property Q is $\min\{m : \mathcal{G}(n, m)$ has Q$\}$.

FACTS

As before, asymptotic statements regarding $\tilde{\mathcal{G}}(n)$ refer to the passage of n to infinity. A number of statements have been proved which show that the first edge giving a graph a certain property is a.a.s. also the first edge for which another (simpler) property holds.

F73: [Bo84] The hitting time for the property of possessing a Hamilton cycle is a.a.s. equal to the hitting time for having minimum degree at least 2.

F74: [BoTh85] The hitting time for possessing a perfect matching is a.a.s. equal to the hitting time for having minimum degree at least 1. Indeed, [BoFr85] the hitting time for possessing $\lfloor k/2 \rfloor$ edge-disjoint Hamilton cycles and, if k is odd, a matching of size $\lfloor n/2 \rfloor$ disjoint from these cycles, is a.a.s. equal to the hitting time for having minimum degree at least k.

F75: [BoTh85] For any function $k = k(n)$, the hitting time for being k-connected is a.a.s. equal to the hitting time for having minimum degree at least k.

F76: Let F_n denote the length of the first cycle to appear in $\tilde{\mathcal{G}}(n)$. Then [Ja87] for fixed $j \geq 3$, $\mathbf{P}(F_n = j) \sim p_j = \frac{1}{2}\int_0^1 t^{j-1}e^{t/2+t^2/4}\sqrt{1-t}\,dt$; $\sum_{j=3}^{\infty} p_j \sim 1$, but on the other hand [FlKnPi89] the expected value of F_n is asymptotic to $n^{1/6}$.

F77: [JaKnLuPi93] The probability that $\tilde{\mathcal{G}}(n)$ at no time contains more than one complex component is $(1+o(1))\frac{5\pi}{18}$.

REMARK

R19: A few of the other random graph processes of interest include processes which were motivated by the world-wide-web [BoRiSpTu01], processes modeling the growth of the giant component [AlPi00], processes which randomly delete from a graph in order to find packings [AlKiSp97], and simple processes for generating graphs with given maximim degree [RuWo92], but these are only a small representative sample.

References

[AcFr99] D. Achlioptas and E. Friedgut, A sharp threshold for k-colorability, *Random Structures Algorithms* 14 (1999), 63–70.

[AjKoSz81] M. Ajtai, J. Komlós, and E. Szemerédi, The longest path in a random graph, *Combinatorica* 1 (1981), 1–12.

[AlPi00] D. J. Aldous and B. Pittel, On a random graph with immigrating vertices: emergence of the giant component, *Random Structures and Algorithms* 17 (2000), 79–102.

[Al93] N. Alon, Restricted colorings of graphs, pp. 1–33 in *Surveys in Combinatorics 1993,* K. Walker (Ed.), Cambridge University Press, 1993.

[AlKiSp97] N. Alon, J. H. Kim, and J. Spencer, Nearly perfect matchings in regular simple hypergraphs, *Israel J. Math.* 100 (1997), 171–187.

[ArGr95] D. Archdeacon and D.A. Grable, The genus of a random graph, *Discrete Math.* 142 (1995), 21–37.

[BaBoDeFlGaGo02] C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy and D. Gouyou-Beauchamps, Generating functions for generating trees, *Discrete Math.* 246 (2002), 29–55.

[BeCa78] E. A. Bender and E. R. Canfield, The asymptotic number of labeled graphs with given degree sequences, *J. Combin. Theory, Ser. A* 24 (1978), 296–307.

[BaHoJa92] A. D. Barbour, L. Holst and S. Janson, *Poisson Approximation.* Clarendon Press, Oxford, 1992.

[Bo79] B. Bollobás, *Graph Theory—An introductory Course*, Graduate Texts in Mathematics, Springer-Verlag, 1979.

[Bo80] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, *Europ. J. Combinatorics* 1 (1980), 311–316.

[Bo81] B. Bollobás, Threshold functions for small subgraphs, *Math. Proc. Cambridge Phil. Soc.* 90 (1981), 197–206.

[Bo81a] B. Bollobás, Degree sequences of random graphs, *Discrete Math.* 33 (1981), 1–19.

[Bo81b] B. Bollobás, The diameter of random graphs, *Trans. Amer. Math. Soc.* 267 (1981), 41–52.

[Bo84] B. Bollobás, The evolution of sparse graphs. In *Graph Theory and Combinatorics,* B. Bollobás (Ed.), Academic Press, pp. 35–57 (1984).

[Bo88] B. Bollobás, The chromatic number of random graphs, *Combinatorica* 8 (1988), 49–56.

[Bo01] B. Bollobás, *Random Graphs*, (2nd edn.) Cambridge Studies in Advanced Mathematics, 73. Cambridge University Press, 2001.

[BoBoChKiWi01] B. Bollobás, C. Borgs, J. Chayes, J. H. Kim, and D. B. Wilson, The scaling window of the 2-SAT transition, *Random Structures and Algorithms* 18 (2001), 201–256.

[BoFe82] B. Bollobás and W. Fernandez de la Vega, The diameter of random regular graphs, *Combinatorica* 2 (1982), 125–134.

[BoFr85] B. Bollobás and A. Frieze, On matchings and Hamiltonian cycles in random graphs. In *Random Graphs '83,* M. Karoński and A. Ruciński (Eds.), North-Holland, Amsterdam, pp. 23–46 (1985).

[BoRiSpTu01] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády, The degree sequence of a scale-free random graph process, *Random Structures and Algorithms* 18 (2001), 279–290.

[BoTh85] B. Bollobás and A. Thomason, Random graphs of small order, pp. 47–97 in *Random Graphs '83,* M. Karoński and A. Ruciński (Eds.), North-Holland, 1985.

[BoTh87] B. Bollobás and A. Thomason, Threshold functions, *Combinatorica* 7 (1987), 35–38.

[BrFrSuUp99] A. Z. Broder, A. Frieze, S. Suen, and E. Upfal, Optimal construction of edge-disjoint paths in random graphs, *SIAM Journal on Computing* 28 (1999), 541–574.

[CoFr94] C. Cooper and A. Frieze, Hamilton cycles in a class of random directed graphs, *J. Combinatorial Theory, Ser. B* 62 (1994), 151–163.

[CoFrRe02] C. Cooper, A. Frieze, and B. Reed, Random regular graphs of non-constant degree: connectivity and Hamilton cycles, *Combinatorics, Probability and Computing* 11 (2002), 249–262.

[CoFrReRi02] C. Cooper, A. Frieze, B. Reed, and O. Riordan, Random regular graphs of non-constant degree: independence and chromatic number, *Combinatorics, Probability and Computing* 11, 323–342.

[ErRé60] P. Erdős and A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.* 5 (1960), 17–61.

[ErRé61] P. Erdős and A. Rényi, On the strength of connectedness of a random graph, *Acta Math. Acad. Sci. Hungar.* 12 (1961), 261–267.

[ErRé66] P. Erdős and A. Rényi, On the existence of a factor of degree one of a connected random graph, *Acta Math. Acad. Sci. Hungar.* 17 (1966), 359–368.

[FlKnPi89] P. Flajolet, D. E. Knuth, and B. Pittel, The first cycles in an evolving graph, *Discrete Math.* 75 (1989), 167–215.

[FrKa96] E. Friedgut and G. Kalai, Every monotone graph property has a sharp threshold, *Proc. Amer. Math. Soc.* 124 (1997), 2993–3002.

[Fr91] J. Friedman, On the second eigenvalue and random walks in random d-regular graphs, *Combinatorica* 11 (1991), 331–362.

[Fr90] A. Frieze, On the independence number of random graphs. *Discrete Math.* 81 (1990), 171–175.

[FrJa95] A. Frieze and S. Janson, Perfect matchings in random s-uniform hypergraphs, *Random Structures and Algorithms* 7 (1995), 41–57.

[FüKo81] Z. Füredi and J. Komlós, The eigenvalues of random symmetric matrices, *Combinatorica* **1** (1981), 233–241.

[GaWo00] Z. Gao and N. C. Wormald, The distribution of the maximum vertex degree in random planar maps, *J. Combinatorial Theory, Ser. A* 89 (2000), 201–230.

[GrJaKiWo02] C. Greenhill, S. Janson, J. H. Kim, and N. C. Wormald, Permutation pseudographs and contiguity, *Combinatorics, Probability and Computing* 11 (2002), 273–298.

[Ja87] S. Janson, Poisson convergence and Poisson processes with applications to random graphs, *Stochastic Processes and their Applications* 26 (1987), 1–30.

[Ja93] S. Janson, *Orthogonal Decompositions and Functional Limit Theorems for Random Graph Statistics*, Memoirs Amer. Math. Soc. 534, Amer. Math. Soc., Providence, R.I. (1994).

[Ja94] S. Janson, Multicyclic components in a random graph process, *Random Structures and Algorithms* 4 (1993), 71–84.

[JaKnLuPi93] S. Janson, D. E. Knuth, T. Łuczak and B. Pittel, The birth of the giant component, *Random Structures and Algorithms* 4 (1993) 233–358.

[JaLuRu90] S. Janson, T. Luczak, and A. Ruciński, An exponential bound for the probability of nonexistence of a specified subgraph in a random graph, pp. 73–87 in *Random Graphs '83,* M. Karoński, J. Jaworski and A. Ruciński (Eds.), Wiley, 1990.

[JaLuRu00] S. Janson, T. Łuczak, and A. Ruciński, *Random Graphs*, Wiley, 2000.

[JaRu02] S. Janson and A. Ruciński, The infamous upper tail. *Random Structures and Algorithms* 20 (2002), 317–342.

[Ka90] R. M. Karp, The transitive closure of a random digraph, *Random Structures and Algorithms* 1 (1990), 73–93.

[KiSuVu02] J. H. Kim, B. Sudakov, and V. Vu, On the asymmetry of random regular graphs and random graphs, *Random Structures and Algorithms* 21 (2002), 216–224.

[KiWo01] J. H. Kim and N. C. Wormald, Random matchings which induce Hamilton cycles, and hamiltonian decompositions of random regular graphs, *J. Combin. Theory Ser. B* 81 (2001), 20–44.

[Ko86] V. F. Kolchin, *Random mappings.* Translation Series in Mathematics and Engineering. Optimization Software, Inc., Publications Division, New York (1986).

[Ko99] V. F. Kolchin, *Random Graphs.* Encyclopedia of Mathematics and its Applications, 53. Cambridge University Press, 1999.

[KoSz83] J. Komlós and E. Szemerédi, Limit distributions for the existence of Hamilton circuits in a random graph, *Discrete Math.* 43 (1983), 55–63.

[Kri00] M. Krivelevich, The choice number of dense random graphs, *Combinatorics, Probability and Computing* 9 (2000), 19–26.

[KrSuVuWo01] M. Krivelevich, B. Sudakov, V. H. Vu, and N. C. Wormald, Random regular graphs of high degree, *Random Structures and Algorithms* 18 (2001), 346–363.

[KrVu01] M. Krivelevich and V. Vu, Choosability in random hypergraphs, *J. Combinatorial Theory, Ser. B* 83 (2001), 241–257.

[Lu90] T. Łuczak, Component behavior near the critical point of the random graph process, *Random Structures and Algorithms* 1 (1990), 287–310.

[Lu91] T. Łuczak, The chromatic number of random graphs, *Combinatorica* 11 (1991), 45–54.

[Lu92] T. Łuczak, Sparse random graphs with a given degree sequence, pp. 165–182 in *Random Graphs Vol. 2,* A. Frieze and T. Łuczak (Eds.), Wiley, 1992.

[ŁuRu91] T. Łuczak and A. Ruciński, Tree-matchings in graph processes, *SIAM J. Discrete Math.* 4 (1991), 107–120.

[ŁuSh95] T. Łuczak and S. Shelah, Convergence in homogeneous random graphs, *Random Structures and Algorithms* 6 (1995), 371–392.

[Mc81] B. D. McKay, The expected eigenvalue distribution of a large regular graph, *Linear Algebra and its Applications* 40 (1981), 203–216.

[McWo97] B. D. McKay and N. C. Wormald, The degree sequence of a random graph. I. The models, *Random Structures and Algorithms* 11 (1997), 97–117.

[MeMo98] A. Meir and J. W. Moon, On the log-product of the subtree-sizes of random trees, *Random Structures Algorithms* 12 (1998), 197–212.

[Mo96] M. Molloy, A gap between the appearances of a k-core and a $(k+1)$-chromatic graph, *Random Structures and Algorithms* 8 (1996), 159–160.

[MoRe98] M. Molloy and B. Reed, The size of the giant component of a random graph with a given degree sequence, *Combinatorics, Probability and Computing* 7 (1998), 295–305.

[OsPrTa03] D. Osthus, H. J. Prömel, and A. Taraz, For which densities are random triangle-free graphs almost surely bipartite?, *Combinatorica* (to appear).

[Pa85] E. M. Palmer, *Graphical Evolution. An Introduction to the Theory of Random Graphs.* Wiley, 1985.

[PiSpWo96] B. Pittel, J. Spencer, and N. C. Wormald, Sudden emergence of a giant k-core in a random graph, *J. Combinatorial Theory, Ser. B* 67 (1996), 111–151.

[PrSt92] H. J. Prömel and A. Steger, The asymptotic number of graphs not containing a fixed color-critical subgraph, *Combinatorica* 12 (1992), 463–473.

[PrSt96] H. J. Prömel and A. Steger, On the asymptotic structure of sparse triangle free graphs, *J. Graph Theory* 21 (1996), 137–151.

[PrSt96a] H. J. Prömel and A. Steger, Counting H-free graphs, *Discrete Math.* 154 (1996), 311–315.

[Ri00] O. Riordan, Spanning subgraphs of random graphs, *Combinatorics, Probability and Computing* 9 (2000), 125–148.

[RoWo94] R. W. Robinson and N. C. Wormald, Almost all regular graphs are hamiltonian, *Random Structures Algorithms* 5 (1994), 363–374.

[RöRu95] V. Rödl and A. Ruciński, Threshold functions for Ramsey properties, *J. American Math. Soc.* 8 (1995), 253–270.

[RöTh95] V. Rödl and R. Thomas, On the genus of a random graph, *Random Structures Algorithms* 6 (1995), 1–12.

[Ru88] A. Ruciński, When are small subgraphs of a random graph normally distributed? *Probability Theory and Related Fields* 78 (1988), 1–10.

[RuWo92] A. Ruciński and N.C. Wormald, Random graph processes with degree restrictions, *Combinatorics, Probability and Computing* 1 (1992), 169–180.

[ShSp88] S. Shelah and J. Spencer, Zero-one laws for sparse random graphs, *J. Amer. Math. Soc.* 1 (1988), 97–115.

[Sp90] J. Spencer, Threshold functions for extension statements, *J. Combin. Theory Ser. A* 53 (1990), 286–305.

[Sp01] J. Spencer, *The Strange Logic of Random Graphs.* Algorithms and Combinatorics, 22. Springer-Verlag, Berlin, 2001.

[SpTó02] J. Spencer and G. Tóth, Crossing numbers of random graphs, *Random Structures and Algorithms* 21 (2002), 347–358.

[Su90] W. C. S. Suen, A correlation inequality and a Poisson limit theorem for nonoverlapping balanced subgaphs of a random graph, *Random Structures and Algorithms* 1 (1990), 231–242.

[Vu01] V. Vu, A large deviation result on the number of small subgraphs of a random graph, *Combin. Probab. Comput.* 10 (2001), 79–94.

[Wo81] N. C. Wormald, The asymptotic distribution of short cycles in random regular graphs, *J. Combinatorial Theory. Ser. B*, 31 (1981), 168–182.

[Wo96] N. C. Wormald, The perturbation method for triangle-free random graphs, *Random Structures and Algorithms* 9 (1996), 253–270.

[Wo99] N. C. Wormald, Models of random regular graphs, pp. 239–298 in *Surveys in Combinatorics, 1999,* J. D. Lamb and D. A. Preece (Eds.), Cambridge University Press, 1999.

[Wr70] E. M. Wright, Graphs on unlabelled nodes with a given number of edges, *Acta Math.* 126 (1970), 1–9.

* Supported by the Canada Research Chairs program.

8.3 RAMSEY GRAPH THEORY

Ralph Faudree, University of Memphis

8.3.1 Ramsey's Theorem
8.3.2 Fundamental Results
8.3.3 Classical Ramsey Numbers
8.3.4 Generalized Ramsey Numbers
8.3.5 Size Ramsey Numbers
8.3.6 Ramsey Minimal Graphs
8.3.7 Generalizations and Variations
References

Introduction

In any group of six people there are always either three who know each other or three mutual strangers. This same statement in the language of graph theory is that if each edge of a complete graph K_6 is colored either Red or Blue, then there is either a Red triangle (K_3) or a Blue triangle (K_3). Moreover, this conclusion is not true for K_5, so six is a minimum such number. This is a special case of a much more general observation of F. P. Ramsey [Ra30]. He observed that for all positive integers m and n, there is an integer r such that if each edge of a K_r is colored either Red or Blue, then there will be either a Red K_m or a Blue K_n. The smallest such integer r is denoted by $r(m,n)$, and is called the (m,n)-Ramsey number. Ramsey graph theory is the study of such numbers and the corresponding graphs. More generally, the number of colors is not restricted to just two, the monochromatic graphs are arbitrary — not just complete graphs, and the graph being edge-colored is not restricted to being complete.

8.3.1 Ramsey's Theorem

Ramsey's original theorem applies to general set theory and has implications to many areas of mathematics other than combinatorics and graph theory. For combinatorial results related to Ramsey's theorem, see [GrRoSp90], [Pa78], [Ne96], and survey articles [Bu74], [Bu79], [ChGr83], and [GrRo87]. A simplified version of Ramsey's Theorem applicable to finite graphs is our starting point.

FACT

F1: (**Ramsey's Theorem** [Ra30]) Given positive integers $k, n_1, n_2, \cdots, n_k \geq 2$, there is a least positive integer $r(n_1, n_2, \cdots, n_k)$ such that, for any partition $\mathcal{C}_1, \mathcal{C}_2, \cdots \mathcal{C}_k$ of the edges of a complete graph K_p with $p \geq r(n_1, n_2, \cdots, n_k)$, there is for some i a complete subgraph K_{n_i} all of whose edges are in $\mathcal{C}_i$.

DEFINITIONS

D1: The number $r(n_1, n_2, \cdots, n_k)$ is called the ***Ramsey number*** for the k-tuple $(n_1, n_2, \cdots, n_k)$.

D2: The partition of the edges of a complete graph K_p into k sets is described as a coloring of the edges of K_p with k colors, or more specifically a ***k-edge-coloring*** of K_p.

Ramsey Numbers for Arbitrary Graphs

Ramsey's theorem implies the existence of a "monochromatic" complete subgraph in the appropriate color in any edge-coloring of a sufficiently large complete graph. Since any graph G on m vertices is isomorphic to a subgraph of K_m, an immediate consequence of Ramsey's Theorem is the existence of the Ramsey numbers for arbitrary graphs.

DEFINITIONS

D3: The ***(generalized) Ramsey number*** $r(G_1, G_2, \cdots, G_k)$ for any collection of k graphs $\{G_1, G_2, \cdots, G_n\}$ is the least positive integer n such that for any k-edge-coloring of K_n, there is for some i a monochromatic copy of G_i in color i.

D4: Given $k \geq 2$ and graphs $G_1, G_2, \cdots, G_k$, a graph F is said to ***arrow*** the k-tuple $(G_1, G_2, \cdots, G_k)$ if for any k-edge-coloring of F there is for some i a monochromatic copy of G_i in the i^{th} color. This is denoted by $F \longrightarrow (G_1, G_2, \cdots, G_n)$. Thus, the Ramsey number $r(G_1, G_2, \cdots, G_k)$ is the smallest order of a graph F such that $F \longrightarrow (G_1, G_2, \cdots, G_n)$.

D5: The ***size Ramsey number*** $\hat{r}(G_1, G_2, \cdots, G_k)$ is the smallest size (i.e., number of edges) of a graph F such that $F \longrightarrow (G_1, G_2, \cdots, G_n)$.

D6: A graph F is $(G_1, G_2, \cdots, G_k)$-***minimal*** if $F \longrightarrow (G_1, G_2, \cdots, G_n)$, but no proper subgraph of F also arrows.

REMARKS

R1: If for each i, $G_i = K_{n_i}$, then $r(n_1, n_2, \cdots, n_k) = r(K_{n_1}, K_{n_2}, \cdots, K_{n_k})$.

R2: Classical graph Ramsey theory deals with the case when each of the required monochromatic graphs is complete, while generalized graph Ramsey theory involves the generalization to arbitrary graphs.

R3: This leads to asking questions about the structure of and the number of "different" graphs that "arrow". Ramsey minimal graphs are considered in §8.3.6.

8.3.2 Fundamental Results

The vast majority of Ramsey graph results concern 2-colorings, so these are featured. Several useful facts are immediate consequences of the definition.

FACTS

F2: For any pair of graphs G_1 and G_2, $r(G_1, G_2) = r(G_2, G_1)$. More generally, the order of the graphs is not important for any number of graphs.

F3: For any graphs G_m and G_n of orders m and n respectively, $r(G_m, G_n) \leq r(m, n)$.

F4: For $n \geq 2$, $r(2, n) = r(n, 2) = n$.

F5: **Erdős and Szekeres** [ErSz35] For $m, n \geq 3$,

$$r(m, n) \leq r(m-1, n) + r(m, n-1)$$

with strict inequality if both $r(m-1, n)$ and $r(m, n-1)$ are even. A consequence of this is

$$r(m+1, n+1) \leq \binom{m+n}{m}$$

REMARKS

R4: The Erdos-Szekeres theorem gives a finite upper bound for the Ramsey numbers of all pairs of finite graphs. There are corresponding bounds for any finite number of colors and collections of finite graphs.

R5: To prove that the Ramsey number $r(G, H) = p$, normally two steps are taken. A proof is given to show that any 2-edge-coloring of K_p, say a Red-Blue coloring, yields either a Red G or a Blue H, and then a Red-Blue coloring of a K_{p-1} is exhibited that has neither a Red G nor a Blue H.

R6: In the case of a 2-edge-coloring, say with Red and Blue, it is sometimes more convenient to just denote a subgraph F, which represents the subgraph induced by the Red edges. Then, the complement $\overline{F}$ of F denotes the Blue subgraph.

EXAMPLE

E1: To show that $r(3, 3) = 6$, observe that in Figure 8.3.1 there is a Red-Blue coloring of K_5 with no K_3 in either color, and observe by the result of Erdős and Szekeres

$$r(3, 3) \leq r(2, 3) + r(3, 2) = 3 + 3 = 6.$$

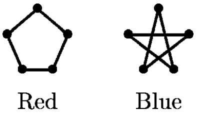


Figure 8.3.1 $r(K_3, K_3) > 5$.

8.3.3 Classical Ramsey Numbers

Determining classical Ramsey numbers is quite difficult, and the number of nontrivial classical Ramsey numbers that are known precisely is very limited. Only one nontrivial multicolor (at least three colors) classical Ramsey number is known and only nine nontrivial two color Ramsey numbers $r(m, n)$ are known, which is strong evidence of the difficulty in determining Ramsey numbers.

Table 8.3.1 The classical Ramsey numbers $r(k,l)$.

l l k		3	4	5	6	7	8	9	10	11	12	13	14	15
	3	6	9	14	18	23	28	36	43 40	51 46	59 52	69 59	78 66	88 73
3		4	18	25	41 35	61 49	84 56	115 69	149 80	191 96	238 128	291 133	349 141	417 153
4	GreGl 55	GreGl 55	5	49 43	87 58	143 80	216 95	316 121	442 141	153	181	193	1280 221	2375 242
5	GreGl 55	McRd 95 Ka65	McRd 97 Ex89b	6	165 102	298 111	495 127	780 153	1171 177	253	262	278	292	374
6	Ke64	McRd 97 Ex93	HuZ 98 Ex93		7	540 205	1031 216	1713	2826					
7	GrYa 68 Ka66	M94 Ex89				8	1870 282	3583	6090 316					
8	McZ 92 GriRb 82	M94 Ex89e					9	6588 565	12677 580					
9	GriRb 82 Ka66	M94 RdK 88						10	23556 798					

Ramsey Numbers for Small Graphs

Table 8.3.1 contains the known classical Ramsey numbers $r(k,l)$ along with the best upper and lower bounds for small values of m and n. The argument k ranges from 3 through 10 and runs along the main diagonal, and the argument l runs along the top row. The exact Ramsey values are centered, upper bounds are the top entries, and lower bounds are the bottom entries. For instance, $r(3,8) = 28$, and $r(4,8)$ is between 56 and 84. The references for each of the exact values are also listed in the table along with references for some of the upper and lower bounds, where the leftmost column is used to indicate the argument l for the particular Ramsey number. For instance, K64 established that $r(3,6) = 18$. A listing of Ramsey numbers much more complete than Table 8.3.1 can be found in an electronic paper by Radzisowski [Ra02].

FACT

F6: **Greenwood and Gleason** [GrGl55] $r(3,3,3) = 17$.

REMARK

R7: The earlier work determining classical Ramsey numbers was done by Greenwood and Gleason [GrGl55], Kery [Ke64], and by Graver and Yackel [GrYa68]. Lower bounds were established with explicit but sophisticated colorings using algebraic techniques, and upper bounds were established using graph theory techniques. More recently Exoo, McKay, and Radzisowski, among other mathematicians, have used computational techniques, powerful algorithms, and more computing power to determine larger Ramsey numbers and to sharpen upper and lower bounds. However, the gaps between upper and lower bounds are still enormous for even many small values of m and n.

Asymptotic Results

Considerable study of the asymptotic behavior of the Ramsey number $r(m,n)$ has not yielded sharp asymptotic results. There have been numerous improvements in the upper bound of Erdős and Szekeres (Fact F5). The lower bounds for $r(m,n)$, or in particular for $r(n,n)$, are far from the upper bounds. The only case in which sharp asymptotic results have been obtained is $m = 3$. Shearer [Sh83] proved an upper bound for $r(3,n)$, and Kim [Ki95] verified an asymptotically sharp lower bound.

FACTS

F7: [Th88] For all $1 \leq m \leq n$, there is an absolute constant c such that

$$r(m+1, n+1) \leq e^{(-m \log n)/2n + c\sqrt{\log m}} \binom{m+n}{m}$$

and in particular

$$r(n+1, n+1) \leq n^{-1/2 + c\sqrt{\log n}} \binom{2n}{2}$$

F8: For some constant c' and for all positive n,

$$r(n,n) \geq c' n 2^{n/2}$$

F9: ([Sh83], [Ki95]) There are absolute constants c and c' such that

$$cn^2/\log n \leq r(3,n) \leq c' n^2/\log n$$

Probabilistic techniques are used to prove the existence of colorings, but specific colorings are not exhibited.

8.3.4 Generalized Ramsey Numbers

There has been more activity and considerably more results in generalized Ramsey theory than in any other area of graph Ramsey theory. It would be impossible to survey even a fraction of the results, so we will review just a few of the highlights.

Initial Generalized Ramsey Results

FACTS

F10: [GeGy67]For positive integers $2 \leq m \leq n$,

$$r(P_m, P_n) = n + \lfloor m/2 \rfloor - 1$$

The lower bound for $r(P_m, P_n)$ comes from the coloring determined by the graph $F = K_{n-1} \cup K_{\lfloor m/2 \rfloor - 1}$. The graph F contains no P_n, and the complement $\overline{F}$ is a bipartite graph with no P_m.

F11: [BuRo73] Let $n_1, n_2, \cdots n_k$ be positive integers with s of them being even and $k \geq 2$. Then

$$r(K_{1,n_1}, K_{1,n_2}, \cdots, K_{1,n_k}) = \sum_{i=1}^{k}(n_i - 1) + \alpha$$

where $\alpha = 1$ if s is positive and even and $\alpha = 2$ otherwise. For $k = 2$,

$$r(K_{1,m}, K_{1,n}) = m + n - \epsilon$$

where $\epsilon = 1$ if m and n are both even and 0 otherwise.

REMARK

R8: There are regular graphs of any order m and degree $k < m$ except when k and m are odd, and, in this case, nearly regular graphs exist. The lower bounds for these Ramsey numbers of Fact 11 depend on colorings derived directly from regular or nearly regular graphs.

Ramsey Numbers for Trees

There are many classes of pairs of trees for which the Ramsey number $r(T_m, T_n)$ is not known. However, for all such numbers that are known, $r(T_m, T_n) \leq m + n - 2$.

FACT

F12: [BuRo73] When at least one of m or n is even and T_m and T_n are stars ($T_m = K_{1,m-1}$ and $T_n = K_{1,n-1}$),

$$r(T_m, T_n) = m + n - 2$$

CONJECTURES

C1: **Tree Conjecture** [BuEr76] For any trees T_m and T_n with $m, n \geq 2$,

$$r(T_m, T_n) \leq m + n - 2$$

C2: **Erdős-Sós Conjecture** Any graph G with n vertices and at least $n(k-2)/2+1$ edges contains any tree T_k $(k \geq 2)$ as a subgraph.

REMARK

R9: The Erdős-Sós conjecture implies the Tree Conjecture.

Cycle Ramsey Numbers

The Ramsey numbers of cycle graphs appear to have some of the same characteristics as the Ramsey numbers of trees.

CONJECTURE

C3: **Bondy and Erdős Conjecture** [BoEr73]) For $n \geq 5$ and odd,

$$r(C_n, C_n, C_n) = 4n - 3$$

EXAMPLE

E2: For $n \geq 3$ and odd, consider the 3-edge-coloring, say with Red, Blue, and Green, of a $K_{4(n-1)}$. The Red subgraph is $4K_{n-1}$, the Blue subgraph is isomorphic to $2K_{n-1,n-1}$ and contains all of the edges between the first two and the last two of the complete graphs in Red, and the remaining edges are Green and form a $K_{2(n-1),2(n-1)}$. For n odd there is no Red C_n, since no component of the Red subgraph has n vertices, and there is no Blue or Green C_n since these graphs are bipartite and have no odd cycles. Thus, $r(C_n, C_n, C_n) > 4n - 4$ for n odd.

FACTS

F13: ([Ro73a], [Ro73b], [FaSc74]) If $3 \leq m \leq n$ with $(m,n) \neq (3,3), (4,4)$, then

$$r(C_m, C_n) = \begin{cases} 2n-1 & \text{when } m \text{ is odd,} \\ n + \frac{m}{2} - 1 & \text{when } m \text{ and } n \text{ are even, and} \\ \max\{n + \frac{s}{2} - 1, 2m - 1\} & \text{when } m \text{ is even and } n \text{ is odd} \end{cases}$$

For 3-edge-colorings the examples for cycle Ramsey numbers have similar properties, but determination of the numbers is much more difficult.

F14: [RoYa92] $r(C_5, C_5, C_5) = 17$.

F15: [FaScSc] $r(C_7, C_7, C_7) = 25$

F16: [Lu99] For all $n \geq 4$, $r(C_n, C_n, C_n) \leq (4 + o(1))n$.

Good Results

Results of [BoEr73] and [Ch77] on complete graphs and trees (see Facts 17 and 18) motivated new lines of investigation into generalized Ramsey numbers.

DEFINITIONS

D7: [Bu81] If $\chi(G)$ is the chromatic number of G, then the ***chromatic surplus*** of G is the largest number $s = s(G)$ such that in every vertex coloring of G with $\chi(G)$ colors, every color class has at least s vertices.

D8: [Bu81] A connected graph H of order $n \geq s(G)$ is called G-***good*** if

$$r(G,H) = (\chi(G)-1)(n-1)+s(G)$$

FACTS

F17: (**Bondy and Erdős** [BoEr73]) If $m \geq 3$ and $n \geq m^2-2$, then

$$r(K_m, C_n) = (m-1)(n-1)+1$$

It was conjectured in [EFRS78b] that Fact 17 is true for $n \geq m$ except for $n = m = 3$, and also verified for $m = 3$. It has now been verified for $m = 4, 5, 6$, and in addition proved in [Ni03p] for $n \geq 4m+2$.

F18: (**Chvátal** [Ch77]) For integers $m, n \geq 1$,

$$r(K_m, T_n) = (m-1)(n-1)+1$$

This theorem, which can be stated as any tree T_n is K_m-good, has been generalized in many ways. The two main approaches have been to replace K_m by a graph with chromatic number m or to replace the tree T_n by a connected sparse graph.

F19: [Bu81] If H is any connected graph of order $n \geq s(G)$, then

$$r(G,H) \geq (\chi(G)-1)(n-1)+s(G)$$

F20: [BuFa93]A graph G satisfies $r(G, T_n) = (m-1)(n-1)+1$ for all trees T_n of sufficiently large order n, if and only if $s(G) = 1$, and there is a $\chi(G)$-vertex coloring of G such that the graph induced by two of the color classes is a subgraph of a matching.

F21: Let G be an arbitrary graph and H a connected graph of order n. Then there are positive constants c, c_1, c_2 and α such that H is G-good if n is sufficiently large and

(i) [BEFRS80a] $G = K_3$, $n \geq 4$, and $|E(H)| \leq (17n+1)/15$, or
(ii) [BEFRS80a] $G = K_m$, $m \geq 4$, $|E(H)| \leq n + cn^{2/(m-1)}$, or
(iii) [EFRS85] $|E(H)| \leq n + c_1 n^{\alpha}$, and $\Delta(H) \leq c_2 n^{\alpha}$, or
(iv) [BEFRS82b] $G = C_{2m+1}$ and $|E(H)| \leq (1+c_3)n$, or
(v) [BuEr83] $G = K_3$ and $H = K_1 + C_n$ (wheel), or
(vi) [FaRoSh91] $G = C_{2m+1}$ and $H = K_2 + \overline{K_{n-2}}$.

EXAMPLES

E3: Consider a Red-Blue coloring of a $K_{(m-1)(n-1)}$ in which the Blue graph is $m-1$ vertex disjoint copies of a complete graph K_{n-1} ($(m-1)K_{n-1}$) and the Red graph is the complementary graph, $K_{n_1, n_2, \cdots, n_{m-1}}$, where $n_1 = n_2 = \cdots n_{m-1} = n-1$. There is no Blue T_n, and in fact no Blue connected graph with n vertices, and there is no Red K_m.

E4: The 2-edge-coloring of Example E3 gives the lower bound for $r(K_m, T_n)$ and also for $r(K_m, C_n)$. Moreover, there is no graph with chromatic number m in the Red graph. This coloring implies that if the chromatic number $\chi(G) = m$ and H is any connected graph of order n, then $r(G,H) > (m-1)(n-1)$.

E5: For $p = (\chi(G) - 1)(n - 1) + s(G) - 1$, consider the Red-Blue edge-coloring of K_p in which the Blue graph consists of $\chi(G) - 1$ disjoint complete graphs of order $n - 1$ and one complete graph of order $s(G) - 1$, and the Red graph is the complementary graph. There is no Blue G and there is no connected graph of order n in Red for $n \geq s(G)$.

REMARK

R10: A comprehensive summary of "good" results can be found in [FaRoSc92].

Small Order Graphs

Most of the generalized Ramsey numbers for very small order graphs were determined in papers by Chvátal and Harary [ChHa72], Clancy [Cl77], and Hendry [He89a]. Figure 8.3.2 pictures all of the graphs with at most five vertices that have no isolated vertices. The graphs are described using standard graphical operations such as $+, -$ and $\cup$ along with $\bullet$, where $G \bullet H$ is a graph (not unique) obtained from G and H by identifying one vertex from each graph.

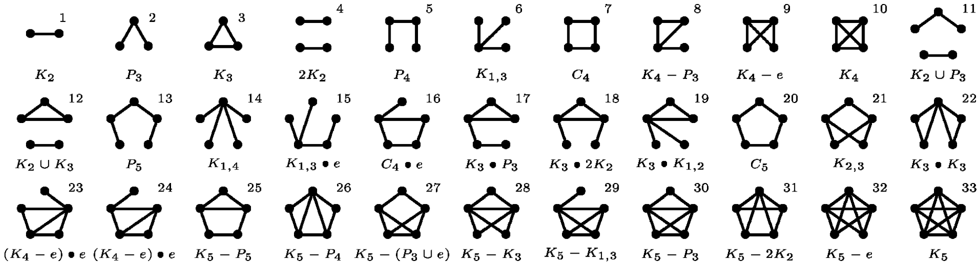


Figure 8.3.2 Graphs of order ≤ 5 without isolates.

REMARKS

R11: Table 2 gives the diagonal Ramsey numbers or the sharpest known bounds for the Ramsey numbers for all graphs of order at most five without isolated vertices. Additional information on these exact numbers and the bounds can be found in [Rd02].

R12: The Ramsey numbers for several other classes of small order graphs have been determined, and there is an excellent survey of this type of result in [Ra02] .

R13: Ramsey numbers for the pair (K_3, G) where G is an arbitrary graph of order p have been determined for $p \leq 6$ in [FaRoSc80], for $p = 7$ or 8 in [Bri98], and for $p = 9$ in [BrBrHa98].

R14: The diagonal Ramsey numbers $r(G, G)$ for all graphs with at most seven edges and without isolated vertices can be found in [He87].

R15: Ramsey numbers for almost all pairs (G, T) where G is a connected graph of order at most 5 and T is an arbitrary tree were calculated in [FaRoSc88].

Table 8.3.2 Generalized Ramsey numbers for small order graphs.

$\hat{r}$		P_3	K_3	$2K_2$	P_4	$K_{1,3}$	C_4	$K_4{-}P_3$	$K_4{-}e$	K_4
	P_3	$3_{[2]}$	$8_{[4]}$	$4_{[2]}$	5	$4_{[2]}$	$6_{[66]}$	8	9	$18_{[4]}$
P_3	$K_{1,3}$	K_3	$15_{[3]}$	6	15	18	20	19	20	$36_{[3]}$
K_3	$K_5{-}2k_2$	K_6	$2K_3$	3	5	6	8	8	10	12
$2K_2$	C_4	$2K_3$	$3K_2$	P_4	$7_{[5]}$	8	9	15	17	35*
P_4	$K_4{-}K_2$	$K_7{-}P_7$	C_5	$K_5{-}P_4$	$K_{1,3}$	$5_{[2]}$	$12_{[6]}$	18	19	42*
$K_{1,3}$	$K_{1,4}$	$K_7{-}K_3$	$2K_{1,3}$	L	$K_{1,5}$	C_4	$15_{[6]}$	20	21	45*
C_4	K_4	$K_7{-}K_2$	$2C_4$	$K_5{-}K_2$	$K_{2,6}$	K_6	$K_4{-}P_3$	$19_{[5]}$	20	43*
$K_4{-}P_3$	$K_5{-}2K_2$	$K_7{-}P_3$	$2(K_4{-}P_3)$	$K_7{-}P_7$	$K_7{-}K_3$	$K_7{-}K_2$	$K_7{-}P_3$	$K_4{-}e$	$39^*_{[5]}$	55*
$K_4{-}e$	$K_5{-}K_2$	$K_7{-}K_2$	$2(K_4{-}e)$	$K_7{-}P_6$	$K_7{-}P_3$	K_7	$K_7{-}K_2$	$K_{10}{-}K_4$	K_4	$153_{[3]}$
K_4	$K_7{-}3K_2$	K_9	$2K_4$	$K_{10}{-}2C_5$	$K_{10}{-}K_3$	K_{10}	$K_{10}{-}P_3$	K_{11}	K_{18}	

Linear Bounds

By using Szemerédi's regularity lemma, [CRST83] verified the following conjecture of Erdős. Chen and Schelp subsequently extended the class of "linearly bounded" graphs to a larger class of c-arrangeable graphs, and this extension has some nice applications.

CONJECTURE

C4: **Erdős Conjecture** [BuEr75] If G is a graph of order n with maximal degree Δ, then $r(G, G)$ has an upper bound that is linear in n.

DEFINITION

D9: A graph G is ***c-arrangeable*** if the vertices of G can be ordered in such a way that for any vertex v, each adjacency of v that succeeds v in the order has at most c adjacencies that precede v in the order.

FACTS

F22: [CRST83] if G is a graph of order n with maximal degree Δ, then $r(G, G) \leq c{\cdot}\Delta{\cdot}n$ for some positive constant c. (This verifies the Erdős Conjecture.)

F23: [ChSc93] If G is a c-arrangeable graph of order n, then there is an upper bound for $r(G, G)$ that is linear in n.

F24: [ChSc93] (Corollary) If G is a planar graph, then $r(G, G)$ has an upper bound that is linear in the order of G.

F25: [RoTh96] If G is a graph of bounded genus, then $r(G, G)$ has an upper bound that is linear in the order of G.

CONJECTURE

C5: **Bounded Density Conjecture** [BuEr75] If the average degree of each subgraph of a graph G of order n is at most c', then there is a constant $c = c(c')$ such that $r(G, G) \leq cn$.

8.3.5 Size Ramsey Numbers

Increased interest in the size Ramsey number $\hat{r}(G, H)$ was created in a paper by Erdős, et al. [EFRS78].

General Bounds

FACTS

F26: $|E(G)| + |E(H)| - 1 \leq \hat{r}(G, H) \leq \binom{r(G,H)}{2}$.

F27: [EFRS78] For $m, n \geq 1$,

(i) $\hat{r}(K_m, K_n) = \binom{r(m,n)}{2}$, and

(ii) $\hat{r}(K_{1,m}, K_{1,n}) = m + n - 1$.

REMARKS

R16: Any graph F such that $F \longrightarrow (G, H)$ must have at least $|E(G)| + |E(H)| - 1$ edges, and the fact that $K_{r(G,H)} \longrightarrow (G, H)$ implies that $\hat{r}(G, H) \leq \binom{r(G,H)}{2}$.

R17: It is natural to investigate the relationship that $\hat{r}(G, H)$ has with both $\binom{r(G,H)}{2}$ and $|E(G)| + |E(H)| - 1$. Both extreme possibilities occur.

Linear Bounds

The size Ramsey $\hat{r}(K_{1,m}, K_{1,n})$ is linear in m and n, while the number of edges in the complete Ramsey graph for the pair $(K_{1,m}, K_{1,n})$ is quadratic in m and n. Beck ([Be83], [Be90]) answered some of the questions posed in [EFRS78] by showing that there were large classes of graphs for which the size Ramsey number has a linear property or near linear property.

FACTS

F28: [Be83], [Be90] There exist constants c and c' such that for any tree T_n of order n and maximum degree Δ and for n sufficiently large,

(i) $\hat{r}(P_n, P_n) \leq cn$,

(ii) $\hat{r}(C_n, C_n) \leq c'n$, and

(iii) $\hat{r}(T_n, T_n) \leq \Delta n(\log n)^{12}$.

F29: [HaKo95] For any tree T_n with maximum degree Δ, then there is a constant c such that $\hat{r}(T_n, T_n) \leq c \cdot \Delta \cdot n$.

REMARK

R18: The previous result was conjectured in [Be90], along with a stronger conjecture dealing with the bipartite structure of the tree T_n. The previous results led Beck and Erdős to make the following conjecture.

CONJECTURE

C6: **Beck Conjecture** [Be90] For a graph G of order n and bounded degree Δ, there is a constant $c = c(\Delta)$ such that $\hat{r}(G,G) \leq cn$.

Bipartite Graphs

For the complete bipartite graph $K_{n,n}$, upper bounds were proved by Erdős, et al. [EFRS78] and Nešetřil and Ródl [NeRo78] and lower bounds were proved by Erdős and Rousseau [ErRo93], but none are asymptotically sharp.

DEFINITION

D10: A star forest with s components each being a star with n edges will be denoted by $sK_{1,n}$.

FACTS

F30: [EFRS78], [NeRo78], [ErRo93] For $n \geq 6$,

$$(1/60)n^2 2^n < \hat{r}(K_{n,n}, K_{n,n}) < (3/2)n^3 2^n$$

F31: [BEFRS78] For positive integers m, n, s and t,

$$\hat{r}(sK_{1,m}, tK_{1,n}) = (m+n-1)(s+t-1)$$

REMARKS

R19: Only a limited number of precise values of size Ramsey numbers are known, since they are much more difficult to calculate than generalized Ramsey numbers. Star forests with all components equal is one class of graphs for which many numbers are known.

R20: The precise value of the size Ramsey numbers for general families of star forests is still open, but results in special cases support the following conjecture from [BEFRS78].

CONJECTURE

C7: **Star Forest Conjecture** [BEFRS78] Let s and t be positive integers with $m_1 \geq m_2 \cdots m_s \geq 1$ and $n_1 \geq n_2 \geq \cdots n_t \geq 1$, and let $F_1 = \cup_{i=1}^{s} K_{1,m_i}$ and $F_2 = \cup_{j=1}^{t} K_{1,n_j}$. Then,

$$\hat{r}(F_1, F_2) = \sum_{k=2}^{s+t} p_k$$

where $p_k = \max\{m_i + n_j - 1 \ : \ i + j = k\}$.

Small Order Graphs

Exact size Ramsey numbers are known for some small graphs, but even for small graphs it is sometimes difficult to calculate the number. An easier number to calculate is the restricted size Ramsey number, which is determined by restricting the "arrowing" graphs to those whose order is the Ramsey number.

Table 8.3.3, part 1. Size Ramsey Numbers

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	$\boxed{1}$	2	3	3	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5
1		$\boxed{2}$	3	5	4	4	5	4	5	7	7	5	5	5	5	5	5	5	5	5	5
2			$\boxed{3}$	6	5	7	7	7	7	7	9	6	6	9	9	9	9	9	9	9	9
3				$\boxed{4}$	5	5	5	5	5	5	6	6	6	6	6	6	6	6	6	6	6
4					$\boxed{5}$	5	5	5	7	7	10	6	7	6	7	6	6	6	7	7	7
5						$\boxed{6}$	6	6	7	7	10	5	7	5	7	6	6	7	7	7	7
6							$\boxed{7}$	6	7	7	10	6	7	6	7	6	6	7	7	7	7
7			*CH*					$\boxed{8}$	7	7	10	6	7	9	9	9	9	9	9	9	9
8									$\boxed{9}$	10	11	6	7	9	9	9	9	9	9	9	9
9										$\boxed{10}$	18	7	9	13	13	13	13	13	13	13	13
10											$\boxed{11}$	6	7	6	6	6	6	7	6	6	7
11	’’’	’’’	’’’	’’’	’’’	’’’	’’’	’’’	’’’	’’’	’’’	$\boxed{12}$	7	9	9	9	9	9	9	9	9
12											|		$\boxed{13}$	6	7	6	6	9	9	9	9
13											|			$\boxed{14}$	7	9	7	9	9	9	9
14											|				$\boxed{15}$	6	6	9	9	9	9
15											|					$\boxed{16}$	6	9	9	9	9
16											|						$\boxed{17}$	9	9	9	9
17											|							$\boxed{18}$	9	9	9
18											|								$\boxed{19}$	9	9
19											|									$\boxed{20}$	9
20											|										$\boxed{21}$
21											|										
22						*C*					|										
23											|										
24											|										
25											|										
26											|										
27											|									*H*	
28											|										
29										*h*	|										
30											|										
31										*h*	|										
32										*ehm*	|										
33										*bh mr*	|										

Table 8.3.3, part 2. Size Ramsey Numbers

	21	22	23	24	25	26	27	28	29	30	31	32	33
	5	5	5	5	5	5	5	5	5	5	5	5	5
1	5	5	5	5	5	5	5	7	7	7	5	7	9
2	9	9	9	9	9	9	9	9	9	9	11	11	14
3	6	6	6	6	6	6	6	6	6	6	6	6	7
4	7	7	7	7	7	7	7	7	10	10	7	10	13
5	7	7	7	7	7	8	7	8	10	10	9	10	13
6	8	7	7	7	7	7	8	9	10	10	9	11	14
7	9	9	9	9	9	9	9	9	10	10	11	11	14
8	10	9	10	10	9	10	10	11	11	11	11	13	16
9	14	13	13	13	13	13	14	14	18	18	17	19	25
10	6	7	6	7	7	7	7	7	7	7	7	7	9
11	9	9	9	9	9	9	9	9	9	9	11	11	14
12	6	9	9	9	9	9	9	9	13	13	9	13	17
13	9	9	9	9	9	9	9	9	13	13	9	13	17
14	7	9	9	9	9	9	9	9	13	13	9	13	17
15	8	9	9	9	9	9	9	9	13	13	9	13	17
16	9	9	9	9	9	9	9	9	13	13	11	13	17
17	9	9	9	9	9	9	9	9	13	13	11	13	17
18	9	9	9	9	9	9	9	9	13	13	11	13	17
19	9	9	9	9	9	9	9	10	13	13	9	13	17
20	10	10	10	10	9	10	10	13	14	14	13	17	20
21	$\vert\boxed{22}\vert$	9	9	9	9	9	10	10	13	13	11	13	17
22		$\vert\boxed{23}\vert$	10	10	9	10	10	11	13	13	11	13	17
23			$\vert\boxed{24}\vert$	10	10	10	10	11	13	13	11	13	17
24				$\vert\boxed{25}\vert$	9	9	9	10	13	13	11	13	17
25					$\vert\boxed{26}\vert$	10	10	13	13	13	13	17	19
26						$\vert\boxed{27}\vert$	10	13	14	14	13	17	20
27							$\vert\boxed{28}\vert$	14	14	14	15	17	20’22
28								$\vert\boxed{29}\vert$	18	18	17	19	25
29									$\vert\boxed{30}\vert$	18	17	19	25’28
30										$\vert\boxed{31}\vert$	15	17	27’29
31											$\vert\boxed{32}\vert$	22	30’34
32										yh		$\vert\boxed{33}\vert$	43’49
33								mr					

DEFINITION

D11: For graphs G and H, the ***restricted size Ramsey number*** $r^*(G,H)$ is the minimum size graph F of order $r(G,H)$ such that $F \longrightarrow (G,H)$.

REMARKS

R21: Table 8.3.3, which can be found in [FaSh83a], gives the size Ramsey number (or restricted size Ramsey number) for all pairs of graphs without isolated vertices and with four or less vertices. Those numbers marked with an * are restricted Ramsey numbers. The references for the numbers in Table 8.3.3 appear below the diagonal, where [CH] represents [ChHa72], [C] represents [Cl77], and [H] represents [He89a], and for the single entries [h] represents [He89b], [bh] represents [BlHa81], [ehm] represents [ExHaMe88], [yh] represents [YuHe95], and [mr] represents [McRa95].

R22: An "arrowing" graph corresponding to the size Ramsey numbers appears below the diagonal in Table 8.3.3. These graphs are not, in general, unique. The subscript is a reference to the paper with this result. In this case [1] refers to [Bu79], [2] refers to [BEFRS78], [3] refers to [EFRS78], [4] refers to [FaSh83b], [5] refers to [HaMi83], and [6] refers to [RoSh].

8.3.6 Ramsey Minimal Graphs

DEFINITIONS

D12: For a pair (G,H) of graphs, $\mathcal{R}'(\mathcal{G},\mathcal{H}) = \{\mathcal{F} \ : \ \mathcal{F} \longrightarrow (\mathcal{G},\mathcal{H})\}$. The graphs F in $\mathcal{R}'(\mathcal{G},\mathcal{H})$ are the ***Ramsey graphs*** for the pair (G,H).

D13: A graph F is (G,H)-***minimal*** if $F \in \mathcal{R}'(\mathcal{G},\mathcal{H})$, but no proper subgraph of F is in $\mathcal{R}'$.

D14: The Ramsey minimal graphs in $\mathcal{R}'(\mathcal{G},\mathcal{H})$ will be denoted by $\mathcal{R}(\mathcal{G},\mathcal{H})$.

D15: The pair (G,H) is ***Ramsey-infinite*** if the number of nonisomorphic graphs in $\mathcal{R}(\mathcal{G},\mathcal{H})$ is infinite. Otherwise, the pair (G,H) is ***Ramsey-finite***.

D16: A graph is 2.5-*connected* if it is 2-connected and any cutset with two vertices is independent.

EXAMPLES

E6: For any graph G, clearly $G \longrightarrow (G,K_2)$, and if $F \longrightarrow (G,K_2)$, then F must have G as a subgraph. Hence, $\mathcal{R}(\mathcal{G},\mathcal{K}_\epsilon) = \{\mathcal{G}\}$, and the pair (G,K_2) is Ramsey-finite.

E7: Observe that any 2-edge-coloring of an odd cycle C_n will have consecutive edges with the same color because of the parity of n. Thus, $C_n \longrightarrow (P_3,P_3)$ for n odd, and no proper subgraph of C_n will "arrow" (P_3,P_3). Thus, the pair (P_3,P_3) is Ramsey-infinite, and it is easy to show that $\mathcal{R}(P_3,P_3) = \{C_n \ : \ n \ \text{ odd}\}$.

FACTS

Nešetřil and Rödl [NeRo78] initiated the Ramsey-infinite and Ramsey-finite line of investigation. Their work was extended in [BEFRS81] and [BEFRS82a], but there is still no complete characterization of Ramsey-finite graphs for forests.

F32: (**Nešetřil and Rödl** [NeRo78]) The pair (G, G) is Ramsey-infinite if

(i) $\chi(G) \geq 3$,

(ii) G is 2.5-connected, or

(iii) G is a forest containing a P_4.

F33: [BEFRS81], [BEFRS82a] Let G and H be forests.

(i) The pair (G, H) is Ramsey-infinite if either G or H has a component that is not a star.

(ii) If G and H are star forests without isolated edges, then (G, H) is Ramsey-finite if and only if each of G and H is a single star with an odd number of edges.

(iii) There are both Ramsey-finite and Ramsey-infinite pairs of star forests (G, H) when G and H have isolated edges.

F34: [BEFS78] For m a positive integer and G an arbitrary graph, the pair (G, mK_2) is Ramsey-finite.

F35: [Lu94] If G is a forest that is not a matching and H contains a cycle, then the pair (G, H) is Ramsey-infinite.

REMARKS

R23: One consequence of Fact 35 is that a matching $G = mK_2$ is the only graph that can be paired with any graph H to yield a Ramsey-finite pair.

R24: Fact 35 also answers a question posed in [BEFRS80b] by showing that the pair (P_3, H), and in fact the pair $(K_{1,2n}, H)$ for $n \geq 1$, is Ramsey-infinite for any graph H that is not a matching.

R25: A complete characterization of the pairs of forests that are Ramsey-finite is not known, and much less is known about pairs of graphs in general.

CONJECTURE

C8: **Ramsey-finite Conjecture** The pair (G, H) is Ramsey-finite if and only if either

(i) G or H is a matching, or

(ii) G and H are appropriate star forests.

8.3.7 Generalizations and Variations

There are an endless number of generalizations to classical Ramsey numbers and only a few of them have been considered in this short survey. We end with a brief mention of some of the directions that have been explored.

Graphs

The induced Ramsey number leads to a stronger "arrowing" result, since the monochromatic graph must be induced. The existence of $r^*(G, H)$ was verified by Rödl in his doctoral thesis [Ro73], and was also verified independently by other mathematicians.

Some general upper bounds on $r^*(G,H)$ for various graphs G and H can be found in [KoPrRo98].

DEFINITIONS

D17: The ***induced Ramsey number*** $r^*(G,H)$ is the least positive integer n such that there exists a graph F of order n such that any 2-edge-coloring (Red and Blue) of F yields an induced copy of G in Red or an induced copy of H in Blue.

D18: For bipartite graphs G and H the ***bipartite Ramsey number*** $r_b(G,H)$ is the smallest order of a bipartite graph F such that $F \longrightarrow (G,H)$.

D19: The ***connected Ramsey number*** $r_c(G,H)$ is the order of the smallest graph F such that $F \longrightarrow (G,H)$ and the graph induced by each color is also connected.

D20: For a graph G, the ***Ramsey multiplicity*** $R(G,G)$ is the minimum number of monochromatic copies of G in any 2-edge-coloring of the Ramsey graph K_n where $n = r(G,G)$.

REMARKS

R26: The existence of the bipartite Ramsey number $r_b(G,H)$ was verified by Erdős and Rado [ErRa56] and has been calculated for some basic graphs as paths, stars, and some small complete bipartite graphs. This bipartite definition can be extended to any chromatic number, not just chromatic number two.

R27: Sumner [Su78] showed that $r_c(G,H) = r(G,H)$ if neither G nor H has a bridge and each has order at least four. However, $r_c(G,H) < r(G,H)$ for some graphs with bridges, such as paths [Su78] and paths with other graphs [FaSc78].

R28: Harary and Prins [HaPr74] calculated $R(G,G)$ for small order graphs and stars, but very few Ramsey multiplicities are known.

Hypergraphs

The discussion of Ramsey theory to this point has been restricted to graphs, and nearly exclusively to 2-colorings of graphs. However, the original Ramsey Theorem applied to k-uniform hypergraphs as well.

DEFINITIONS

D21: A ***hypergraph*** consists of a set of vertices V and a set of edges, each of which is a subset of V. A hypergraph is ***k-uniform*** if its edges all have cardinality k.

D22: For k-uniform hypergraphs $(G_1, G_2, \cdots, G_m)$, the ***k-hypergraph Ramsey number*** $r_k(G_1, G_2, \cdots, G_m)$ is the smallest integer n such that if the k-sets of a set of order n are colored with m colors, there will be for some i an isomorphic copy of G_i in color i.

FACT

F36: (**McKay, Radiziszowski** [McRa91]) $r_3(4,4) = 13$

REMARKS

R29: The only classical hypergraph Ramsey number known is $r_3(4,4)$.

R30: Fact 36 says that if the triples of a set with 13 elements are 2-colored, then there will be a set of order 4 with all of its subsets of order 3 having the same color, and it is not true for all colorings of the triples on a set with 12 elements.

References

[Be83] J. Beck, On size Ramsey number of paths, trees, and circuits, I, *J. Graph Theory* 7 (1983), 115–129.

[Be90] J. Beck, On size Ramsey number of paths, trees and circuits, II, *Mathematics of Ramsey theory*, (V. Rödl, J. Nešetřil eds.), *Algorithms Combin.* 5 (1990), 34–45.

[BoHa81] R. Bolze and H. Harborth, The Ramsey number $r(K_4 - x, K_5)$, pp. 109–116 in *The Theory and Applications of Graphs* (G. Chartrand, Y. Alavi, D. L. Goldsmith, L. Lesniak-Foster, and D. R. Lick, eds.), Wiley, 1981.

[BoEr73] J. A. Bondy and P. Erdös, Ramsey numbers for cycles in graphs, *J. Combin. Theory Ser. B* 14 (1973), 46–54.

[Br98] G. Brinkman, All Ramsey numbers $r(K_3, G)$ for connected graphs of order 7 and 8, *Combin. Prob. and Comp.* 7 (1998), 129–140.

[BrBrHa98] S. Brandt, G. Brinkman, and T. Harmuth, All Ramsey numbers $r(K_3, G)$ for connected graphs of order 9, *Electronic J. Combin.* 5 (1998).

[Bu74] S. A. Burr, Generalized Ramsey theory for graphs – a survey, *Graphs and Combinatorics* (R. Bari and F. Harary eds.), Springer LNM 406, Berlin (1974), 52–75.

[Bu79] S. A. Burr, A survey of noncomplete Ramsey theory for graphs, *Ann. N. Y. Acad. Sci* 328 (1979), 58–75.

[Bu81] S. A. Burr, Ramsey numbers involving graphs with long suspended paths, *J. London Math. Soc. 2 24* (1981), 405–413.

[BuEr75] S. A. Burr, P. Erdős, On the magnitude of generalized Ramsey numbers for graphs, *Infinite and Finite Sets*, I, *Colloq. Math. Soc. János Bolyai* 10 (1975), 215–240.

[BuEr76] S. A. Burr, P. Erdös, Extremal Ramsey theory for graphs, *Utilitas Math.* 9 (1976), 247–258.

[BuEr83] S. A. Burr and P. Erdös, Generalizations of a Ramsey-theoretic result of Chvátal, *J. Graph Theory* 7 (1983), 39–51.

[BEFRS78] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Ramsey minimal graphs for multiple copies, *Nederl. Akad. Wetensch. Indag. Math.* 81 (1978), 187–195.

[BEFRS80a] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, An extremal problem in generalized Ramsey theory, *Ars Combin.* 10 (1980), 193–203.

[BEFRS80b] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Ramsey minimal graphs for the pair star, connected graph, *Studia Sci. Math. Hungar.* 15 (1980), 265–273.

[BEFRS81] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Ramsey minimal graphs for star-forests, *Discrete Math.* 33 (1981), 227–237.

[BEFRS82a] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Ramsey minimal graphs for forests, *Discrete Math.* 38 (1982), 23–32.

[BEFRS82b] S. A. Burr, P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Ramsey numbers for the pair sparse graph-path or cycle, *Trans. Amer. Math. Soc.* 269 (1982), 501–512.

[BEFS78] S. A. Burr, P. Erdös, R. J. Faudree, and R. H. Schelp, A class of Ramsey-finite graphs, *Congr. Numer.* XXI (1978), 171–180.

[BuFa93] S. A. Burr and R. J. Faudree, On graphs G for which all large trees are G-good, *Graphs Combin.* 9 (1993), 305–313.

[BRb73] S. A. Burr and J. A. Roberts, On Ramsey numbers for stars, *Utilitas Math.* 4 (1973), 217–220.

[ChSc93] G. Chen and R. H. Schelp, Graphs with linearly bounded Ramsey numbers, *J. Combin. Theory Ser. B 57*, 138–149.

[ChGr83] F. R. K. Chung and C. M. Grinstead, A survey of bounds for classical Ramsey numbers, *J. Graph Theory* 7 (1983), 25–38.

[Ch77] V. Chvátal, Tree-complete graph Ramsey numbers, *J. Graph Theory* 7 (1977), 93.

[ChHa72] V. Chvátal and F. Harary, Generalized Ramsey theory for graphs, *Bull. Amer. Math. Soc.* 78 (1972), 423–426.

[CRST83] V. Chvátal, V. Ródl, E. Szemerédi, W. T. Trotter, The Ramsey number of a raph with bounded maximum degree, *J. Combin. Theory, B* 34 , 239–243.

[Cl77] M. Clancy, Some small Ramsey numbers, *J. Graph Theory* 1 (1977), 89–91.

[ErRa56] P. Erdős and R. Rado, A partition calculus in set theory, *Bull. Amer. Math. Soc.* 62 (1956), 427–489.

[EFRS78] P. Erdős, R. J. Faudree, C. C. Rousseau, R. H. Schelp, The size Ramsey number, a new concept in generalized Ramsey theory, *Periodica Mathematica Hungarica* 9 (1978), 145–161.

[EFRS78b] P. Erdős, R. J. Faudree, C. C. Rousseau, R. H. Schelp, On cycle-complete graph Ramsey numbers, *J. Graph Theory* 2 (1978), 53–64.

[EFRS85] P. Erdös, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, Multipartite graph–sparse graph Ramsey numbers, *Combinatorica* 5 (1985), 311–318.

[ErRo93] P. Erdős and C. C. Rousseau, The size Ramsey number of a complete biparite graph, *Discrete Math.* 113 (1993), 259–262.

[ErSz35] P. Erdős and G. Szekeres, A combinatorial problem in geometry, *Compositio Math.* 2 (1935), 463–470.

[Ex89a] G. Exoo, Applying optimization algorithm to Ramsey problems, pp. 175–179 in *Graph Theory, Combinatorics, Algorithms, and Applications* (Y. Alavi ed.), SIAM, 1989.

[Ex89b] G. Exoo, A lower bound for $R(5,5)$, *J. Graph Theory* 13 (1989), 97–98.

[Ex93] G. Exoo, Announcement: On the Ramsey numbers $R(4,6)$, $R(5,6)$ and $R(3,12)$, *Ars Combinatoria* 35 (1993), 85.

[Ex02] G. Exoo, Some applications of pq-graphs in graph theory, Preprint (2002).

[ExHrMe88] G. Exoo, H. Harborth, and I. Mengersen, The Ramsey number of K_4 versus $K_5 - e$, *Ars Combin.* 25A (1988), (Proc. Eleventh British Combinatorial Conf.), 277–286.

[FaRoSc92] R. J. Faudree, C. C. Rousseau and R. H. Schelp, A good idea in Ramsey theory, pp. 180–189 in *Graph Theory, Combinatorics, Algorithms, and Applications*, Soc. Indust. Appl. Math., 1992.

[FaRoSh91] R. J. Faudree, C. C. Rousseau, and J. Sheehan, Cycle - book Ramsey numbers, *Ars Combinatoria* 31 (1991), 239–248.

[FaSc74] R. J. Faudree and R. H. Schelp, All Ramsey numbers for cycles in graphs, *Discrete Math.* 8 (1974), 313–329.

[FaSc78] R. J. Faudree and R. H. Schelp, Some connected Ramsey numbers, *J. Graph Theory* 2 (1978), 119–128.

[FaScSc] R. J. Faudree, A. Schelten, and I. Schiermeyer, The Ramsey number $r(C_7, C_7, C_7)$, to appear in *Graphs and Combinatorics.*

[FaRoSc80] R. J. Faudree, C. C. Rousseau and R. H. Schelp, All triangle-graph Ramsey numbers for connected graphs of order six, *J. Graph Theory* 4 (1980), 293–300.

[FaRoSc88] R. J. Faudree, C. C. Rousseau and R. H. Schelp, Small order graph-tree Ramsey numbers, *Discrete Math.* 72 (1988), 119–127.

[FaSh83a] R. J. Faudree and J. Sheehan, Size Ramsey numbers for small-order graphs, *J. Graph Theory* 7 (1983), 53–55.

[FaSh83b] R. J. Faudree and J. Sheehan, Size Ramsey numbers involving stars, *Discrete Math.* 46 (1983), 151–157.

[GeGy67] L. Geréncser and A. Gyárfas, On Ramsey-type problems, *Ann. Univ. Sci. Budapest Eötvös Sect. Math.* 10 (1967), 167–170.

[GrRo87] R. L. Graham and V. Rödl, Numbers in Ramsey theory, *Surveys in Combinatorics* (C. Whitehead ed.), Cambridge University Press, 1987.

[GaRoSp90] R. L. Graham, B. L. Rothschild, and J. H. Spencer, *Ramsey Theory*, John Wiley & Sons, 1990.

[GrYa68] J. E. Graver and J. Yackel, Some graph theoretic results associated with Ramsey's theorem, *J. Combin. Theory* 4 (1968), 125–175.

[GrGl55] R. E. Greenwood and A. M. Gleason, Combinatorial relations and chromatic graphs, *Canad. J. Math.* 7 (1955), 1–7.

[GrRo82] C. Grinstead and S. Roberts, On the Ramsey numbers $R(3,8)$ and $R(3,9)$, *J. Combin. Theory, Ser. B* 33 (1982), 27–51.

[HaMi83] F. Harary and Z. Miller, Generalized Ramsey theory VIII: the size Ramsey number of small graphs, pp. 271–283 in *Studies in Pure Mathematics - To the Memory of Paul Turán* (P. Erdös, L. Alpár, G. Halász, and A. Sárközy, eds.), Birkhaüser, 1983.

[HaPr74] F. Harary and G. Prins, Generalized Ramsey theory for graphs, IV: The Ramsey multiplicity of a graph, *Networks* 4 (1974), 163–173.

[HxKo95] P. E. Haxell, and Y. Kohayakawa, The size-Ramsey number of trees, *Israel J. Math.* 89 (1995), 261–274.

[He87] G. R. T. Hendry, The diagonal Ramsey numbers for graphs with seven edges, *Utilitas Math.* 32 (1987), 11–34.

[He89a] G. Hendry, Ramsey numbers for graphs with five vertices, *J. Graph Theory* 13 (1989), 245–248.

[He89b] G. Hendry, The Ramsey numbers $r(K_2+\overline{K_3})$ and $r(K_1+C_4,K_4)$, *Utilitas Math.* 35 (1989), 40–54.

[HuZh98] Y. R. Huang and K. M. Zhang, An upper bound formula for two color classical Ramsey numbers, *J. Combin. Math. Combin. Comp.* 28 (1998), 347–350.

[Ka65] J. G. Kalbfleisch, Construction of special edge-chromatic graphs, *Canadian Math. Bull.* 8 (1965), 575–584.

[Ka66] J. G. Kalbfleisch, Chromatic graphs and Ramsey's theorem, *Ph.D. Thesis* University of Waterloo, (1966).

[Ke64] G. Kéry, On a theorem of Ramsey (Hungarian), *Matematikai Lapok* 15 (1964), 204–224.

[Ki95] J. H. Kim, The Ramsey number $R(3,t)$ has order of magnitude $t^2/\log t$, *Random Structures and Algorithms* 7 (1995), 173–207.

[KoPrRo98] Y. Kohayakawa, H. J. Prömel, land V. Rödl, Induced Ramsey numbers, *Combinatorica* 18 (1998), 373–404.

[Lu94] T. Luczak, On Ramsey minimal graphs, *Electron. J. Combin.* 1 (1994).

[Lu99] T. Luczak, $R(C_n, C_n, C_n) \leq (4 + o(1))n$, *J. Combin. Theory, Ser. B* 75 (1999), 174–187.

[Ma94] J. Mackey, Combinatorial remedies, *Ph.D. Thesis*, University of Hawaii, 1994.

[McRa91] B. D. McKay and S. P. Radziszowski, The first classical Ramsey number for hypergraphs is computed, pp. 304–308 in *Proc. of the Second Annual ACM-SIAM Symposium on Discrete Algorithms* SODA'91, San Francisco, 1991.

[McRa95] B. D. McKay and S. P. Radziszowski, $R(4, 5) = 25$, *J. Graph Theory* 19 (1995), 309–322.

[McRa97] B. D. McKay and S. P. Radziszowski, Subgraphs counting identities and Ramsey numbers, *J. Combin. Theory, Ser. B* 69 (1997), 193–209.

[McZh92] B. D. McKay and K. M. Zhang, The value of the Ramsey number $R(3, 8)$, *J. Graph Theory* 16 (1992), 99–105.

[Ne96] J. Nešetřil, Ramsey theory, Chapter 25 in *Handbook of Combinatorics, II* (R. L. Graham, M. Grótschel, and L. Lovász eds.), MIT Press, (1996), 1331–1403.

[NeRo78] J. Nešetřil and V. Rödl, The structure of critical graphs, *Acta. Math. Acad. Sci. Hungar.* 32 (1978), 295–300.

[Ni03p] V. Nikiforov, The cycle-complete graph Ramsey Numbers, preprint.

[Pa78] T. D. Parsons, Ramsey graph theory, pp. 361–384 in *Selected Topics in Graph Theory*, (L. W. Beineke and R. J. Wilson eds.), Academic Press, 1978.

[Ra02] S. P. Radiziszowski, Small Ramsey numbers, *Electronic J. Combin.* Dynamic Surveys (2002).

[RaKr88] S. P. Radiziszowski and D. L. Kreher, Search algorithms for Ramsey graphs by union of group orbits, *J. Graph Theory* 12 (1988), 59–72.

[Ram30] F. P. Ramsey, On a problem of formal logic, *Proc. London Math. Soc. (2)* 30 (1930), 264–286.

[Ro73] V. Rödl, The dimension of a graph and generalized Ramsey theorems, Thesis, Charles Univ., Praha, 1973.

[RoTh96] V. Rödl and R. Thomas, Arrangeability and clique subdivisions, pp. 236–239 in *The Mathematics of Paul Erdős* II (R. L. Graham and J. Nešetřil eds.), Springer-Verlag, 1996.

[Ro73a] V. Rosta, On a Ramsey type problem of J. A. Bondy and P. Erdös, I, *J. Combin. Theory, Ser. B* 15 (1973), 94–104.

[Ro73b] V. Rosta, On a Ramsey type problem of J. A. Bondy and P. Erdös, II, *J. Combin. Theory, Ser. B* 15 (1973), 105–120.

[RoSh] C. C. Rousseau and J. Sheehan, Size Ramsey numbers for certain bipartite graphs, preprint.

[RoYa92] P. Rowlinson and Y. Yang, On the third Ramsey numbers of graphs with five edges, *Combin. Math. Combin. Comp.* 11 (1992), 213–222.

[Sh83] J. B. Shearer, A note on the independence number of a triangle free-graph, *Discrete Math* 46 (1983), 83–87.

[Su78] D. P. Sumner, The connected Ramsey number, *Discrete Math.* 22 (1978), 49–55.

[Th88] A. Thomason, An upper bound for some Ramsey numbers, *J. Graph Theory* 12 (1988), 509–517.

[YuHe95] Y. Yuansheng and G. R. T. Hendry, The Ramsey number $r(K_1 + C_4, K_5 - e)$, *J. Graph Theory* 19 (1995), 13–15.

8.4 THE PROBABILISTIC METHOD

Alan Frieze, Carnegie Mellon University

8.4.1 The First Moment Method
8.4.2 Alterations
8.4.3 The Lovász Local Lemma
8.4.4 The Rödl Nibble
8.4.5 Bounds on Tails of Distributions
References

Introduction

Probabilistic tools can be used to prove non-probabilistic results in graph theory. In particular, it is often used to prove that a sufficiently large graph has a specified property.

8.4.1 The First Moment Method

DEFINITION

The *first moment method* has been used numerous times to prove results which are more difficult or perhaps impossible to prove using constructive methods.

D1: The ***first moment method*** involves defining a random variable, the knowledge of whose expected value (i.e., first moment) can resolve the question of the existence of a particular structure.

EXAMPLES

E1: To prove that there is a *tournament* T with n vertices and at least $n!2^{-(n-1)}$ Hamilton paths, choose a random tournament and compute the expected number of Hamilton paths. Since there are $n!$ paths in the undirected graph, each aligned in orientation with probability $2^{-(n-1)}$, the expected number is $n!2^{-(n-1)}$. This implies that at least one tournament has this many!

E2: [BFMRS03?] A coloring of the edges of a graph G is said to be k-bounded if no color is used more than k times. A subgraph H of G is said to be multi-colored if every edge has a different color. To prove that every k-bounded coloring of the edges of the hypercube graph Q_m contains a multi-colored copy of the hypercube graph Q_n, for $m \geq kn^2 2^n$, simply choose a random sub-cube and estimate the expected number of colors that appear twice or more. This is less than one for the given parameters. It follows that some Q_n does not have a color that appears twice.

E3: A *hypergraph* $H = (V, \{E_i : i = 1, 2, \ldots, m\})$ is *2-colorable* if there exists a partition of its vertex set V into two color classes $R \cup B$ such that $E_i \cap R \neq \emptyset$ and $E_i \cap B \neq \emptyset$ for $1 \leq i \leq m$. In general it is NP-hard to tell whether or not a hypergraph is 2-colorable. However, Erdős [Er63] showed that if $|E_i| \geq k$ for $1 \leq i \leq m$ and

$m < 2^{k-1}$ then H is 2-colorable. Simply partition V at random and show that the expected number of edges which are mono-colored is less than one.

E4: [Sp94] A tournament T is said to have property S_k if for every set S of k vertices (players), there is a vertex $v = v(S) \notin S$ such that all the edges of T which join v to S are directed toward v, i.e., player v beats everyone in S. Do finite tournaments exist with property S_k? The answer is yes. To prove this, simply choose an n which satisfies

$$\binom{n}{k}(1-2^{-k})^{n-k} < 1 \tag{2}$$

and then randomly orient the edges of K_n. Let Z be the number of sets of k vertices which for which $v(S)$ does not exist. The left-hand side of (2) is the expected value of Z, and so the first moment method proves the existence of a tournament with property S_k. For large k this gives

$$n > 2^k k^2(\ln 2)(1+o(1))$$

and this is close to being best possible, since Szekeres has proved that if $f(k)$ is the smallest number of vertices in a tournament with property S_k, then $f(k) > ck2^k$ for some constant $c > 0$ (see Moon [Mo79]).

Other results like this are given in the book by Alon and Spencer [AlSp00].

Ramsey Numbers

DEFINITIONS

D2: A ***red-blue edge-coloring*** of a graph is an edge coloring in which every edge is colored either red or blue.

D3: The ***Ramsey number*** $R(k,k)$ is the smallest integer such that for $n \geq R(k,k)$, every red-blue edge-coloring of the complete graph K_n contains either an all-red K_k or an all-blue $K-k$.

Determining the precise values of $R(k,k)$ has proven to be extremely difficult, and $R(k,k)$ is not known exactly for any $k \geq 5$. All we have are bounds. See Section 8.3.

FACT

One of the earliest bounds was proved by Erdős. It should be stated right away that Paul Erdős was a pioneer in the use of the probabilistic method and proved many beautiful results as well as inspiring numerous researchers to follow in his footsteps.

F1: Erdős [Er47] For sufficiently large values of k,

$$R(k,k) \geq (1-o(1))\frac{k}{e\sqrt{2}}2^{k/2}. \tag{1}$$

REMARKS

R1: The proof of Fact 1 is quite elementary. One wants to show that if n is smaller than the right-hand side of (1), then one can find an edge coloring without a monochromatic copy of K_k. It has proven very difficult to produce an explicit coloring that will give this result. So we proceed as follows: **we randomly color the edges of K_n**

and show that with positive probability this coloring will have the property we want, which is that no K_k will be monochromatic. This proves the *existence* of such a coloring without actually explicitly constructing one. This is the essence of the probabilistic method.

R2: In the random construction above one concentrates on the random variable Z which counts the number of monochromatic K_k in the coloring. A simple calculation shows that the expected value $\mathbf{E}(Z) < 1$ and then one can use the fact that

$$\mathbf{Pr}(Z > 1) \leq \mathbf{E}(Z) < 1$$

to show that $\mathbf{Pr}(Z = 0) > 0$.

R3: Fact 1 is one of the basic results in a deep and difficult theory. See Section 8.3 and also, for example, the books by Graham, Rothschild, and Spencer [GrRoSp90] or by Nesetril and Rödl [NeRö90].

8.4.2 Alterations

Our main example of another probablistic proof technique concerns the possible relationship between chromatic number and girth.

DEFINITION

D4: The ***alteration method*** is first to generate a random object and then to alter it, in order to obtain a property we desire.

EXAMPLES

It would be reasonable to conjecture that graphs with large girth have small chromatic number, i.e., that there is some function f such that every graph with no cycle of length less than g can be properly colored with $f(g)$ colors. In spite of its appeal, it just is not true.

E5: Erdős [Er59] proved that for any pair of integers k, ℓ there exists a graph with girth at least k and chromatic number at least ℓ. For a probabilitic proof, let $G_{n,p}$ denote the random graph with vertex set $[n] = \{1, 2, \ldots, n\}$ and in which each of the $\binom{n}{2}$ possible edges occurs with probability p, and start with a careful choice of $p = O(1/n)$ and n sufficiently large. Erdős showed that one can delete edges and vertices to create a graph G' with n' vertices, girth at least k and no independent set of size n'/ℓ. A moment's thought will convince the reader that G' has the required property.

E6: By altering the randomly colored complete graph used to prove (1) one can show that

$$R(k, k) \geq (1 - o(1))\frac{k}{e}2^{k/2}$$

E7: A similar, but more sophisticated alteration was used by Beck [Be78] to replace the inequality $m < 2^{k-1}$ in the question of 2-colorability of hypergraphs by $m = \Omega(2^k k^{1/3})$. Beck's proof was modified and improved so that the current best value for m is $m = \Omega(2^k (k/\ln k)^{1/2})$. This was done by Radhakrishnan and Srinivasan [RaSr00].

8.4.3 The Lovász Local Lemma

After the first moment method, perhaps the next most useful tool is the *local lemma* in Fact 2.

FACTS

F2: (Lovász) *Symmetric* version of the Local Lemma: Given a collection of *bad* events $A_1, A_2, \ldots, A_m$, we wish to prove that there is some point in our probability space for which none of the A_i occurs. If (i) $\mathbf{Pr}(A_i) \leq p, i = 1, 2, \ldots, m$ and (ii) the maximum degree Δ of Γ satisfies $e(\Delta + 1)p < 1$ (or even $4\Delta p < 1$), then $\mathbf{Pr}(\bigcap_{i=1}^{m} \overline{A_i}) > 0$ and so the desired point in the probability space exists.

F3: The local lemma yields a slight improvement of the lower bound on $R(k, k)$. We once again randomly color and the bad events are that a particular k-clique gets monocolored. The computations lead to a slight improvement

$$R(k, k) \geq (1 - o(1)) \frac{k\sqrt{2}}{e} 2^{k/2} \tag{3}$$

which only doubles the bound of (1).

EXAMPLES

E8: Let $H = (V, \{E_i : i = 1, 2, \ldots, m\})$ be a hypergraph in which every edge has at least k elements and suppose that each edge of H intersects at most d other edges. If $e(d+1) < 2^{d-1}$ then H is 2-colorable. We randomly 2-color V and event A_i is defined to be {edge E_i is monocolored}. Given the set of events $\mathcal{A} = \{A_i, i = 1, 2, \ldots, m\}$, we define a *dependency graph* Γ with vertex set $\mathcal{A}$ such that event A_i is independent of the events which are not adjacent to A_i in the graph Γ. In the present context of hypergraph coloring, $p = 2^{k-1}$ and $\Delta = d$ and so the local lemma proves the *existence* of a coloring, i.e., proves that the hypergraph is 2-colorable.

E9: This example concerns list coloring. Here we have a graph $G = (V, E)$ and each $v \in V$ has a list of allowable colors L_v and the question is can one choose a color $c_v \in L_v$ for each $v \in V$ so that the coloring is *proper*. The graph G is k-list colorable if $|L_v| \geq k$ for all $v \in V$ implies that a proper coloring exists. The *list chromatic number* of a graph is the minimum k such that G is k-list colorable. The following simple result that can be proven by one simple application of the local lemma is taken from Molloy and Reed [MoRe02]. Suppose that every $v \in V$ has the following properties (i) $|L_v| \geq \ell$ and (ii) each $c \in L_v$ appears at most $\frac{\ell}{8}$ times on the color list of a neighbor of v. The random experiment is to choose a random color c_v independently from L_v for each $v \in V$. The collection of events is $A_{c,e}$, which denotes {color c is chosen at both ends of edge e}. The local lemma immediately implies that there is a positive probability that none of these events will occur and so a proper coloring exists under these circumstances.

E10: Here is another simple application. Suppose that $G = (V, E)$, $|V| = n$, and r divides n. Let Δ denote maximum degree. Suppose that $r > 8\Delta$ then we can show that for any partition of V into sets $V_1, V_2, \ldots, V_m$, $m = n/r$ of size r, there is an independent set of G of size m which contains exactly one member of each V_i. Simply choose a random member of each V_i and use the local lemma to show that it is an independent set with positive probability. The events are defined by the edges of G. Event A_e will denote both endpoints of e are chosen.

REMARKS

R4: Sometimes a modification of the local lemma can be used, even when all the events are dependent. What is needed is some notion of being only weakly dependent. This has been called the *lop-sided* local lemma and has been used by Erdős and Spencer [ErSp91] and Albert, Frieze and Reed [AlFrRe95] to show the existence of multi-colored perfect matchings and Hamilton cycles.

R5: As described the local lemma is non-constructive and does not yield polynomial time algorithms for finding the objects of interest. Starting with a breakthrough by Beck [Be91], algorithmic versions have been developed by Alon [Al91], Molloy and Reed [MoRe98a], Czumaj and Scheideler [CzSc00], and Salavatipour [Sa03].

8.4.4 The Rödl Nibble

The alteration method proceeds by altering the results of a random experiment. The ***Rödl nibble*** takes this a step further. It was first used by Ródl [Rö85] to affirm a conjecture of Erdős and Hanani. This nibbling approach has become a powerful but technically demanding tool.

DEFINITION

D5: The ***Rödl nibble*** considers a random process that builds the required object of interest a little piece at a time.

FACT

F4: [Rö85] Let $M(n, k, \ell)$ denote the minimum size of a family of k-subsets of $[n]$ which contain every ℓ-subset of $[n]$ at least once. Then as $n \to \infty$, we have

$$M(n,k,l) = (1 + o(1))\frac{\binom{n}{\ell}}{\binom{k}{\ell}}.$$

This was generalized to general hypergraphs with small *co-degree* by Pippenger and Spencer [PiSp89].

EXAMPLES

E11: Johansson [Jo96] used the nibble to show that the chromatic number of a triangle free graph is $O(\frac{\Delta}{\ln \Delta})$. The main idea is to randomly color a small fraction of the vertex set, update the lists of colors available at each vertex and repeat. The proof is complicated by the need to choose colors non-uniformly. Also, one needs to use the local lemma to show that with positive probability the vertex coloring has some regularity properties.

E12: Kim [Ki95] used the nibble to show that $R(3,t) = \Omega(\frac{t^2}{\ln t})$ where $R(3,t)$ is the minimum n such that every Red-Blue coloring of the complete graph K_n contains either a Red triangle or a Blue copy of K_t. This coincides with the upper bound of $O(\frac{t^2}{\ln t})$ proved earlier by Ajtai, Komlós, and Szemerédi [AtKoSz80].

E13: Kahn [Ka96] used the nibble to prove that the *list chromatic index* of a graph G is $\Delta + o(\Delta)$. Here we properly color the edges of a graph G, using lists of colors for each edge.

E14: Molloy and Reed [MoRe98b] used the nibble to show that the *total chromatic number* of a graph is at most $\Delta + O(1)$. The total chromatic number is the minimum number of colors needed to color the edges *and* vertices of a graph so that no edge or vertex is incident/adjacent to an edge/vertex of the same color.

8.4.5 Bounds on Tails of Distributions

The probabilistic method often deals with events of low probability and has to use estimates for the probability of a large deviation of some random variable.

FACTS

The following two inequalities are widely used in probabilistic combinatorics.

F5: (Corollary to the Azuma-Hoeffding inequality — e.g., see [AlSp00]): Let $Z = Z(Y_1, Y_2, \ldots, Y_m)$ be a random variable, with $Y_1, Y_2, \ldots, Y_m$ independent. Suppose also that changing the value of one variable Y_i only changes the value of Z by at most one. Then for any $t > 0$ we have

$$\mathbf{Pr}(|Z - \mathbf{E}(Z)| \geq t) \leq 2\exp\left\{-\frac{t^2}{2m}\right\}. \qquad (4)$$

F6: Suppose that we choose a random subset S of some set X, such that each $x \in X$ is chosen independently with probability p_x. For a collection $A_1, A_2, \ldots, A_m$ of subsets of X, we want an estimate of the probability Π that S does not contain any of the A_i. Janson [Ja90] proved an upper bound on Π which is the meat of the inequality. The lower bound

$$\Pi \geq \prod_{i=1}^{m}(1 - \mathbf{Pr}(A_i \subseteq X)) \qquad (5)$$

follows directly from the FKG inequality of Fortuin, Kasteleyn and Ginibre [FoKaGi71].

EXAMPLES

E15: Inequality (4) was used by Frieze, Gould, Karonski, Pfender [FGKP02] in their paper on *graph irregularity strength*. Suppose that we weight the edges of a graph G with integers from $\{1, 2, \ldots, m\}$. The weight of a vertex is the weight of all its incident edges. A weighting is proper if every vertex has a different weight. The strength $\sigma(G)$ of a graph G is the minimum m for which a proper weighting exists. One result from [FGKP02] is that if G is r-regular and $r \leq (n/\ln n)^{1/4}$ then $s(G) \leq 1 + 10n/r$. Part of the proof involves randomly weighting each edge with a one or a two and then using (4) to bound the probability that some vertex weighting is repeated much more than its expected number of times.

E16: Inequality (5) used in [BFMRS03?] to give a simple proof of the following result. Suppose we have k-bounded *proper* coloring of the edges of K_m amd $m > 2k^{1/2}n^{3/2}$. Then there must be a multi-colored copy of K_n. We simply choose a random set, where each vertex of K_m is chosen with probability $p = 2n/m$. Then we use (5) to bound the probability that we do not choose two edges from the same color class.

REMARKS

R6: Sometimes a related inequality due to Talagrand [Ta96] can be used in place of Inequality (4).

R7: The interested reader can learn more about this subject and the related subject of random graphs from [AlSp04], [MoRe02], [Bo01], and [JaLuRu00].

References

[AjKoSz80] M. Ajtai, J. Komlós, and E. Szemerédi, A note on Ramsey numbers, *J. Combinatorial Theory, Ser. A* 29 (1980), 354–360.

[AlFrRe95] M. Albert, A. M. Frieze, and B. A. Reed, Multicoloured Hamilton cycles, *Electronic J. of Combinatorics* 2 (1995), R10.

[Al91] N. Alon, A parallel algorithmic version of the local lemma, *Random Structures and Algorithms* 2 (1991), 367–379.

[AlSp00] N. Alon and J. Spencer, *The Probabilistic Method*, Second Edition, Wiley-Interscience, 2000.

[Be78] J. Beck, On 3-chromatic hypergraphs, *Discrete Mathematics* 24 (1978), 127–137.

[Be91] J. Beck, An algorithmic approach to the Lovász local lemma, I, *Random Structures and Algorithms* 2 (1991), 343–365.

[BFMRS03?] T. Bohman, A. M. Frieze, R. Martin, M. Ruszinko, and C. Smyth, Polychromatic cliques and related questions, to appear.

[Bo01] B. Bollobás, *Random Graphs*, Second Edition, Cambridge University Press, 2001.

[CzSc00] A. Czumaj and C. Scheideler, Coloring nonuniform hypergraphs: a new algorithmic approach to the general Lovsz local lemma, *Random Structures and Algorithms* 17 (2000), 213–237.

[Er47] P. Erdős, Some remarks on the the theory of graphs, *Bulletin of the American Mathematical Society* 53 (1947), 292–294.

[Er59] P. Erdős, Graph theory and probability, *Canad. J. Math.* 11 (1959), 34–38.

[Er63] P. Erdős, On a combinatorial problem, *Nordisk Mat. Tidskr.* 11 (1963), 5–10.

[ErSp91] P. Erdős and J. Spencer, Lopsided local lemma and latin transversals, *Discrete Applied Mathematics* 30 (1991), 151–154.

[FoKaGi71] C. Fortuin, P. Kasteleyn, and J. Ginibre, Correlation inequalities in some partially ordered sets, *Communications of Mathematical Physics* 22 (1971), 89–103.

[FGKP02] A. M. Frieze, R. Gould, M. Karoński, and F. Pfender, On graph irregularity strength, *J. Graph Theory* 41 (2002), 120–137.

[GrRoSp90] R. Graham, B. Rothschild, and J. Spencer, *Ramsey Theory*, Wiley, 1990.

[Ja90] S. Janson, Poisson approximation for large deviations, *Random Structures and Algorithms* 1 (1990), 221–230.

[JaLuRu00] S. Janson, T. Łuczak, and A. Ruciński, *Random Graphs*, Wiley, 2000.

[Jo96] A. Johansson, Asymptotic choice number for triangle free graphs, DIMACS Technical report, 1996.

[Ki95] J. Kim, The Ramsey number $R(3,t)$ has order of magnitude $t^2/\log t$, *Random Structures and Algorithms* 7 (1995), 173–207.

[Ka96] J. Kahn, Asymptotically good list colorings, *J. Combinatorial Theory, Ser. A* 73 (1996), 1–59.

[MoRe98a] M. Molloy and B. A. Reed, Further algorithmic aspects of the local lemma, *Proc. 30th Annual ACM Symposium on Theory of Computing* (1998), 524–529.

[MoRe98b] M. Molloy and B. A. Reed, A bound on the total chromatic number, *Combinatorica* 18 (1998), 241–280.

[MoRe02] M. Molloy and B. A. Reed, *Graph Colouring and the Probabilistic Method*, Springer, 2002.

[Mo79] J. Moon, *Topics in Tournaments*, Holt, Reinhart, and Winston, 1979.

[NeRó90] J. Nesetřil and V. Rödl, *Mathematics of Ramsey Theory*, Springer-Verlag, 1990.

[PiSp89] N. Pippenger and J. Spencer, Asymptotic behaviour of the chromatci index for hypergraphs, *J. Combin. Theory, Ser. A* 51 (1989), 24–42.

[RaSr00] J. Radhakrishnan and A. Srinivasan, Improved bounds and algorithms for hypergraph two-colouring, *Random Structures and Algorithms* 16 (2000), 4–32.

[Ro85] V. Rödl, On a packing and covering problem, *European J. Combinatorics* 6 (1985), 69–78.

[Sa03] M. R. Salavatipour, A $(1+\epsilon)$-approximation algorithm for partitioning hypergraphs using a new algorithmic version of the Lovasz local lemma, to appear in *Proceedings of Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[Sp94] J. Spencer, *Ten Lectures on the Probabilistic Method*, Second Edition, SIAM Publications, 1994.

[Ta96] M. Talagrand, Concentration of measures and isoperimetric inequalities in product spaces, *Publications Mathematiques de l'I.H.E.S.* 81 (1996), 73–205.

GLOSSARY FOR CHAPTER 8

Andrásfai graph A_i: for $i \geq 2$, the complement of the $(i-1)^{\text{th}}$ power of the cycle graph C_{3i-1}.

c**-arrangeable graph**: a graph G whose vertices can be ordered in such a way that for any vertex v, each neighbor of v that succeeds v in the order has at most c adjacencies that precede v in the order.

arrowing graph F **for** $(G_1, G_2, \cdots, G_n)$ – a concept in Ramsey theory: we write $F \longrightarrow (G_1, G_2, \cdots, G_n)$ if for any k-edge-coloring of F there is for some i a monochromatic copy of G_i in the i^{th} color.

asymptotically normal sequence $\{X_n\}$ – of random variables: a sequence such that for all fixed a, $\mathbf{P}(\hat{X} \leq a) = o(1) + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{a} e^{-x^2/2}\, dx$ as $n \to \infty$, where $\hat{X} = (X - \mathbf{E}X)/\sqrt{\mathbf{Var}X}$.

asymptotically Poisson sequence $\{X_n\}$ – of random variables: a sequence such that $\mathbf{P}(X_n = k) = e^{-\lambda}\lambda^k/k! + o(1)$ for all fixed integers $k \geq 0$, as $n \to \infty$. [Valid in the case that λ is any bounded function of n.]

book: a collection of K_3-graphs sharing a common edge.

___, r**-book**: a collection of $(r+1)$-cliques sharing a common K_r.

booksize – of a graph G: the number of K_3-subgraphs in the largest *book* subgraph of G.

chromatic surplus: the largest number $s = s(G)$ such that in every vertex coloring of a graph G with $\chi(G)$ colors, every color class has at least s vertices.

circumference – of a graph: the length of the longest cycle.

clique$_1$ – in a graph: a maximal complete subgraph; note that there are two different definitions.

clique$_2$ – in a graph: any complete subgraph, not necessarily maximal under inclusion; note that there are two different definitions.

clique number $\omega(G)$ – of a graph G: the number of vertices of its largest clique.

closure – of a graph: the graph obtained from $G(n)$ by successively joining all nonadjacent vertices u and v with $deg(u) + deg(v) \geq n$.

complex graph: a connected graph G with $|E(G)| > |V(G)|$.

concentration inequality: a bound on the probability that a random variable differs from its mean by a large amount; a technique within the *probabilistic method.*

2.5-**connected graph**: a 2-connected graph such that any cutset with two vertices is independent.

convex graph property: a property $\mathcal{P}$ that a graph G must have, if a subgraph and a supergraph of G on the same vertex set both have $\mathcal{P}$.

k**-core** – of a graph: the largest subgraph with all vertex degrees at least k.

decreasing graph property: a property that cannot be lost by deleting edges alone.

density$_1$ – of a graph G: the ratio $|E(G)|/|V(G)|$.

___, **maximum** – of a graph G: the maximum ratio of the number of edges divided by the number of vertices, taken over all non-null subgraphs of G.

density$_2$ – of a graph G: the ratio $|E(G)| \Big/ \binom{|V(G)|}{2}$.

excess – of a graph G: the difference $|E(G)| - |V(G)|$.

extremal function $ex(n, \mathcal{F})$ – for an integer n and a family $\mathcal{F}$ of *forbidden graphs*: the function whose value is the largest number of edges of a simple n-vertex graph that contains none of the forbidden subgraphs.

extremal graph – for an integer n and a family $\mathcal{F}$ of *forbidden graphs*: a graph with no forbidden subgraphs and with largest possible number of edges.

H**-factor** – of a graph G: a collection of disjoint copies of the graph H that covers the vertices of G.

first moment method: using the expected value of a random variable to bound the probability; a technique within the *probabilistic method.*

H**-free graph** – where H is a graph: a graph that has no subgraph isomorphic to H.

girth – of a graph: the length of its shortest cycle.

hereditary property $\mathcal{P}$ – of graphs: a property that is shared with every induced subgraph of a graph having $\mathcal{P}$.

homomorphism from G **to** H: for simple graphs, a mapping $f : V(G) \to V(H)$ such that $uv \in E(G)$ implies $f(u)f(v) \in E(H)$; for general graphs, there is also an edge mapping $f : E(G) \to E(H)$ such that if u and v are the endpoints of edge $e \in E(G)$, then $f(u)$ and $f(v)$ are the endpoints of $f(e) \in E(H)$.

hypergraph: a set of vertices V and a set of *edges*, each of which is a subset of V.

___, k**-uniform**: a hypergraph whose edges are all of size k.

k**-hypergraph Ramsey number** $r_k(G_1, G_2, \cdots, G_m)$: the smallest integer n such that if the k-sets of a set of order n are colored with m colors, there will be for some i an isomorphic copy of G_i in color i.

increasing graph property: a property of a simple graph that cannot be lost by adding edges alone.

independence number $\alpha(G)$ – of a graph G: the largest possible number of vertices in an independent set.

independent set – of vertices in a graph G: a set of mutually nonadjacent vertices.

Lovász Local Lemma: a tool within the *probabilistic method* for dealing with *weakly* dependent events.

minor of a graph G: any graph formed from G by a sequence of edge deletions and contractions.

monotone graph property: a property that is either increasing or decreasing.

Mycielski graphs: any of the graphs $M_1, M_2, \ldots$ in a particular sequence constructed inductively by Mycielski.

pancyclic graph: a Hamiltonian graph having cycles of all possible lengths up to its number of vertices.

phase transition – of the random graph: loosely, the range of density during which the largest component grows from very small to very large.

k^{th} **power of a cycle**: a graph G with $V(G) = \{1, \ldots, n\}$ and $(i, j) \in E(G)$ if and only if $i - j = \pm 1, \pm 2, \ldots, \pm k \bmod n$.

probabilistic method: proving the existence of an object by showing that it exists with a positive probability.

Rödl nibble: a randomized constructive method that proceeds in rounds; a technique within the *probabilistic method.*

Ramsey graph for the pair (G, H): a graph F such that $F \longrightarrow (G, H)$.

Ramsey multiplicity $R(G, G)$: the minimum number of monochromatic copies of the graph G in any 2-edge-coloring of the Ramsey graph K_n where $n = r(G, G)$.

Ramsey number $r(G_1, G_2, \cdots, G_k)$: the least positive integer n such that for any k-edge-coloring of K_n there is for some i a monochromatic copy of the graph G_i in color i.

___, **bipartite** $r_b(G, H)$: smallest number of vertices of a bipartite graph F such that $F \longrightarrow (G, H)$.

___, **classical** – $r(G_1, G_2, \cdots, G_k)$ where each G_i is a complete graph K_{n_i}.

___, **connected**, denoted by $r_c(G, H)$: the least number of vertices in a graph F such that $F \longrightarrow (G, H)$ and such that the graph induced by each color is connected.

___, **diagonal**: a Ramsey number $r(G, H)$ with $G = H$.

___, **induced**, denoted by $r^*(G, H)$: the least positive integer n such that there exists a graph F of order n such that any 2-edge-coloring (Red and Blue) of F yields an induced copy of G in Red or an induced copy of H in Blue.

Ramsey-finite pair (G, H): a pair of graphs such that there are only finitely many nonisomorphic minimal graphs F with $F \longrightarrow (G, H)$.

Ramsey-infinite pair (G, H): a pair of graphs such that there are infinitely many nonisomorphic minimal graphs F with $F \longrightarrow (G, H)$.

(Ramsey) minimal graph – for $(G_1, G_2, \cdots, G_k)$: a graph F such that $F \longrightarrow (G_1, G_2, \cdots, G_n)$, but no proper subgraph of F also arrows $(G_1, G_2, \cdots, G_k)$.

random graph$_1$: a probability space whose domain is the set of n-vertex simple (labeled) graphs on the vertex-set $[n] = \{1, \ldots, n\}$; the probability function is determined by specifying that the edges occur independently with probability $\frac{1}{2}$ each.

___, **binomial** $\mathcal{G}(n, p)$: a probability space whose domain is the set of n-vertex simple (labeled) graphs on the vertex-set $[n] = \{1, \ldots, n\}$; the probability function is determined by specifying that the edges occur independently with probability $p = p(n)$ each; also called the *Bernoulli random graph.*

___, **uniform** $\mathcal{G}(n, m)$: the uniform probability space on those simple (labeled) graphs on the vertex-set $[n] = \{1, \ldots, n\}$ with exactly m edges; also called the *Erdős-Rényi random graph*; we observe that the probability of each edge is $p(n) = m/\binom{n}{2}$.

sharp threshold function – regarding the binomial random graph $\mathcal{G}(n, p)$, for a graph property Q: a function $f(n)$ such that for all $\epsilon > 0$, $\mathcal{G}(n, p)$ a.a.s. does not have Q if $\frac{p(n)}{f(n)} < 1 - \epsilon$, and $\mathcal{G}(n, p)$ a.a.s. has Q if $\frac{p(n)}{f(n)} > 1 + \epsilon$ (or the same with "does not have" and "has" interchanged).

sharp threshold function – regarding the uniform random graph $\mathcal{G}(n, m)$, for a graph property Q: a function $f(n)$ such that for all $\epsilon > 0$, $\mathcal{G}(n, p)$ a.a.s. does not have Q if $\frac{p(n)}{f(n)} < 1 - \epsilon$, and $\mathcal{G}(n, p)$ a.a.s. has Q if $\frac{p(n)}{f(n)} > 1 + \epsilon$ (or the same with "does not have" and "has" interchanged).

size Ramsey number: smallest number of edges of a graph F such that $F \longrightarrow (G_1, G_2, \cdots, G_n)$.

___, **restricted**: the minimum size, i.e., number of edges, of a graph F with $r(G, H)$ vertices such that $F \longrightarrow (G, H)$.

square of a cycle: the second *power* of a cycle.

star forest: a graph in which each component is a star.

star: any bipartite graph $K_{1,m}$.

strictly balanced graph: a graph whose maximum density is strictly greater than the maximum density of any of its proper subgraphs.

threshold function – for a graph property Q: regarding $\mathcal{G}(n, p)$, a function $f(n)$ such that $\mathcal{G}(n, p)$ a.a.s. does not have Q if $p = o(f)$, and $\mathcal{G}(n, p)$ a.a.s. has Q if $f = o(p)$ (or the same with "does not have" and "has" interchanged). Regarding $\mathcal{G}(n, m)$, the definition is the same but with p replaced by m.

Turán graph $T_r(n)$: the complete r-partite graph on n vertices whose classes (*partite sets*) are as nearly equal as possible.

Turán number $t_r(n)$: the number of edges of the *Turán graph* of n vertices and chromatic number r.

Turán-type problem: the problem of finding the largest possible number of edges of a graph having no graphs is a specified family of *forbidden subgraphs.*

weakly pancyclic graph: a graph having cycles of all possible lengths up to its circumference.

Chapter 9

GRAPHICAL MEASUREMENT

9.1 DISTANCE IN GRAPHS

Gary Chartrand, Western Michigan University
Ping Zhang, Western Michigan University

9.2 DOMINATION IN GRAPHS

Teresa W. Haynes, East Tennessee State University
Michael A. Henning, University of Natal, South Africa

9.3 TOLERANCE GRAPHS

F.R. McMorris, Illinois Institute of Technology

9.4 BANDWIDTH

Robert C. Brigham, University of Central Florida

GLOSSARY

9.1 DISTANCE IN GRAPHS

Gary Chartrand, Western Michigan University
Ping Zhang, Western Michigan University

9.1.1 Standard Distance in Graphs
9.1.2 Geodetic Parameters
9.1.3 Total Distance and Medians of Graphs
9.1.4 Steiner Distance in Graphs
9.1.5 Distance in Digraphs
References

Introduction

How far two objects (or sets of objects) are apart in a discrete structure is of interest, both theoretically and for its applications. Since discrete structures are naturally modeled by graphs, this leads us to studying distance in graphs. A book entirely devoted to this subject has been written (see [BuHa90]).

9.1.1 Standard Distance in Graphs

Although there is not a unique way to define the distance between two vertices in a graph, there is one definition of distance that has been used most often and is commonly accepted as the standard definition of distance.

Distance and Eccentricity

Many of the distance concepts that have been studied have their origins in maximizing distance from a vertex.

DEFINITIONS

D1: For two vertices u and v in a graph G, the ***distance*** $d(u,v)$ from u to v is the length (number of edges) of a shortest u-v path in G.

D2: A u-v path of length $d(u,v)$ is called a u-v ***geodesic***.

D3: For a vertex v in a connected graph G, the ***eccentricity*** $e(v)$ of v is the distance from v to a vertex farthest from v. That is,

$$e(v) = \max_{x \in V_G}\{d(v,x)\}$$

EXAMPLE

E1: Each vertex in the graph G of Figure 9.1.1 below is labeled with its distance from the vertex v. The distance from v to a vertex farthest from v is 3 and so $e(v) = 3$.

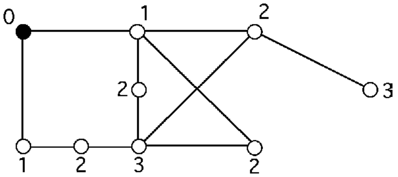


Figure 9.1.1 **Distances from the vertex v to the vertices of G.**

FACTS

F1: For vertices u and v in a connected graph G, $d(u,v) \geq 2$ if and only if u and v are distinct and non-adjacent.

F2: The distance d defined in Definition 1 above is a *metric* (and hence, $(V(G), d)$ is a *metric space*), that is, d satisfies the following four properties:

1. $d(u,v) \geq 0$ for all $u, v \in V(G)$.
2. $d(u,v) = 0$ if and only if $u = v$.
3. $d(u,v) = d(v,u)$ for all $u, v \in V(G)$ [***the symmetric property***].
4. $d(u,w) \leq d(u,v) + d(v,w)$ for all $u, v, w \in V(G)$ [***the triangle inequality***].

Radius and Diameter

Two major distance parameters associated with a connected graph G are the minimum and maximum eccentricities of the vertices of G.

DEFINITION

D4: The minimum eccentricity among the vertices of a connected graph G is the ***radius*** of G, denoted $rad(G)$, and the maximum eccentricity is its ***diameter***, $diam(G)$.

REMARK

R1: The diameter of a connected graph G also equals $\max\limits_{x,y \in V(G)} \{d(x,y)\}$.

EXAMPLE

E2: Each vertex in the graph G of Figure 9.1.2 is labeled with its eccentricity. So $rad(G) = 2$ and $diam(G) = 4$.

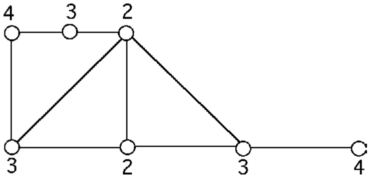


Figure 9.1.2 **Eccentricities, Radius, and Diameter.**

FACTS

F3: For every nontrivial connected graph G, $rad(G) \leq diam(G) \leq 2rad(G)$. More generally, for *any* "distance" function on $V(G)$ that is a metric, if eccentricity, radius, and diameter are defined as expected, then the inequalities $rad(G) \leq diam(G) \leq 2rad(G)$ always hold.

F4: [Os73] For every pair r, d of positive integers with $r \leq d \leq 2r$, there exists a connected graph G with $rad(G) = r$ and $diam(G) = d$. Furthermore, the minimum order (number of vertices) of such a graph is $r + d$.

Center and Periphery

The radius and diameter of a connected graph G give rise to two subgraphs of G.

DEFINITIONS

D5: A vertex v in a connected graph G is a ***central vertex*** if $e(v) = rad(G)$; while a vertex v in G is a ***peripheral vertex*** if $e(v) = diam(G)$.

D6: The subgraph induced by the central vertices of a connected graph G is the ***center*** of G, denoted $C(G)$, and the subgraph of G induced by its peripheral vertices is its ***periphery***, $P(G)$.

EXAMPLE

E3: Each vertex in the graph G of Figure 9.1.3 is labeled with its eccentricity. So $rad(G) = 3$ and $diam(G) = 5$. The center and periphery of G are also shown in Figure 9.1.3.

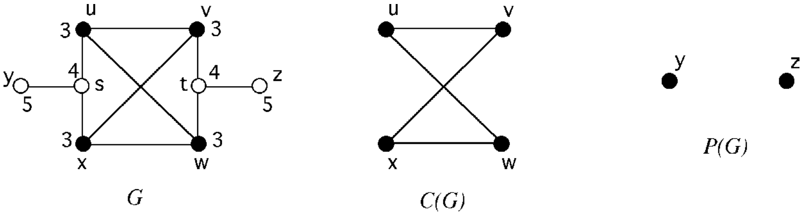


Figure 9.1.3 **The center and periphery of a graph.**

FACTS

F5: [BuMiSl81] Every graph is (isomorphic to) the center of some graph.

F6: [HaNi53] The center of every connected graph G lies in a single block of G.

F7: [Jo69] The center of every tree either consists of a single vertex or is isomorphic to K_2. Furthermore, the center of a tree T consists of a single vertex if and only if $diam(T) = 2rad(T)$.

F8: [BiSy83] A nontrivial graph G is (isomorphic to) the periphery of some graph if and only if every vertex of G has eccentricity 1 or no vertex of G has eccentricity 1.

Self-Centered Graphs

DEFINITIONS

D7: A graph G is ***self-centered*** if $C(G) = G$.

D8: An ***automorphism*** of a graph G is an isomorphism between G and itself. The set of all automorphisms of a graph G under the operation of composition forms a group called the ***automorphism group*** of G.

EXAMPLE

E4: The vertices of each of the graphs in Figure 9.1.4 are labeled with their eccentricities. The graphs $K_{2,3}$ and C_5 are self-centered, while P_4 is not.

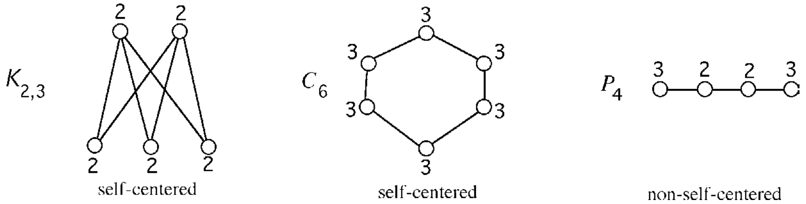


Figure 9.1.4 **Self-centered graphs and a non-self-centered graph.**

FACTS

F9: [Bu79] Let $n \geq 5$ and $r \geq 2$ be integers such that $n \geq 2r$. Then there exists a self-centered graph of order n, size m, and radius r if and only if

$$\lceil (nr - 2r - 1)/(r - 1) \rceil \leq m \leq (n^2 - 4nr + 5n + 4r^2 - 6r)/2$$

If $n = 2r = 4$, then $m = 4$.

F10: Fact 9 implies that if G is a self-centered connected graph of order n, size m, and radius 2, then $m \geq 2n - 5$.

F11: [LeWa90] For a given graph H that is not self-centered, there exists a self-centered graph G whose order exceeds the order of H by 3 and such that (1) G contains H as an induced subgraph and (2) the automorphism group of G is isomorphic to the automorphism group of H.

F12: [LeWa90] For every finite group Γ, there exists a self-centered graph whose automorphism group is isomorphic to Γ.

9.1.2 Geodetic Parameters

Geodetic Sets and Geodetic Numbers

For every connected graph G, there exists at least one set S of vertices of G such that every vertex of G lies on a geodesic connecting two vertices of S.

DEFINITIONS

D9: A vertex w is said to ***lie in*** a u-v path P if w is a vertex of P but $w \neq u, v$ (i.e., w is an internal vertex of P).

D10: For two vertices u and v in a connected graph G, the ***closed interval*** $I[u,v]$ consists of u, v, and all vertices lying in some u-v geodesic of G; while for $S \subseteq V(G)$, the ***closed interval*** $I[S]$ of S is given by $I[S] = \bigcup_{u,v \in S} I[u,v]$.

D11: A set S of vertices of a connected graph G is called a ***geodetic set*** in G if $I[S] = V(G)$. A geodetic set of minimum cardinality is a ***minimum geodetic set***. The cardinality of a minimum geodetic set is called the ***geodetic number***, $g(G)$.

D12: A graph F is a ***minimum geodetic subgraph*** if there exists a graph G containing F as an induced subgraph such that $V(F)$ is a minimum geodetic set for G.

REMARK

R2: Each vertex of a connected graph G whose neighborhood induces a complete subgraph in G belongs to every geodetic set of G. In particular, each end-vertex of G belongs to every geodetic set of G.

EXAMPLES

E5: The set $S_1 = \{x, y, z\}$ is a geodetic set of the graph G_1 in Figure 9.1.5. Since there is no 2-element geodetic set in G_1, it follows that S_1 is a minimum geodetic set of G_1 and so $g(G_1) = 3$. In G_2, the set $S_2 = \{u, v, w, t\}$ is a minimum geodetic set of G_2, so $g(G_2) = 4$.

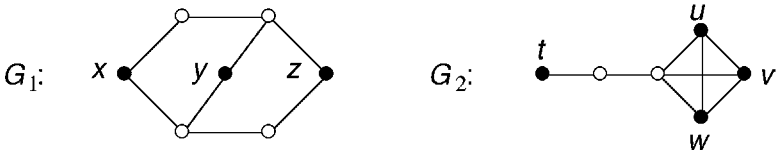


Figure 9.1.5 **Minimum geodetic sets in graphs.**

E6: The set $S = \{u, v, w, x\}$ is a minimum geodetic set of the graph G of Figure 9.1.6. Since the subgraph $\langle S \rangle$ of G induced by S is (isomorphic to) C_4, it follows that C_4 is a minimum geodetic subgraph.

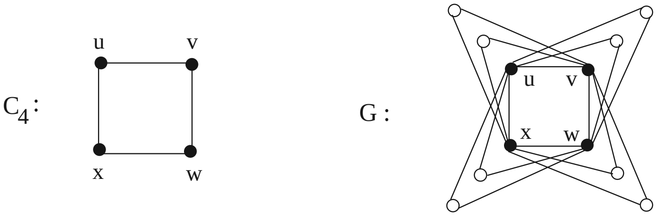


Figure 9.1.6 **A minimum geodetic subgraph.**

FACTS

F13: [HaLoTs93] Determining the geodetic number of a graph is an NP-hard problem.

F14: [ChHaZh02] If G is a connected graph of order $n \geq 2$ and diameter d, then $g(G) \leq n - d + 1$, and this bound is sharp.

F15: [BuHaQu] Let G be a nontrivial connected graph of order n. Then (a) $g(G) = n$ if and only if $G = K_n$ and (b) $g(G) = n-1$ if and only if $G = (K_{n_1} \cup K_{n_2} \cup \cdots \cup K_{n_r}) + K_1$, where $r \geq 2$ and $n_1, n_2, \cdots, n_r$ satisfy $n_1 + n_2 + \cdots + n_r = n - 1$.

F16: [ChHaZh02] For every three positive integers r, d, and $k \geq 2$ with $r \leq d \leq 2r$, there exists a connected graph G with $rad\ (G) = r$, $diam(G) = d$, and $g(G) = k$.

F17: [ChHaZh02] A nontrivial graph F is a minimum geodetic subgraph if and only if every vertex of F has eccentricity 1 or no vertex of F has eccentricity 1.

F18: [ChHaZh02] A nontrivial graph F is a minimum geodetic subgraph of a connected graph G if and only if F is the periphery of some connected graph H.

Convex Sets and Hull Sets

There are sets S of vertices in a connected graph with the property that every geodesic connecting two vertices of S contains only vertices of S.

DEFINITIONS

D13: A set S of vertices of a connected graph G is ***convex*** if $I[S] = S$ and the ***convex hull*** $[S]$ is the smallest convex set containing S.

D14: For a set S of vertices in a connected graph G, let $I^0[S] = S$, $I^1[S] = I[S]$, and $I^k[S] = I[I^{k-1}[S]]$ for $k \geq 2$. From some term on, this sequence is constant. The smallest nonnegative integer p for which $I^p[S] = I^{p+1}[S]$ is the ***geodetic iteration number***, $gin(S)$. The set $I^p[S]$ is, in fact, the convex hull of S, denoted $[S]$. The ***geodetic iteration number*** of G, denoted $gin(G)$, is given by

$$gin(G) = \max_{S \subseteq V(G)} \{gin(S)\}$$

D15: Let S be a set of vertices of a connected graph G. If $[S] = V(G)$, then S is called a ***hull set*** in G. A hull set of minimum cardinality is a **minimum hull set**. The cardinality of a minimum hull set in G is called the ***hull number*** $h(G)$.

D16: A graph F is a ***minimum hull subgraph*** if there exists a graph G containing F as an induced subgraph such that $V(F)$ is a minimum hull set for G.

EXAMPLES

E7: In the graph G of Figure 9.1.7 below, let $S_1 = \{u, v, w\}$ and $S_2 = \{u, v, w, x\}$. Since $[S_1] = S_2 \neq S_1$ and $[S_2] = S_2$, it follows that S_1 is not a convex set in G; while S_2 is a convex set in G. Furthermore, S_2 is the convex hull of S_1.

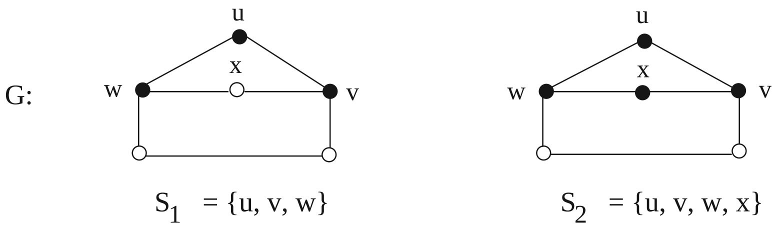


Figure 9.1.7 **Convex sets in a graph.**

E8: In the graph G of Figure 9.1.8, let $S = \{s, t, y\}$. Since $I[S] = \{s, t, u, v, w, x, y\}$ and $I^2[S] = V(G)$, it follows that S is a hull set of G. In fact, S is a minimum hull set and so $h(G) = 3$. Furthermore, $gin(S) = 2$.

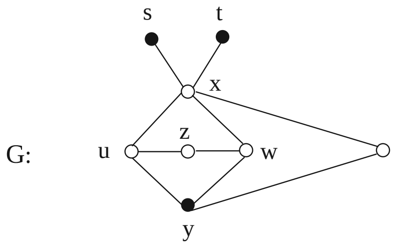


Figure 9.1.8 **A minimum hull set in a graph.**

REMARK

R3: Every geodetic set in a connected graph G is a hull set of G. The converse is not true in general. Thus $h(G) \leq g(G)$ for every connected graph G.

FACTS

F19: [HaNi81] Let n be the minimum order of a graph G for which $gin(G) = k$. Then $n = 1$ if $k = 0$, $n = 3$ if $k = 1$, and $n = k + 3$ if $k \geq 2$.

F20: [EvSe85] If G is a connected graph of order $n \geq 2$ and diameter d, then $h(G) \leq n - d + 1$.

F21: [EvSe85] If G is a k-connected graph of order n and diameter d, then $h(G) \leq n - k\lfloor d/2 \rfloor$.

F22: [ChHaZh00] For every pair a, b of integers with $2 \leq a \leq b$, there exists a connected graph G such that $h(G) = a$ and $g(G) = b$.

F23: [ChHaZh00] For every nontrivial connected graph G, $h(G) = h(G \times K_2)$.

F24: [ChHaZh00] For a connected graph G of order $n \geq 3$, $H(G) = n - 1$ if and only if $G = (K_{n_1} \cup K_{n_2} \cup \cdots \cup K_{n_r}) + K_1$, where $r \geq 2$, $n_1, n_2, \cdots, n_r$ are positive integers, and $n_1 + n_2 + \cdots + n_r = n - 1$.

F25: [ChHaZh00] A nontrivial graph F is a minimum hull subgraph of some connected graph if and only if every component of F is complete.

9.1.3 Total Distance and Medians of Graphs

Total Distance of a Vertex

DEFINITION

D17: The ***total distance*** $td(u)$ of a vertex u in a connected graph G is defined by

$$td(u) = \sum_{v \in V(G)} d(u,v)$$

ALTERNATIVE TERMINOLOGY: The total distance of a vertex has also been referred to as the ***distance*** or ***status*** of that vertex.

EXAMPLE

E9: Each vertex in the graph G of Figure 9.1.9 is labeled with its distance from the vertex u. Thus, the total distance of u is

$$td(u) = \sum_{v \in V(G)} d(u,v) = 0+1+1+2+2+2+3+4+4+5+6+7 = 37$$

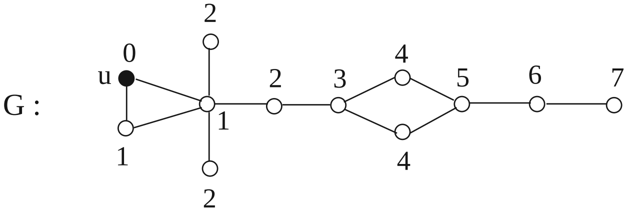


Figure 9.1.9 The total distance of a vertex u.

FACT

F26: [EnJaSn76] Let G be a connected graph of order n and size m and $v \in V(G)$. Then

$$n-1 \leq td(v) \leq (n-1)(n+2)/2 - m$$

and these bounds can be attained for each m with $n-1 \leq m \leq \binom{n}{2}$.

The Median of a Connected Graph

The center of a connected graph is not the only subgraph that's been used to describe the middle of a connected graph.

DEFINITIONS

D18: A vertex v in a connected graph G is a ***median vertex*** if v has the minimum total distance among the vertices of G.

D19: The ***median*** $M(G)$ of a connected graph G is the subgraph of G induced by its median vertices.

EXAMPLE

E10: Each vertex in the graph G of Figure 9.1.10 is labeled with its total distance. Therefore, u and v are the two median vertices of G. The median of G is also shown in Figure 9.1.10.

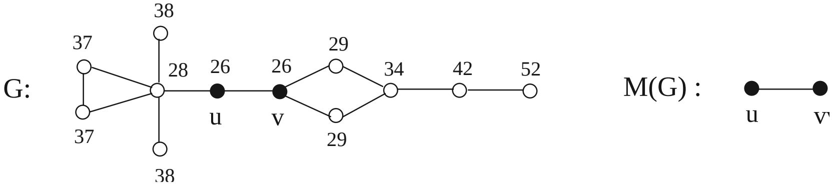


Figure 9.1.10 **A graph and its median.**

FACTS

F27: [Sl80] Every graph is (isomorphic to) the median of some graph.

F28: [Tr85] The median of every connected graph G lies in a single block of G.

Centers and Medians of a Connected Graph

There is no restriction on the relative locations of the center and median of a connected graph.

DEFINITION

D20: For two subgraphs F and H in a connected graph G, the ***distance between F and H*** is

$$d(F,H) = \min\{d(u,v) : \; u \in V(F), v \in V(H)\}$$

FACTS

F29: [Ho89] For every two graphs G_1 and G_2 and positive integer k, there exists a connected graph G such that $C(G)$ is isomorphic to G_1, $M(G)$ is isomorphic to G_2, and $d(C(G), M(G)) = k$.

F30: [NoTi91] For every three graphs G_1, G_2, and G_3, where G_3 is isomorphic to an induced subgraph of both G_1 and G_2, there exists a connected graph G such that $C(G)$ is isomorphic to G_1, $M(G)$ is isomorphic to G_2, and $C(G) \cap M(G)$ is isomorphic to G_3.

9.1.4 Steiner Distance in Graphs

There is a generalization of the distance between two vertices in a connected graph G for any set of vertices in G that is an analogue of the Euclidean Steiner Problem which seeks, for a given set S of points in the plane, the smallest network connecting the points of S.

Steiner Radius and Steiner Diameter

All of the basic distance parameters can be extended to Steiner distance.

DEFINITIONS

Observe that Steiner distance generalizes ordinary distance. That is, if $W = \{u, v\}$ for vertices u and v in a connected graph, then $sd(W) = d(u, v)$.

D21: For a nonempty set W of vertices in a connected graph G, the ***Steiner distance*** $sd(W)$ of W is the minimum size (number of edges) of a connected subgraph of G containing W. Necessarily, each such subgraph is a tree, called a ***Steiner tree with respect to*** W.

D22: Let G be a connected graph of order n. For an integer k with $1 \leq k \leq n$, the ***k-eccentricity*** of a vertex v in G, $e_k(v)$, is the maximum Steiner distance among all k-element sets of vertices of G containing v.

D23: The ***k-radius*** of a connected graph G, denoted $rad_k(G)$, and the ***k-diameter*** of G, denoted $diam_k(G)$, are given by

$$rad_k(G) = \min_{v \in V(G)} \{e_k(v)\} \quad \text{and} \quad diam_k(G) = \max_{v \in V(G)} \{e_k(v)\}$$

EXAMPLES

E11: Let $S = \{u, v, x\}$ in the graph G of Figure 9.1.11. Here $sd(S) = 4$. There are several trees of size 4 containing S, one of which is the Steiner tree T of Figure 9.1.11.

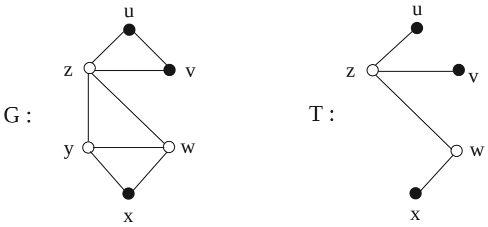


Figure 9.1.11 **A graph G and a Steiner tree T.**

E12: Each vertex in the graph G of Figure 9.1.12 is labeled with its 3-eccentricity so that $rad_3(G) = 4$ and $diam_3(G) = 6$.

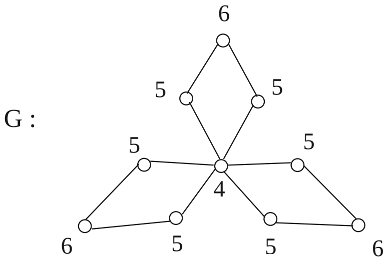


Figure 9.1.12 **The 3-eccentricities of the vertices of a graph.**

FACTS

F31: [ChOeTiZo89] Let k and n be integers with $3 \leq k \leq n$. For every tree T of order n, $diam_{k-1}(T) = rad_k(T)$.

F32: [ChOeTiZo89] For every integer $k \geq 3$ and every tree T of order at least k,

$$diam_k(T) \leq \left(\frac{k}{k-1}\right) diam_{k-1}(T)$$

Fact 32 is not true for graphs in general. [HeOeSw90]

F33: [HeOeSw90] If G is a connected graph of order n, then

(a) $diam_3(G) \leq (\frac{8}{5})\, rad_3(G)$ if $n \geq 3$.

(b) $diam_4(G) \leq (\frac{10}{7})\, rad_4(G)$ if $n \geq 4$.

F34: [HeOeSw91] For every connected graph G and every integer $k \geq 3$,

$$diam_k(G) \leq \left(\frac{k+1}{k-1}\right) diam_{k-1}(G)$$

Steiner Centers

There are a number of centers associated with Steiner distance.

DEFINITION

D24: For $k \geq 2$, a vertex v in a connected graph G is a ***k*-central vertex** if $e_k(v) = rad_k(G)$. The subgraph induced by the k-central vertices of G is the Steiner ***k*-center** of G.

FACTS

F35: [OeTi90] Let $k \geq 2$ be an integer. Every graph is (isomorphic to) the Steiner k-center of some graph.

F36: [OeTi90] Let $k \geq 3$ be an integer and T a tree. Then T is (isomorphic to) the Steiner k-center of some tree if and only if T has at most $k-1$ end-vertices.

9.1.5 Distance in Digraphs

There is a natural definition of distance from one vertex to another in digraphs as well.

DEFINITIONS

D25: Let u and v be vertices in a digraph D. If D contains one or more directed u-v paths, then the ***directed distance*** $\vec{d}(u,v)$ is the length of a shortest directed u-v path in D.

D26: A digraph D is ***strongly connected*** (or ***strong***) if D contains both a directed u-v path and a directed v-u path for every pair u, v of distinct vertices of D.

Radius and Diameter in Strong Digraphs

The definitions of eccentricity, radius, and diameter in a digraph are analogous to those in an undirected graph (see Definitions 3 and 4).

DEFINITION

D27: The ***eccentricity*** $e(v)$ of v in a strong digraph D is the greatest directed distance from v to a vertex of D. The minimum eccentricity among the vertices of D is its ***radius***, $rad(D)$, and the maximum eccentricity is its ***diameter***, $diam(D)$.

EXAMPLES

E13: There are three directed u-v paths in the digraph D of Figure 9.1.13. A shortest directed u-v path has length 2 and so $\vec{d}(u, v) = 2$. On the other hand, there is no directed v-u path in D. In fact, there is no directed x-u path in D for any vertex x ($\neq u$) of D since the indegree of u is 0. Therefore, D is not a strong digraph.

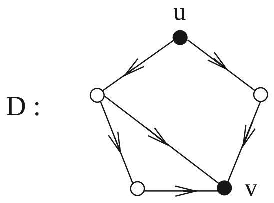


Figure 9.1.13 **A digraph that is not strong.**

E14: The vertices of the strong digraph D of Figure 9.1.14 are labeled by their eccentricities. Observe that $rad(D) = 2$ and $diam(D) = 5$. So, in general, it is not true that $diam(D) \leq 2rad(D)$.

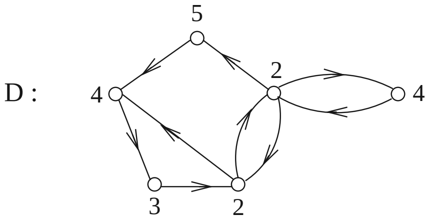


Figure 9.1.14 **The eccentricities of the vertices of a strong digraph.**

FACT

F37: [ChJoTi92] For every two positive integers a and b with $a \leq b$, there exists a strong digraph D with $rad(D) = a$ and $diam(D) = b$.

The Center of a Strong Digraph

DEFINITION

D28: The ***center*** $C(D)$ of a strong digraph D is the subdigraph induced by those vertices v with $e(v) = rad(D)$.

EXAMPLE

E15: The strong digraph D of Figure 9.1.14 is repeated in Figure 9.1.15, where its center is also shown.

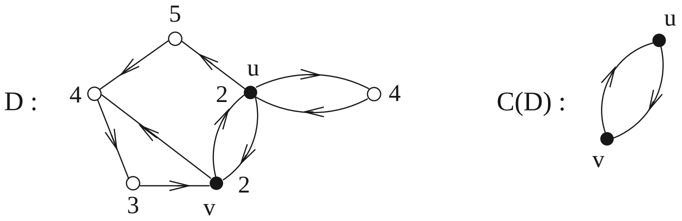


Figure 9.1.15 **The center of a strong digraph.**

FACT

F38: [ChJoTi92] For every digraph D, there exists a strong digraph whose center is (isomorphic to) D.

Strong Distance in Strong Digraphs

There is yet another reasonable way to define distance in strong digraphs, and this definition is analogous to Steiner distance in an undirected graph (see Definition 21).

DEFINITIONS

D29: For a strong digraph D, the ***strong (Steiner) distance*** $sd(u, v)$ from u to v is the minimum size of a strong subdigraph of D containing u and v.

D30: The ***strong eccentricity*** $se(v)$ of a vertex v in a strong digraph D is the largest strong distance from v to a vertex in D.

D31: The minimum strong eccentricity among the vertices of a connected graph G is the ***strong radius*** $srad(D)$ of D and the maximum strong eccentricity is its ***strong diameter*** $sdiam(D)$.

EXAMPLE

E16: In the strong digraph D_1 of Figure 9.1.16 below , $sd(v, w) = 3$, $sd(u, y) = 4$, and $sd(u, x) = 5$. The vertices of the digraph D_2 are labeled with their strong eccentricities. Therefore, $srad(D_2) = 6$ and $sdiam(D_2) = 10$.

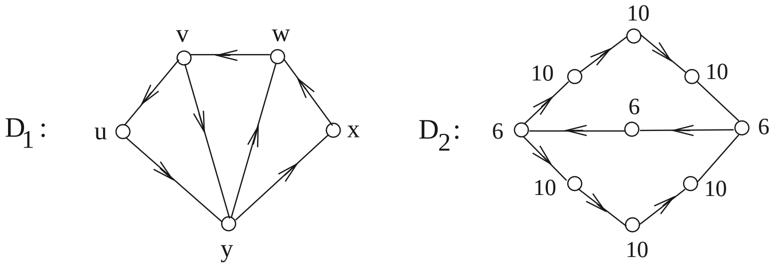


Figure 9.1.16 **Strong eccentricities, strong radius, and strong diameter.**

FACTS

F39: [ChErRaZh99-a] Strong distance is a metric on the vertex set of a strong digraph. Thus $srad(D) \leq sdiam(D) \leq 2srad(D)$ for every strong digraph D.

F40: [ChErRaZh99-a] For every pair r, d of integers with $3 \leq r \leq d \leq 2r$, there exists a strong digraph D with $srad(D) = r$ and $sdiam(D) = d$.

F41: [ChErRaZh99-a] If D is a strong digraph of order $n \geq 3$, then $sdiam\ (D) \leq \lfloor 5(n-1)/3 \rfloor$ and this bound is sharp.

Strong Centers of Strong Digraphs

DEFINITIONS

D32: A vertex v in a strong digraph D is called a ***strong central vertex*** if $se(v) = srad(G)$. The subgraph induced by the strong central vertices of D is the ***strong center*** $SC(D)$ of D.

D33: A strong digraph D is called ***strongly self-centered*** if $srad(D) = sdiam(D)$, that is, if D is its own strong center.

EXAMPLE

E17: The strong center $SC(D)$ of a digraph D is shown in Figure 9.1.17.

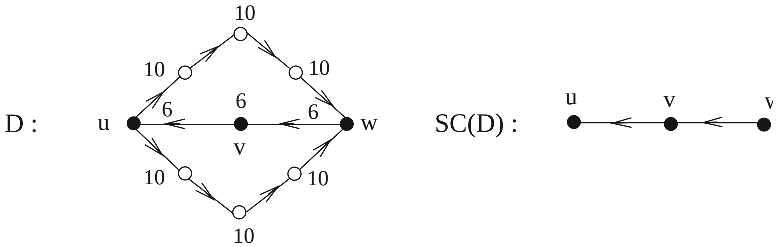


Figure 9.1.17 **The strong center of a digraph.**

FACTS

F42: [ChErRaZh99-b] Every digraph is (isomorphic to) the strong center of some strong digraph.

F43: [ChErRaZh99-b] For every integer $r \geq 3$, there exist infinitely many strongly self-centered digraphs of strong radius r.

References

[BiSy83] H. Bielak and M.M. Syslo, Peripheral vertices in graphs, *Studia Sci. Math. Hungar.* 18 (1983), 269–275.

[Bu79] F. Buckley, Self-centered graphs with a given radius, *Congr. Numer.* 23 (1979), 211–215.

[BuHa90] F. Buckley and F. Harary, *Distance in Graphs*, Addison-Wesley, 1990.

[BuHaQu] F. Buckley, F. Harary, and L. V. Quintas, Extremal results on the geodetic number of a graph, *Scientia* to appear.

[BuMiSl81] F. Buckley, Z. Miller, and P. J. Slater, On graphs containing a given graph as center, *J. Graph Theory* 5 (1981) 427–434.

[ChErRaZh99-a] G. Chartrand, D. Erwin, M. Raines, and P. Zhang, Strong distance in strong digraphs, *J. Combin. Math. Combin. Comput* 31 (1999), 33–44.

[ChErRaZh99-b] G. Chartrand, D. Erwin, M. Raines, and P. Zhang, On strong distance in strong oriented graphs, *Congr. Numer.* 141 (1999), 49–63.

[ChHaZh00] G. Chartrand, F. Harary, and P. Zhang, On the hull number of a graph, *Ars Combin.* 57 (2000), 129–138.

[ChHaZh02] G. Chartrand, F. Harary, and P. Zhang, On the geodetic number of a graph, *Networks* 39 (2002), 1–6.

[ChJoTi92] G. Chartrand, G. L. Johns, and S. Tian, Directed distance in digraphs: center on peripheries, *Congr. Numer.* 89 (1992) 89–95.

[ChOeTiZo89] G. Chartrand, O. R. Oellermann, S. Tian, and H. B. Zou, Steiner distance in graphs, *Časopis Pro Pest. Mat.* 114 (1989), 399–410.

[EnJaSn76] R. C. Entringer, D. E. Jackson, and D. E. Snyder, Distance in graphs, *Czech. Math. J.* 26 (1976), 283–296.

[EvSe85] M. G. Everett and S. B. Seidman, The hull number of a graph, *Discrete Math.* 57 (1985), 217–223.

[HaLoTs93] F. Harary, E. Loukakis, and C. Tsouros, The geodetic number of a graph, *Mathl. Comput. Modeling* 17 (1993), 89–95.

[HaNi81] F. Harary and J. Nieminen, Convexity in graphs, *J. Differential Geom.* 16 (1981), 185–190.

[HaNi53] F. Harary and R. Z. Norman, The dissimilarity characteristic of Husimi trees, *Ann. of Math.* 58 (1953), 134–141.

[HeOeSw90] M. A. Henning, O. R. Oellermann, and H. C. Swart, On Steiner radius and Steiner diameter of a graph, *Ars Combin.* 29 (1990), 13–19.

[HeOeSw91] M. A. Henning, O. R. Oellermann, and H. C. Swart, On vertices with maximum Steiner eccentricity in graphs, *Graph Theory, Combinatorics, Algorithms and Applications* (eds. Y. Alav, F. R. K. Chung, R. L. Graham, and D. F. Hsu), SIAM Publications (1991), 393–403.

[Ho89] K. S. Holbert, A note on graphs with distant center and median, *Recent Studies in Graph Theory*, Vishwa, Gulbarga (1989) 155–158.

[Jo69] C. Jordan, Sur les assemblages de lignes, *J. Reine Agnew. Math* 70 (1869), 185–190.

[LeWa90] S. M. Lee and P. C. Wang, On groups of automorphisms of self-centered graphs, *Bull. Math. Soc. Sci. Math. Roumanie* 34 (1990), 11–316.

[NoTi91] K. Novotny and S. Tian, On graphs with intersecting center and median, *Advances in Graph Theory*, Vishwa, Gulbarga (1991) 297–300.

[OeTi90] O. R. Oellermann and S. Tian, Steiner center in graphs, *J. Graph Theory* 14 (1990), 585–597.

[Os73] P. A. Ostrand, Graphs with specified radius and diameter, *Discrete Math.* 4 (1973), 71–75.

[Sl80] P. J. Slater, Medians of arbitrary graphs, *J. Graph Theory* 4 (1980), 389–392.

[Tr85] M. Truszczyński, Centers and centroids of unicyclic graphs, *Math. Slovaca* 35 (1985), 223–228.

9.2 DOMINATION IN GRAPHS

Teresa W. Haynes, East Tennessee State University
Michael A. Henning, University of Natal, South Africa

9.2.1 Introduction
9.2.2 Minimality Conditions
9.2.3 Bounds on the Domination Number
9.2.4 Nordhaus-Gaddum-Type Results
9.2.5 Domination in Planar Graphs
9.2.6 Vizing's Conjecture
9.2.7 Domination Critical Graphs
9.2.8 Domination Parameters
References

9.2.1 Introduction

We consider sets of vertices that "are near" (dominate) all the vertices of a graph. The idea of **domination** is an area of research in graph theory that is experiencing significant growth. Its application in design and analysis of communication networks, social sciences, optimization, bioinformatics, computational complexity, and algorithm design may explain in part the increased interest.

The books by Haynes, Hedetniemi and Slater [HaHeSl98, HaHeSl98b] deal exclusively with domination in graphs. Recent survey articles on domination in graphs can be found in [HAa97], [He96], and [HeLa90]. For a comprehensive bibliography of papers on dominating sets in graphs, see the current reference list compiled in [HaHeSl98] that contains over 1200 entries.

DEFINITIONS

D1: A set $S \subset V$ is a ***dominating set*** of a graph $G = (V, E)$ if each vertex in V is either in S or is adjacent to a vertex in S. A vertex is said to dominate itself and all its neighbors.

D2: The ***domination number*** $\gamma(G)$ is the minimum cardinality of a dominating set of G. We refer to a minimum dominating set of a graph G as a $\gamma(G)$***-set***.

EXAMPLE

E1: The sets $S_1 = \{a, c, e\}$ and $S_2 = \{a, d, f, h\}$ are both dominating sets in the graph G shown in Figure 9.2.1. Since no two vertices dominate G, it follows that S_1 is a dominating set of minimum cardinality (a $\gamma(G)$-set) and $\gamma(G) = 3$.

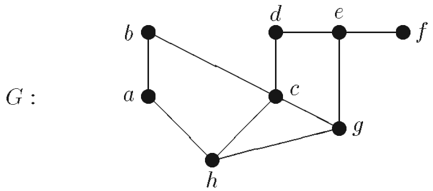


Figure 9.2.1 $S_1 = \{a, c, e\}$ **and** $S_2 = \{a, d, f, h\}$ **are two dominating sets.**

Equivalent Definitions of a Dominating Set

The domination number appears in many different mathematical contexts or frameworks. We mention a few of the equivalent definitions of a dominating set.

DEFINITIONS

D3: *Vertex Set Covering*

Set $S \subseteq V$ is a ***dominating set*** of a graph G if each vertex in $V - S$ has at least one neighbor (i.e., is *covered* by a vertex) in S.

D4: *Set Intersection*

The ***open neighborhood*** of a vertex v, denoted $N(v)$, is the set of vertices which are adjacent to v and the ***closed neighborhood*** $N[v] = N(v) \cup \{v\}$. Set $S \subseteq V$ is a ***dominating set*** if for every vertex $v \in V$, $|N[v] \cap S| \geq 1$.

D5: *Union of Neighborhoods*

Set $S \subseteq V$ is a ***dominating set*** if $\bigcup_{v \in S} N[v] = V$.

D6: *Dominating Function*

Let f be the function $f : V \to \{0, 1\}$ such that for each $v \in V$,

$$\sum_{u \in N[v]} f(u) \geq 1.$$

The vertices with the value 1 under f form a ***dominating set***.

D7: *Distance from the set*

Set $S \subseteq V$ is a ***dominating set*** if for every vertex $v \in V - S$, $d(v, x) \leq 1$ for some vertex $x \in S$.

D8: *Integer Programming*

The integer programming formulation for the ***domination number*** $\gamma(G)$ is given by:

$$\gamma(G) = \min \sum_{i=1}^{n} x_i$$

$$\text{subject to} \quad N \cdot X \geq \overline{1}_n$$

$$\text{with} \quad x_i \in \{0, 1\}$$

Applications of Domination

The applications of domination in a wide variety of fields have surely added to its escalating popularity. For a sample of its applications, consider communication networks, facility and guard location problems, surveillance systems, and coding theory.

EXAMPLES

E2: Berge [Be73] mentions the problem of keeping a number of strategic locations under surveillance by a set of radar stations. The minimum number of radar stations needed to survey all the locations is the domination number of the associated graph.

E3: Liu [Li68] discusses the application of dominance to communications in a network, where a dominating set represents a set of cities which, acting as transmission stations, can transmit messages to every city in the network.

E4: The notion of domination is a standard one in coding theory. If one defines a graph whose vertices are the n-dimensional vectors with co-ordinates chosen from $(1, \ldots, p)$ and two vertices are adjacent if they differ in one co-ordinate, then sets of vectors which are (n, p)-covering sets, single error correcting codes, or perfect covering sets are all dominating sets of the graph with certain additional properties. See, for example, Kalfleisch, Stanton, and Horton [KaStHo71].

E5: A desirable property for a committee from a collection of people might be that every nonmember know at least one member of the committee, for ease of communication. A committee with this property is a dominating set of the acquaintance graph of the set of people.

9.2.2 Minimality Conditions

Notice that if S is a dominating set of a graph G, then so too is every superset of S. However, not every subset of S is necessarily a dominating set.

DEFINITION

D9: A ***minimal dominating set*** in a graph G is a dominating set that contains no dominating set as a proper subset.

EXAMPLE

E6: For the graph G of Figure 9.2.1, $S_2 = \{a, d, f, h\}$ is a minimal dominating set that is not a minimum dominating set.

Early work on the topic of domination focused on properties of minimal dominating sets. We begin with two classical results of Ore [Or62].

FACTS

F1: (Ore's Theorem) [Or62] Let D be a dominating set of a graph $G = (V, E)$. Then D is a minimal dominating set of G if and only if each $v \in D$ has at least one of the following two properties: $[P_1]$: there exists a vertex $w \in V - D$ such that $N(w) \cap D = \{v\}$; and $[P_2]$: the vertex v is adjacent to no other vertex of D.

F2: [Or62] If $G = (V, E)$ is a graph with no isolated vertex and D is a minimal dominating set of G, then $V - D$ is a dominating set of G.

F3: [BoCo79] If G is a graph with no isolated vertex, then there exists a minimum dominating set of vertices of G in which every vertex has property P_1.

The Domination Chain

Here we discuss a domination inequality chain.

DEFINITIONS

D10: A set S of vertices is said to be ***independent*** if no two vertices in S are adjacent.

D11: The ***vertex independence number***, denoted $\beta_0(G)$, is the maximum cardinality of an independent set in G.

D12: The ***independent domination number***, denoted $i(G)$, equals the minimum cardinality of a maximal independent set of G. (A maximal independent set must be a dominating set.)

D13: While the domination number $\gamma(G)$ is the smallest cardinality of a minimal dominating set in a graph G, the ***upper domination number***, denoted $\Gamma(G)$, is the maximum cardinality of a minimal dominating set in G.

D14: For any set $S \subseteq V$, the ***open neighborhood*** $N(S)$ is defined as $\cup_{v \in S} N(v)$ and the ***closed neighborhood*** $N[S] = N(S) \cup S$. A set S of vertices is ***irredundant*** if for every vertex $v \in S$, $N[v] - N[S - \{v\}] \neq \emptyset$.

D15: The minimum cardinality of a maximal irredundant set in G is called the ***irredundance number***, and is denoted $ir(G)$.

D16: The maximum cardinality of an irredundant set in G is called the ***upper irredundance number***, and is denoted $IR(G)$.

EXAMPLES

E7: The tree T in Figure 9.2.2 has maximal independent sets of two different sizes: $\{1, 2, 3, 6, 7, 8\}$ and $\{1, 2, 3, 5\}$. Thus, $i(G) = 4$ and $\beta_0(T) = 6$.

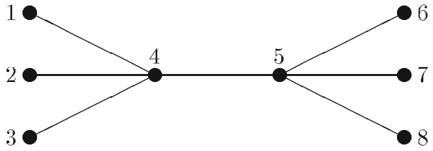


Figure 9.2.2 **A tree T with $i(T) = 4$ and $\beta_0(T) = 6$.**

E8: The tree T in Figure 9.2.3 has minimal dominating sets of three different sizes: $\{4, 5\}$, $\{4, 6, 7, 8\}$ and $\{1, 2, 3, 6, 7, 8\}$. Thus, $\gamma(T) = 2$ and $\Gamma(T) = 6$. (Notice that for this tree, $ir(T) = 2$ and $IR(T) = 6$.)

E9: The tree T in Figure 9.2.3 has maximal irredundant sets of two different sizes: $\{2, 3, 8, 9\}$ and $\{2, 4, 6, 8, 10\}$. Thus, $ir(T) = 4$ and $IR(T) = 5$. (Notice that for this tree T, $\gamma(T) = 5$.)

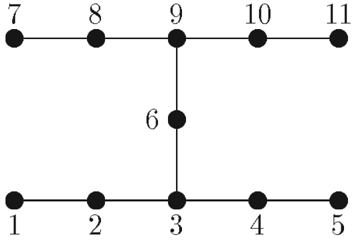


Figure 9.2.3 **A tree T with $ir(T) = 4$ and $\gamma(T) = IR(T) = 5$.**

FACTS

F4: [Be62] An independent set is maximal independent if and only if it is independent and dominating.

F5: [Be62] Every maximal independent set in a graph is a minimal dominating set of the graph.

F6: [CoHeMi78] A dominating set is a minimal dominating set if and only if it is dominating and irredundant.

F7: [BoCo79] Every minimal dominating set in a graph is a maximal irredundant set of the graph.

Since every maximal independent set is a dominating set, and every minimal dominating set is a maximal irredundant set, we have the following inequality chain, which was first observed by Cockayne, Hedetniemi, and Miller in 1978.

F8: [CoHeMi78] For every graph G,

$$ir(G) \leq \gamma(G) \leq i(G) \leq \beta_0(G) \leq \Gamma(G) \leq IR(G)$$

F9: [CoFaPaTh81] If G is a bipartite graph, then $\beta_0(G) = \Gamma(G) = IR(G)$.

REMARK

R1: This inequality chain, known as the ***domination chain***, has become one of the strongest focal points for research in domination theory; approximately 100 research papers have been published on various aspects of this sequence of inequalities. For example, Cockayne, Favaron, Mynhardt, and Puech [CoFaMyPu00] characterized trees T with $\gamma(T) = i(T)$ in terms of the sets of vertices of T which are contained in all its minimum dominating and minimum independent dominating sets. These sets were characterized by Mynhardt [My99] who used a tree pruning procedure. A simple constructive characterization of such trees is given in [DoGoHeMy03].

9.2.3 Bounds on the Domination Number

Since determining whether a graph has domination number at most k is NP-complete (see [GaJo79] and Chapter 1 of [HaHeSl98]), it is of interest to find bounds for this parameter.

REMARK

R2: Obviously if G is a graph of order n, then $1 \leq \gamma(G) \leq n$. Equality of the lower bound is attained if and only if $\Delta(G) = n - 1$, and equality holds for the upper bound if and only if $\Delta(G) = 0$, i.e., $G = \overline{K}_n$.

Bounds in Terms of Order and Minimum Degree

Restricting ourselves to graphs without isolated vertices, we have as a consequence of Facts 2 and 3 that the upper bound on the domination number can be improved from its order to one-half its order (Fact 10).

DEFINITION

D17: The ***corona*** of two graphs G_1 and G_2 is the graph $G = G_1 \circ G_2$ formed from one copy of G_1 and $|V(G_1)|$ copies of G_2 where the *ith* vertex of G_1 is adjacent to every vertex in the *ith* copy of G_2. The corona $H \circ K_1$, in particular, is the graph constructed from a copy of H and for each vertex $v \in V(H)$, a new vertex v' and the pendant edge vv' are added.

FACTS

F10: [Or62] If G is a graph of order n with no isolated vertex, then $\gamma(G) \leq n/2$.

F11: [FiJaKiRo85, PaXu82] If G is a graph of order n with no isolated vertex, then $\gamma(G) = n/2$ if and only if the components of G are the cycle C_4 or the corona $H \circ K_1$ for any connected graph H.

F12: The graphs G for which $\gamma(G) = \lfloor n/2 \rfloor$ were characterized independently in [BaCoHaSh00] and [RaVo98].

F13: If we restrict the minimum degree $\delta(G)$ of G to be at least two, then the upper bound in Fact 12 on the domination number can be improved from one-half its order to two-fifths its order except for seven exceptional graphs (one of order four and six of order seven). More precisely, McCraig and Shepherd [McSh89] defined a collection $\mathcal{B}$ of "bad" graphs shown in Figure 9.2.4.

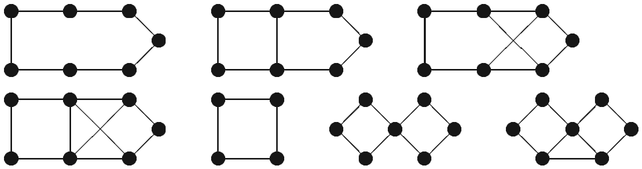


Figure 9.2.4 **A collection B of "bad" graphs.**

F14: [McSh89] If G is a connected graph of order n with $\delta(G) \geq 2$, and if $G \notin \mathcal{B}$, then $\gamma(G) \leq 2n/5$.

$\frac{2}{5}$-Minimal Graphs

McCraig and Shepherd [McSh89] introduced a family $\mathcal{F}$ of graphs whose construction is described in the sequence of definitions starting with Definition 19. They used this family to characterize the $\frac{2}{5}$-*minimal graphs* (Definition 18 below).

DEFINITIONS

D18: An n-vertex graph G is a $\frac{2}{5}$-***minimal graph*** if G is edge-minimal with respect to satisfying the following three conditions:

(i) $\delta(G) \geq 2$,

(ii) G is connected, and

(iii) $\gamma(G) \geq 2n/5$.

D19: For $m \geq 3$ and $n \geq 1$, the graph obtained by joining with an edge a vertex in cycle C_m to an end-vertex of path P_n is called a ***key*** (of order $n+m$) and is denoted by $L_{m,n}$. (See Figure 9.2.5(a).)

D20: A ***daisy*** is a graph that can be constructed from $k \geq 2$ disjoint cycles by identifying a set of k vertices, one from each cycle, into one vertex. In particular, if the k cycles have lengths $n_1, n_2, \ldots, n_k$, then we denote the daisy by $D(n_1, n_2, \ldots, n_k)$. (See Figure 9.2.5(b) and (c).)

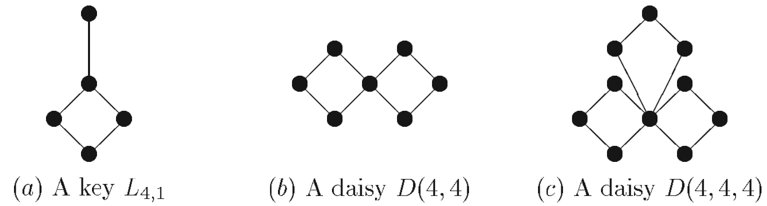

(a) A key $L_{4,1}$ (b) A daisy $D(4,4)$ (c) A daisy $D(4,4,4)$

Figure 9.2.5 **Keys and daisies.**

D21: A ***type-(a) unit*** is a graph that is isomorphic to a cycle C_5, and a ***type-(b) unit*** is a graph that is isomorphic to a key $L_{4,1}$. (See Figure 9.2.6.)

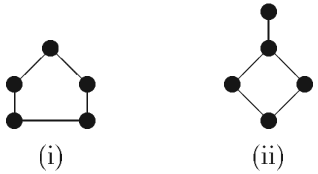

(i) (ii)

Figure 9.2.6 **A type-(a) unit and a type-(b) unit.**

D22: ***Link vertices*** in a type-(a) unit are two non-adjacent vertices selected in that unit, and the ***link vertex*** in a type-(b) unit is the vertex of degree 1 in that unit. (See Example 10 below.)

NOTATION: Let $\mathcal{F}$ denote the family of all graphs G that are obtained from the disjoint union of any combination of $\ell \geq 1$ type-(a) and type-(b) units by adding $\ell - 1$ edges so that G is connected and each new edge joins link vertices from different units. Observe that these new edges are the bridges of G.

D23: Each bridge of a graph $G \in \mathcal{F}$ is called a ***link edge*** of G.

D24: A vertex v of a graph $G \in \mathcal{F}$ is called an ***outer vertex*** of G if either v is a link vertex or v is a vertex in a type-(a) unit such that neither of its neighbors in that unit is an endpoint of a link edge. (See Example 10 below.)

NOTATION: Let $\mathcal{F}_0 = \mathcal{F} \cup \{K_{2,3}, C_{10}, D(4,7), D(4,4,4), F_1, F_2\}$, where F_1 and F_2 are the two graphs shown in Figure 9.2.7.

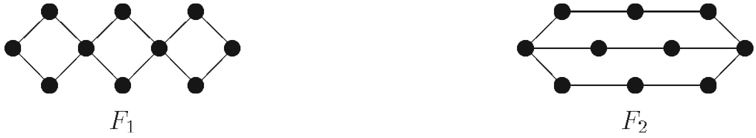


Figure 9.2.7 **Graphs F_1 and F_2.**

FACTS

F15: [McSh89] If $G \in \mathcal{F}_0$ has order n, then G is a connected graph, $\delta(G) = 2$, $\gamma(G) = 2n/5$, and for any vertex v of G, there is a $\gamma(G)$-set containing v. In particular, G is a $\frac{2}{5}$-minimal graph. Furthermore, if G_1 and G_2 are two disjoint graphs in $\mathcal{F}_0$ with $v_i \in V(G_i)$ for $i = 1, 2$ and if $(G_1 \cup G_2) + v_1v_2$ is a $\frac{2}{5}$-minimal graph, then for $i = 1, 2$, $G_i \in \mathcal{F}$ and v_i is an outer vertex of G_i.

F16: [McSh89] A graph G is a $\frac{2}{5}$-minimal graph if and only if $G \in \mathcal{F}_0 \cup \{C_4, C_7, D(4,4)\}$.

F17: [McSh89] If G is a connected graph of order $n \geq 8$ with $\delta(G) \geq 2$, then $\gamma(G) \leq 2n/5$.

F18: [Re96] If G is a connected graph of order n and $\delta(G) \geq 3$, then $\gamma(G) \leq 3n/8$. Moreover, this bound is sharp (see Example 11).

EXAMPLES

E10: A graph in the family $\mathcal{F}$ with three type-(a) units and two type-(b) units is shown in Figure 9.2.8. The link vertices are indicated by the large darkened vertices, and the outer vertices that are not link vertices are circled.

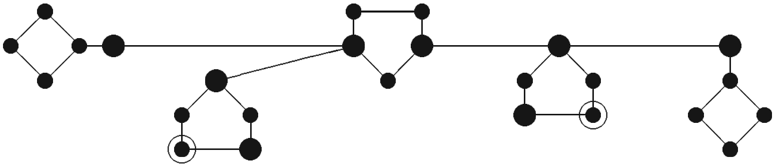


Figure 9.2.8 **A graph in the family $\mathcal{F}$.**

E11: The two non-planar cubic graphs of order $n = 8$ (shown in Figure 9.2.9) both have domination number $3 = 3n/8$.

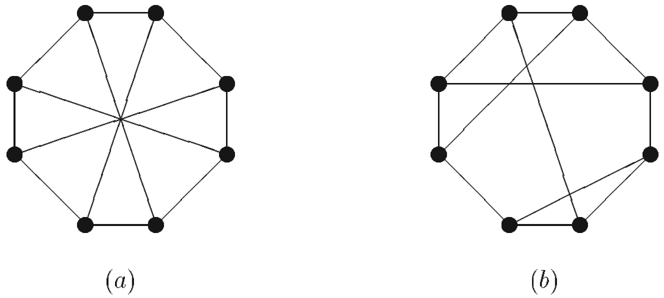


Figure 9.2.9 The two non-planar graphs of order eight.

E12: The following family of graphs shows that there are connected graphs of arbitrarily large order and minimum degree 3 that satisfy the Reed bound in Fact 18 at equality. Let H' be any connected graph. For each vertex v of H', add a (disjoint) copy of the non-planar cubic graph of order eight shown on the right in Figure 9.2.9 and identify any one of its vertices that is in a triangle with v. Let H denote the resulting graph and let $\mathcal{H}$ denote the family of all such graphs H (see Figure 9.2.10). Each graph in the family $\mathcal{H}$ has domination number three-eighths its order.

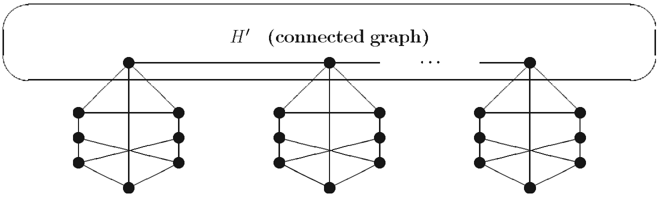


Figure 9.2.10 A graph H in the family $\mathcal{H}$.

More Bounds Involving Minimum Degree

FACTS

F19: [CaRo85, CaRo90] For any graph G of order n and minimum degree δ,

$$\gamma(G) \leq n \left[1 - \delta \left(\frac{1}{\delta + 1} \right)^{1+1/\delta} \right]$$

F20: [AlSp92, Ar74, Pa75] For any graph G of order n and minimum degree δ,

$$\gamma(G) \leq n \left(\frac{1 + \ln(\delta + 1)}{\delta + 1} \right)$$

F21: [Ar74, Pa75] For a graph G of order n and minimum degree δ,

$$\gamma(G) \leq \frac{n}{\delta + 1} \sum_{j=1}^{\delta+1} \frac{1}{j}$$

F22: [We81] Let $k = \lfloor(\log_2 n - 2\log_2\log_2 n + \log_2\log_2 e)\rfloor$. Then for almost every graph G of order n,

$$k+1 \leq \gamma(G) \leq k+2$$

Bounds in Terms of Size and Degree

TERMINOLOGY: The ***size*** of a graph G is the number of edges in G.

FACTS

F23: If G is a connected graph with m edges, then $\gamma(G) \leq (m+1)/2$.

F24: [Sa97] If G is a connected graph with m edges and $\delta(G) \geq 2$, then $\gamma(G) \leq (m+2)/3$ with equality if and only if G is a cycle of length $n \equiv 1 \pmod 3$.

DEFINITION

D25: A graph G with m edges is called an $\frac{m}{3}$***-graph*** if G satisfies the following three conditions:

(i) $\delta(G) \geq 2$,

(ii) G is connected, and

(iii) $\gamma(G) > m/3$,

EXAMPLE

E13: An example of an $\frac{m}{3}$-graph is shown in Figure 9.2.11.

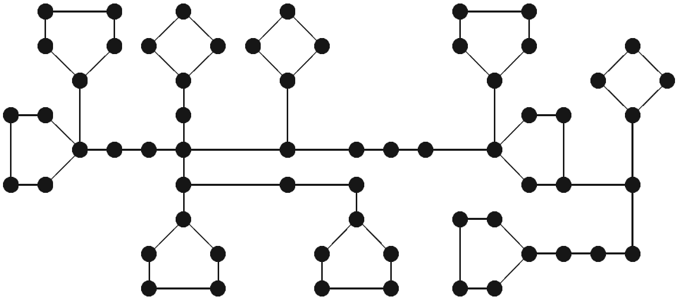


Figure 9.2.11 **A graph G of size m with $\gamma(G) = (m+1)/3$.**

REMARK

R3: A characterization of $\frac{m}{3}$-graphs can be found in [He99].

Bounds in terms of Order and Maximum Degree

FACTS

F25: [Be73, WaAcSa79] For any graph G of order n with maximum degree Δ,

$$\left\lceil \frac{n}{1+\Delta} \right\rceil \leq \gamma(G) \leq n - \Delta$$

F26: [FlVo90] For any graph G of order n with no isolated vertex, minimum degree δ and maximum degree Δ,

$$\gamma(G) \leq \frac{1}{2}\left(n+1-(\delta-1)\frac{\Delta}{\delta}\right)$$

The following fact is an immediate consequence of Fact 26.

F27: [Pa75] For a graph G of order n with no isolated vertex and minimum degree δ,

$$\gamma(G) \leq \frac{1}{2}(n+2-\delta)$$

Bounds in Terms of Order and Size

Vizing [Vi65] bounded the size of a graph having a given order and domination number. It follows that if a graph of a given order has sufficiently many edges, then it is guaranteed to have a dominating set of some specified order.

DEFINITION

D26: A ***minimum edge cover*** in a graph is a minimum number of edges required to cover all the vertices of the graph.

FACTS

F28: [Vi65] If G is a graph of order n and size m with domination number $\gamma \geq 2$, then $m \leq \frac{1}{2}(n-\gamma)(n-\gamma+2)$. Furthermore, the maximum size is attained by starting with the complete graph on $n-\gamma+2$ vertices, removing a minimum edge cover, and then adding $\gamma-2$ isolated vertices.

F29: The graphs constructed in Fact 28 also achieve the upper bound of Fact 25; that is, such a graph G has $\Delta(G) = n-\gamma(G)$.

F30: [Sa91] If G is a graph of order n and size m with domination number $\gamma \geq 2$ and with $\Delta(G) \leq n-\gamma-1$, then $m \leq \frac{1}{2}(n-\gamma)(n-\gamma+1)$.

F31: [Be62,Vi65] For any graph G of order n and size m,

$$n-m \leq \gamma(G) \leq n+1-\sqrt{1+2m}$$

Furthermore, $\gamma(G) = n-m$ if and only if every component of G is a star.

Bounds in Terms of Packing

DEFINITIONS

D27: A set $S \subset V$ is called a ***packing*** in a graph $G = (V, E)$ if the vertices in S are pairwise at distance at least 3 apart in G, i.e., if $u, v \in S$, then $d_G(u, v) \geq 3$.

D28: The ***packing number*** of a graph G is the maximum cardinality of a packing in G. It is denoted $\rho(G)$.

REMARK

R4: Note that if S is a packing in G, then for each pair of vertices $u, v \in S$, $N[u] \cap N[v] = \emptyset$. Hence, the packing number provides a lower bound on $\gamma(G)$.

FACTS

F32: For any graph G, $\rho(G) \leq \gamma(G)$.

F33: [MeMo75] For a tree T, $\rho(T) = \gamma(T)$.

9.2.4 Nordhaus-Gaddum-Type Results

In 1956 the original paper [NoGa56] by Nordhaus and Gaddum appeared. In it they gave sharp bounds on the sum and product of the chromatic numbers of a graph and its complement. Since then such results have been given for several parameters. Fact 34 below is the first such result for domination.

DEFINITION

D29: For a pair of graphs G and H, the ***cartesian product*** $G \times H$ of graphs G and H is the graph with vertex set $V(G) \times V(H)$ and where two vertices are adjacent if and only if they are equal in one coordinate and adjacent in the other.

FACTS

F34: [JaPa72] For any graph G of order $n \geq 2$,

(a) $3 \leq \gamma(G) + \gamma(\overline{G}) \leq n + 1$, and

(b) $2 \leq \gamma(G)\gamma(\overline{G}) \leq n$.

F35: [PaXu82] Let G be a graph of order $n \geq 2$. Then, $\gamma(G)\gamma(\overline{G}) = n$ if and only if G is one or the complement of one of the following graphs: K_n, disjoint union of cycles of length 4 and the corona $H \circ K_1$ for any graph H, and $K_3 \times K_3$.

F36: [JoAr95] If G is a graph of order $n \geq 2$ such that G and $\overline{G}$ have no isolated vertices, then $\gamma(G) + \gamma(\overline{G}) \leq (n + 4)/2$.

NOTATION: Let $G_1 \oplus G_2 \oplus G_3$ denote an edge-disjoint factoring of the complete graph.

F37: [GoHeSw92] Let $G_1 \oplus G_2 \oplus G_3 = K_n$. Then,
(a) $\gamma(G_1) + \gamma(G_2) + \gamma(G_3) \leq 2n + 1$, and
(b) the maximum value of the product $\gamma(G_1)\gamma(G_2)\gamma(G_3)$ is $n^3/27 + \Theta(n^2)$.

REMARK

R5: From Fact 37, there exist constants c_1 and c_2 such that the maximum triple product always lies between $n^3/27 + c_1 n^2$ and $n^3/27 + c_2 n^2$.

9.2.5 Domination in Planar Graphs

The Dominating Set decision problem remains NP-hard even when restricted to planar graphs of maximum degree 3 (see [GaJo79]). Hence it is of interest to determine upper bounds on the domination number of a planar graph.

It is easy to see that a tree of radius 2 and diameter 4 can have arbitrarily large domination number. So the interesting question is what happens when the diameter of a planar graph is 2 or 3. Bounding the diameter of a planar graph is a reasonable restriction to impose because planar graphs with small diameter are often important in applications (see [FeHeSe95]).

FACTS

F38: [MaSe96] The domination number of a planar graph of diameter 2 is bounded above by 3, and this bound is sharp.

F39: [GoHe02] Every planar graph of diameter 2 has domination number at most 2 except for the graph of Figure 9.2.11. That is, the graph of Figure 9.2.11 is the unique planar graph of diameter 2 with domination number 3.

F40: [DoGoHe03] Every planar graph of diameter 3 and of radius 2 has domination number at most 5.

F41: [DoGoHe03] Every planar graph of diameter 3 has domination number at most 9.

F42: [DoGoHe03] Every sufficiently large planar graph of diameter 3 has domination number at most 6, and this bound is sharp.

F43: [GoHe02] For each orientable surface, there are finitely many graphs with diameter 2 and domination number more than 2.

F44: [GoHe02] For each orientable surface, there is a maximum domination number of graphs with diameter 3.

EXAMPLES

E14: The graph of Figure 9.2.12, constructed by MacGillivray and Seyffarth [MaSe96], shows that the bound in Fact 38 is sharp.

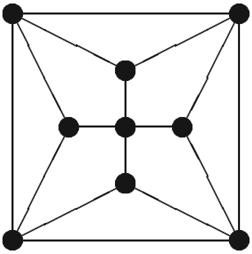


Figure 9.2.12 **A planar graph F of diameter 2 and domination number 3.**

E15: The sharpness of the bound in Fact 42 is shown by the graph of Figure 9.2.13. It can be made arbitrarily large by duplicating any of the vertices of degree 2. Furthermore, by adding edges joining vertices of degree 2, it is possible to construct such a planar graph with minimum degree equal to 3.

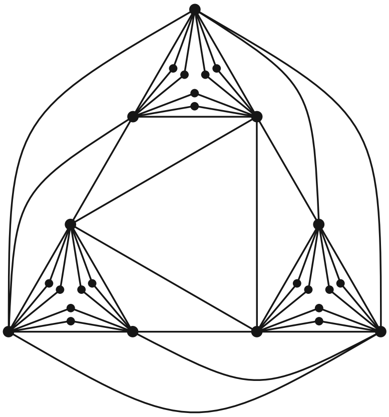


Figure 9.2.13 **A planar graph with diameter 3 and domination number 6.**

REMARKS

R6: Using Fact 38, MacGillivray and Seyffarth [MaSe96] proved that planar graphs with diameter two or three have bounded domination numbers. In particular, this implies that the domination number of such a graph can be determined in polynomial time. On the other hand, they observed that, in general, graphs with diameter 2 have unbounded domination number.

R7: While Fact 42 shows that there are finitely many planar graphs of diameter 3 with domination number more than 6, no such graph is known as of this writing.

9.2.6 Vizing's Conjecture

One of the oldest unsolved problems in domination theory involves graph products. In 1963 Vizing [Vi63] suggested the problem of determining a lower bound on the domination number of a product graph in terms of the domination numbers of its factors G and H. Five years later he offered it as conjecture, and it remains one of the most famous open problems involving domination.

CONJECTURE

Vizing's Conjecture [Vi68]: For any graphs G and H, $\gamma(G)\gamma(H) \le \gamma(G \times H)$.

DEFINITION

D30: A graph G is said to be ***decomposable*** if $\gamma(G) = k$ and $V(G)$ can be partitioned into k subsets $C_1, C_2, \ldots, C_k$ such that each of the induced subgraphs $G[C_i]$ is a complete subgraph of G.

FACTS

F45: [BaGe79] If F is a spanning subgraph of a decomposable graph G where $\gamma(F) = \gamma(G)$, then for any graph H, $\gamma(F)\gamma(H) \leq \gamma(F \times H)$.

F46: Since the family of graphs in Fact 45 includes trees and cycles, Vizing's Conjecture is true if one of G and H is a tree (independently proved by Jacobson and Kinch [JaKi86]) or a cycle (independently proved by [ElPa91]).

The best general result to date is due to Clark and Suen [ClSu00].

F47: [ClSu00] For any graphs G and H, $\gamma(G)\gamma(H) \leq 2\gamma(G \times H)$.

REMARK

R8: For surveys on graph products and Vizing's conjecture, see [HaRa91], [HaRa95], and Hartnell and Rall's Chapter 7 of [HaHeSl98b].

9.2.7 Domination Critical Graphs

In this section we consider graphs which are critical with respect to their domination number. There are several ways in which a graph could be critical. First, a graph may be critical in the sense that its domination number increases when any edge is deleted as studied in [BaHaNiSu83] and [WaAc79]. Brigham, Chinn, and Dutton [BrChDu88] studied graphs which are vertex critical in the sense that their domination number decreases when any vertex is deleted. However, the most attention has probably been directed to those graphs that are critical in the sense that their domination number drops when any missing edge is added (see [SuBl83]).

DEFINITION

D31: A graph G is ***domination critical*** if for every edge $e \notin E(G)$, $\gamma(G+e) = \gamma(G) - 1$. If G is a domination critical graph with $\gamma(G) = k$, then the graph G is said to be ***k-γ-critical***. Thus G is k-γ-critical if $\gamma(G) = k$ and $\gamma(G+e) = k-1$ for each edge $e \notin E(G)$.

FACTS

F48: The addition of an edge to G cannot increase $\gamma(G)$ and can decrease it by at most one.

F49: The 1-γ-critical graphs are (vacuously) K_n for $n \geq 1$.

F50: [SuBl83] A graph G is 2-γ-critical if and only if each component of $\overline{G}$ is a star.

F51: [Su90] A disconnected graph G is 3-γ-critical if and only if $G = A \cup B$ where either A is trivial and B is any 2-γ-critical graph or A is complete and B is a complete graph minus a 1-factor.

F52: [SuBl83] The diameter of a connected 3-γ-critical graph is at most 3.

F53: [SuBl83] Every connected 3-γ-critical graph of even order has a 1-factor.

F54: [Wo90] Every connected 3-γ-critical graph on more than six vertices has a hamiltonian path.

REMARKS

R9: To date only the 1-γ-critical and 2-γ-critical have been characterized. For $k > 2$, the structure of the k-γ-critical graphs is more complex. Most of the known results concentrate primarily on the concept of 3-γ-critical graphs.

R10: It remains an open problem to determine whether every connected 3-γ-critical graph on more than six vertices is hamiltonian.

R11: For a survey of edge domination critical graph results, see Sumner's Chapter 16 in [HaHeSl98b].

R12: Graphs for which the domination number remains unchanged when a vertex is deleted, or an edge is deleted or added have also been studied. For a survey, see Chapter 5 of [HaHeSl98].

R13: Note that six classes of graphs result from the effect these three graph modifications have on the domination number, that is, the changing or unchanging of the domination number. For example, the class for which the domination number changes when an arbitrary edge is added is the class of edge domination critical graphs. A second class is the family of graphs for which the domination number remains the same when any arbitrary edge is added. Haynes and Henning [HaHe03] investigate relationships among these six classes of graphs, illustrate the relationships in a Venn diagram, and characterize the trees in each subset of the Venn diagram.

9.2.8 Domination Parameters

Many domination parameters are formed by combining domination with another graph theoretical property P. We consider the parameters defined by imposing an additional constraint on the dominating set. Harary and Haynes [HaHa95] formalized this concept with Definition 32.

DEFINITION

D32: For a given graph property P, the ***conditional domination number*** $\gamma(G : P)$ is the smallest cardinality of a dominating set $S \subseteq V$ such that the induced subgraph $G[S]$ satisfies property P.

EXAMPLE

E16: Here are six conditional domination parameters.

$P1$. $G[S]$ is an independent set (independent domination [CoHe77]).

$P2$. $G[S]$ has no isolated vertices (total domination [CoDaHe80]).

$P3$. $G[S]$ is connected (connected domination [SaWa79]).

$P4$. $G[S]$ is a complete graph (clique domination [Ke85],[KeCo88]).

$P5$. $G[S]$ has a perfect matching (paired-domination [HaSl98]).

$P6$. $G[S]$ has a hamiltonian cycle (cycle domination [LeWi77]).

REMARKS

R14: By definition, $\gamma(G) \leq \gamma(G : P)$ for any property P. With the exception of independent domination, these conditional domination parameters do not exist for all graphs. However, graphs with no isolated vertices have total and paired-dominating sets.

R15: The generic nature of this formalization provides a method for defining many new invariants by considering different properties P.

R16: Also, new domination parameters may be defined by changing the method of dominating. For example, requiring that each vertex outside the dominating set has at least k neighbors in the dominating set is k-domination [FiJa85].

References

[AlSp92] N. Alon and J. H. Spencer, *The Probablistic Method*, John Wiley and Sons, 1992.

[Ar74] V. I. Arnautov, Estimation of the exterior stability number of a graph by means of the minimal degree of its vertices, (Russian) *Prikl. Mat. i Programmirovanie Vyp.*, 11 (126) (1974), 3–8.

[BaCoHaSh00] Xu Baogen, E. J. Cockayne, T.W. Haynes, S.T. Hedetniemi, and Z. Shangchao, Extremal graphs for inequalities involving domination parameters, *Discrete Math.* 216 (2000), 1–10.

[BaGe79] A. M. Barcalkin and L. F. German, The external stability number of the Cartesian product of graphs, *Bul. Akad. Stiince RSS Moldoven*, 94(1) (1979), 5–8.

[BaHaNi83] D. Bauer, F. Harary, J. Nieminen, and C. L. Suffel, Domination alteration sets in graphs, *Discrete Math.* 47 (1983), 153–161.

[Be62] C. Berge, *Theory of Graphs and its Applications* (Methuen, London, 1962), 40–51.

[Be73] C. Berge, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.

[BoCo79] B. Bollobás and E. J. Cockayne, Graph-theoretic parameters concerning domination, independence, and irredundance, *J. Graph Theory* 3 (1979), 241–249.

[BrChDu88] R. C. Brigham, P. Z. Chinn, and R. D. Dutton, Vertex domination-critical graphs, *Networks* 18 (1988), 173–179.

[CaRo85] Y. Caro and Y. Roditty, On the vertex-independence number and star decomposition of graphs, *Ars Combin.* 20 (1985), 167–180.

[CaRo90] Y. Caro and Y. Roditty, A note on the k-domination number of a graph, *Internat. J. Math. Sci.* 13 (1990), 205–2060.

[ClSu00] W. E. Clark and S. Suen, An inequality related to Vizing's conjecture, *Electronic J. Combin.* 7 (2000), no. 1, Note 4, 3pp. (electronic)

[CoDaHe80] E. J. Cockayne, R. M. Dawes, and S. T. Hedetniemi, Total domination in graphs, *Networks* 10 (1980), 211–219.

[CoFaMyPu00] E. J. Cockayne, O. Favaron, C. M. Mynhardt, and J. Puech, A characterization of (γ, i)-trees, *J. Graph Theory* 34(4) (2000), 277–292.

[CoFaPaTh81] E. Cockayne, O. Favaron, C. Payan, and A. Thomason, Contributions to the theory of domination, independence and irredundance in graphs, *Discrete Math.* 33 (1981), 249–258.

[CoHe77] E. J. Cockayne and S. T. Hedetniemi, Towards a theory of domination in graphs, *Networks* 7 (1977), 247–261.

[CoHeMi78] E. J. Cockayne, S. T. Hedetniemi, and D. J. Miller, Properties of hereditary hypergraphs and middle graphs, *Canad. Math. Bull.* 21 (1978), 461-468.

[ElPa91] M. El-Zahar and C. M. Pareek, Domination number of products of graphs, *Ars Combin.* 31 (1991), 223–227.

[DoGoHe03] M. Dorfling, W. Goddard, and M. A. Henning, Domination in planar graphs with small diameter II, Preprint.

[DoGoHeMy03] M. Dorfling, W. Goddard, M. A. Henning, and C. M. Mynhardt, A characterization of trees with equal domination and independent domination numbers, Preprint.

[FeHeSe95] M. Fellows, P. Hell, and K. Seyffarth, Large planar graphs with given diameter and maximum degree, *Discrete Applied Math.* 61 (1995), 133–154.

[FiJa85] J. F. Fink and M. S. Jacobson, n-domination in graphs, In Y. Alavi and A.J. Schwenk, editors, *Graph Theory with Applications to Algorithms and Computer Science* (Kalamazoo, MI 1984), 1985, 283–300. Wiley.

[FiJaKiRo85] J. F. Fink, M. S. Jacobson, L. F. Kinch, and J. Roberts, On graphs having domination number half their order, *Period. Math. Hungar.* 16 (1985), 287–293.

[FlVo90] P. Flach and L. Volkmann, Estimations for the domination number of a graph, *Discrete Math.* 80 (1990), 145–151.

[GaJo79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Francisco (1979).

[GoHeSw92] W. Goddard, M. A. Henning, and H. C. Swart, Some Nordhaus-Gaddum-type results, *J. Graph Theory* 16 (1992), 221–231.

[GoHe02] W. Goddard and M. A. Henning, Domination in planar graphs with small diameter, *J. Graph Theory* 40 (2002), 1–25.

[HaHa95] F. Harary and T. W. Haynes, Conditional graph theory IV: Dominating sets, *Utilitas Math.* 40 (1995) 179–192.

[HaRa91] B. Hartnell and D. F. Rall, On Vizing's conjecture, *Congr. Numer.* 82 (1991), 87–96.

[HaRa95] B. Hartnell and D. F. Rall, On Vizing's conjecture and the one-half argument, *Discuss. Math. Graph Theory* 15 (1995), 205–216.

[Ha97] T. W. Haynes, Domination in graphs: A brief overview, *J. Combin. Math. Combin. Comput.* 24 (1997), 225–237.

[HaHeSl98] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, 1998.

[HaHeSl98b] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Domination in Graphs: Advanced Topics*, Marcel Dekker, New York, 1998.

[HaSl98] T. W. Haynes and P. J. Slater, Paired-domination in graphs, *Networks* 32 (1998) 199–206.

[HaHe03] T. W. Haynes and M. A. Henning, Changing and unchanging domination: a classification, Preprint.

[HeLa90] S. T. Hedetniemi and R. C. Laskar, editors, Topics on Domination, volume 48. North Holland, New York, 1990.

[He96] M. A. Henning, Domination in graphs: A survey. *In Surveys in Graph Theory* (eds. G. Chartrand and M. Jacobson), *Congress. Numer.* 116 (1996), 139–172.

[He99] M. A. Henning, A characterisation of graphs with minimum degree 2 and domination number exceeding a third their size, *J. Combin. Math. Combin. Comput.* 31 (1999), 45–64.

[JaKi86] M. S. Jacobson and L. F. Kinch, On the domination of the products of graphs II: trees, *J. Graph Theory* 10 (1986), 97–106.

[JaPa72] F. Jaeger and C. Payan, Relations du type Nordhaus-Gaddum pour le nombre d'absorption d'un graphe simple, *C.R. Acad. Sci. Ser. A* 274 (1972), 728–730.

[JaAr95] J. P. Joseph and S. Arumugam, Domination in graphs, *Internat. J. Management Systems* 11 (1995), 177–182.

[KaStHo71] J. G. Kalfleisch, R. G. Stanton, and J. D. Horton, On covering sets and error-correcting codes, *J. Comb. Theory* 11A (1971), 233–250.

[Ke85] L. L. Kelleher, *Domination in Graphs and its Application to Social Network Theory*, PhD thesis, Northeastern Univ., 1985.

[KeCo88] L. L. Kelleher and M. B. Cozzens, Dominating sets in social networks, *Math. Social. Sci.* 16 (1988) 267–279.

[LeWi77] L. Lesniak-Foster and J. E. Williamson. On spanning and dominating circuits in graphs, *Canad. Math. Bull.* 20 (1977) 215–220.

[Li68] C. L. Liu, *Introduction to Combinatorial Mathematics*, McGraw-Hill, New York, 1968.

[MaSe96] G. MacGillivray and K. Seyffarth, Domination numbers of planar graphs, *J. Graph Theory* 22 (1996), 213–229.

[McSh89] W. McCuaig and B. Shepherd, Domination in graphs with minimum degree two, *J. Graph Theory* 13 (1989), 749–762.

[MeMo75] A. Meir and J. W. Moon, Relations between packing and covering numbers of a tree, *Pacific J. Math.* 61 (1975), 225–233.

[My99] C. M. Mynhardt, Vertices contained in every minimum domination set of a tree, *J. Graph Theory* 31(3) (1999), 163–177.

[NoGa56] E. A. Nordhaus and J. W. Gaddum, On complementary graphs, *Amer. Math. Monthly* 63 (1956), 175–177.

[Pa75] C. Payan, Sur le nombre d'absorption d'un graphe simple, *Cahiers Centre Études Recherche Opér.* 17 (1975), 307–317.

[PaXu82] C. Payan and N. H. Xuong, Domination-balanced graphs, *J. Graph Theory* 6 (1982), 23–32.

[Or62] O. Ore, *Theory of graphs*, *Amer. Math. Soc. Transl.* 38 (Amer. Math. Soc., Providence, RI, 1962), 206–212.

[RaVo98] B. Randerath and L. Volkmann, Characterization of graphs with equal domination and covering number, *Discrete Math.* 191 (1998) 159–169.

[Re96] B. A. Reed, Paths, stars and the number three, *Combin. Probab. Comput.* 5 (1996), 277–295.

[SaWa79] E. Sampathkumar and H. B. Walikar, The connected domination number of a graph, *J. Math. Phys. Sci.* 13 (1979), 607–613.

[Sa91] L. A. Sanchis, Maximum number of edges in connected graphs with a given domination number, *Discrete Math.* 87 (1991), 64–72.

[Sa97] L. A. Sanchis, Bounds related to domination in graphs with minimum degree two, *J. Graph Theory* 25 (1997), 139–152.

[Su90] D. P. Sumner, Critical concepts in domination, *Discrete Math.* 86 (1990), 33–46.

[SuBl83] D. P. Sumner and P. Blitch, Domination critical graphs, *J. Combin. Theory Ser. B* 34 (1983), 65–76.

[Vi63] V. G. Vizing, The cartesian product of graphs, *Vychisl/ Sistemy* 9 (1963), 30–43.

[Vi65] V. G. Vizing, A bound on the external stability number of a graph, *Doklady A. N.* 164 (1965), 729–731.

[Vi68] V. G. Vizing, Some unsolved problems in graph theory, *Uspekhi Mat. Nauk* 23(6) (1968), 117–134.

[WaAc79] H. B. Walikar and B. D. Acharya, Domination critical graphs, *Natl. Acad. Sci. Lett.* 2 (1979), 70-72.

[WaAcSa79] H. B. Walikar, B. D. Acharya, and E. Sampathkumar, Recent developments in the theory of domination in graphs, Mehta Research Institute, Allahabad, MRI *Lecture Notes in Math*, 1 (1979).

[We81] K. Weber, Domination number for almost every graph, *Rostock. Math. Kolloq.*, 16 (1981), 31–43.

[Wo90] E. Wojcicka, Hamiltonian properties of domination-critical graphs, *J. Graph Theory* 14 (1990), 205–215.

9.3 TOLERANCE GRAPHS

F.R. McMorris, Illinois Institute of Technology

9.3.1 Intersection Graphs
9.3.2 Tolerance
References

Introduction

Suppose each vertex of a graph is assigned a set. Two vertices are then declared to be adjacent when the size of their corresponding set intersection is sufficiently large, i.e., exceeds some predetermined tolerance threshold. In this section, several versions of tolerance are presented matched together with different types of sets such as intervals and subtrees.

All graphs in this section are simple.

NOTATION: Arcs in digraphs can be represented unambiguously by ordered pairs of vertices. Similarly, an edge in an undirected graph is represented by juxtaposing its endpoints.

9.3.1 Intersection Graphs

An intersection graph is in many ways a trivial tolerance graph, in that two vertices are adjacent if their corresponding sets simply have a nonempty intersection.

Ordinary Intersection Graphs

DEFINITIONS

D1: Let $\mathcal{F} = \{S_1, \ldots, S_n\}$ be a family of sets. The ***intersection graph*** of $\mathcal{F}$, denoted $\Omega(\mathcal{F})$, is the graph having $\mathcal{F}$ as vertex set with S_i adjacent to S_j if and only if $i \neq j$ and $S_i \cap S_j \neq \emptyset$. A graph G is an ***intersection graph*** if there exists a family $\mathcal{F}$ such that $G \cong \Omega(\mathcal{F})$, where typically this isomorphism is displayed by writing $V(G) = \{v_1, \ldots, v_n\}$ with each v_i corresponding to S_i. Thus, $v_iv_j \in E(G)$ if and only if $S_i \cap S_j \neq \emptyset$. When $G \cong \Omega(\mathcal{F})$, $\mathcal{F}$ is called a ***set representation*** of G.

D2: The ***intersection number*** of a graph G, denoted $i(G)$, is the minimum cardinality of a set S such that G is an intersection graph of a family of subsets of S.

D3: An ***edge clique cover*** of a graph G is any family $\mathcal{E} = \{C_1, \ldots, C_k\}$ of complete subgraphs of G such that every edge of G is in at least one of $E(C_1), \ldots,$ $E(C_k)$. For technical reasons we allow a complete subgraph to be the null subgraph of G. The smallest k for which there exists an edge clique cover $\mathcal{E} = \{C_1, \ldots, C_k\}$ of G is denoted by $\theta(G)$.

TERMINOLOGY NOTE: Elsewhere, a clique is required to be a maximal complete subgraph.

EXAMPLE

E1: Suppose $\mathcal{F} = \{S_1, S_2, S_3, S_4, S_5\}$ where $S_1 = \{x_1\}$, $S_2 = \{x_1, x_2, x_3\}$, $S_3 = \{x_4\}$, $S_4 = \{x_1, x_3, x_4, x_5\}$, and $S_5 = \{x_1, x_5\}$. Then $G \cong \Omega(\mathcal{F})$ is shown in Figure 9.3.1.

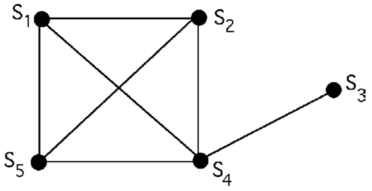


Figure 9.3.1 **An intersection graph G.**

FACTS

F1: (Marczewski's Theorem) [Ma45] Every graph is an intersection graph.

F2: [ErGoPo66] For every graph G, $i(G) = \theta(G)$.

F3: [KoStWo78] Determining $\theta(G)$ is NP-hard.

F4: [ErGoPo66] For any graph G with $p = |V(G)|$, $i(G) \leq \lfloor p^2/4 \rfloor$.

REMARK

R1: Many other results with proofs can be found in [McMc99].

Interval Graphs

Interval graphs form an interesting and important class of graphs first studied mathematically in [Ha57] and used in a biology application in [Be59]. Many facts about interval graphs can be found in [Fi85], [Go80], [McMc99], and [MiRo84]. Only a few are stated here for background material for the tolerance subsection.

DEFINITIONS

D4: An ***interval graph*** is a graph isomorphic to an intersection graph of a family of intervals of the real line.

D5: An interval graph that has a representation having intervals all of the same length is a ***unit interval graph***.

D6: An interval graph that has a representation with no interval in the representation properly containing another is a ***proper interval graph***.

D7: An ***asteroidal triple*** of a graph G is a set of three vertices of G such that, for any two of these vertices, there is a path containing those two but no neighbor of the third.

D8: A ***transitive orientation*** of graph is an assignment of directions to the edges such that the resulting arc-set $A(G)$ satisfies $(x,y),(y,z) \in A(G) \to (x,z) \in A(G)$ for all vertices $x, y, z \in V(G)$.

EXAMPLE

E2: For the graph shown in Figure 9.3.2, the vertices x, y, z form an asteroidal triple.

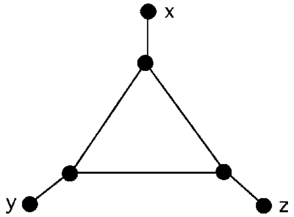


Figure 9.3.2 (x, y, z) **is an asteroidal triple.**

FACTS

F5: [LeBo62] A graph is an interval graph if and only if it contains no asteroidal triples and contains no C_4 as an induced subgraph.

F6: [GiHo64] A graph is an interval graph if and only if it contains no C_4 as an induced subgraph and its complement has a transitive orientation.

F7: [Ro69] Let G be a graph. The following statements are equivalent:
(a) G is a proper interval graph.
(b) G is a unit interval graph.
(c) G is an interval graph with no $K_{1,3}$ as an induced subgraph.

Chordal Graphs

DEFINITIONS

D9: A graph is a ***chordal graph*** if it does not contain any C_n (n-cycle) for $n > 3$, as an induced subgraph.

D10: A graph is a ***subtree graph*** if it is isomorphic to the intersection graph of a family of subtrees of a tree.

D11: A vertex in a graph is ***simplicial*** if its neighbors induce a complete subgraph.

D12: An ordering $\langle v_1, \ldots, v_n \rangle$ of all the vertices of G is a ***perfect elimination ordering*** of G if, for each $i \in \{1, \ldots, n\}$, v_i is a simplicial vertex of the subgraph induced on the vertex subset $\{v_i, v_{i+1}, \ldots, v_n\}$.

REMARK

R2: Two of the most important results on chordal graphs are the following. Others can be found in [BrLeSp99] and [McMc99], for example.

FACTS

F8: [Bu74, Ga74, Wa78] A graph is chordal if and only if it is a subtree graph.

F9: [FuGr65, Ro70] A graph is chordal if and only if it has a perfect elimination ordering.

Competition Graphs

DEFINITION

D13: Let D be a digraph with vertex-set $\{v_1, \ldots, v_n\}$ and arc-set A. For each $v_i \in A$, denote its ***out-set*** by $O(v_i) = \{x \in V | (v_i, x) \in A\}$, and let $\mathcal{F} = \{O(v_1), \ldots, O(v_n)\}$. The ***competition graph of D*** is the intersection graph $\Omega(\mathcal{F})$. A graph G is a ***competition graph*** if there exists a digraph D such that $G \cong \Omega(\mathcal{F})$ with $\mathcal{F}$ the family of out-sets of vertices of D.

EXAMPLE

E3: Figure 9.3.3 shows a digraph D and its competition graph G.

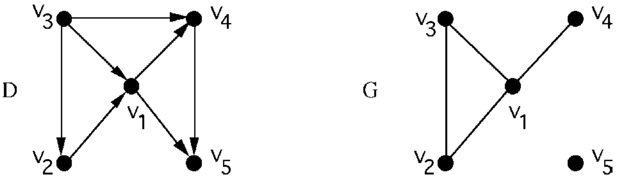


Figure 9.3.3 **A digraph D and its competition graph G.**

REMARK

R3: Competition graphs were orginally introduced by Cohen in [Co78] where the digraph was the food web of group of predator and prey species in an ecosystem. In this context, the digraph D is usually acyclic. Roberts in [Ro78] shows that after adding a sufficient number of isolated vertices to a graph, the resulting graph is the competition graph of an acyclic digraph. The smallest number of isolated vertices required to do this is called the ***competition number*** of the graph and this parameter has been extensively studied. (See Lu89, Ki93.) The following results give a sample of characterizations of competition graphs that are internal to the graphs. See [McMc99] for many others.

FACTS

F10: [DuBr83, LuMa83] A graph G is the competition graph of some acyclic digraph if and only if the vertex-set of G can be labeled $V(G) = \{v_1, \ldots, v_n\}$ such that there is an edge clique cover of G, $\{C_1, \ldots, C_k\}$, that satisfies $v_i \in C_j \Rightarrow i < j$.

F11: [DuBr83] A graph G is the competition graph of an arbitrary digraph if and only if $\theta(G) \leq |V(G)|$.

F12: [DuBr83] A graph G is the competition graph of a loopless digraph if and only if the vertex set of G can be labeled as $V(G) = \{v_1, \ldots, v_n\}$, and G has an edge clique cover $\{C_1, \ldots, C_k\}$ such that $v_i \in C_j$ implies $i \neq j$.

F13: [Gu98] A graph G with $|V(G)| \geq 3$ is the competition graph of a hamiltonian digraph if and only if G has an edge clique cover $\{C_1, \ldots, C_k\}$ with a *system of distinct representatives* (i.e., there exists $v_i \in C_i$, $i = 1, 2, \ldots, k$, such that $v_1, v_2, \ldots, v_k$ are all distinct).

9.3.2 Tolerance

Two influential papers, [GoMo82] and [GoMoTr84], introduced tolerance graphs as a generalization of interval graphs. In [JaMcMu91] a general model of tolerance was presented which extended tolerance graphs to allow different types of sets and measured intersection. Breaking with historical order, we introduce this more general model first.

DEFINITIONS

D14: Let $\mathcal{F} = \{S_1, \ldots, S_n\}$ be a family of subsets of a finite set S, let μ be a non-negative, real-valued function on the subsets of S, and assign to each S_i a positive real number called a ***tolerance***. Also, let ϕ be a non-negative, real-valued, symmetric function defined on pairs of positive real numbers. The ***ϕ-tolerance intersection graph*** G of the family $\mathcal{F}$ with respect to ϕ, μ, and the t_i's has vertex-set $V(G) = \mathcal{F}$ with $S_iS_j \in E(G)$ if and only if $i \neq j$ and $\mu(S_i \cap S_j) \geq \phi(t_i, t_j)$.

D15: A graph G is a ***ϕ-tolerance intersection graph*** if there exists $\mathcal{F}$, ϕ, μ, and t_i's, as defined above, such that G is isomorphic to the ϕ-tolerance intersection graph of $\mathcal{F}$ with respect to ϕ, μ, and the t_i's.

p-Intersection Graphs

Probably the simplest type of ϕ-tolerance intersection graph is when μ measures the size of the set, and ϕ is a constant function.

DEFINITIONS

D16: For each integer $p \geq 1$, the ***p-intersection graph*** of a family $\mathcal{F} = \{S_1, \ldots, S_n\}$ of subsets of a finite set S is defined to be the graph G having $V(G) = \mathcal{F}$ with $S_iS_j \in E(G)$ if and only if $i \neq j$ and $|S_i \cap S_j| \geq p$. A graph G is a ***p-intersection graph*** on S if there exists a family $\mathcal{F}$ and G is isomorphic to the p-intersection graph of $\mathcal{F}$.

D17: A ***p-edge clique cover*** of a graph G is a family $\{V_1, \ldots, V_m\}$ of not necessarily distinct subsets of $V(G)$ such that, for every set $\{i_1, \ldots, i_p\}$ of p distinct subscripts, $T = V_{i_1} \cap \cdots \cap V_{i_p}$ induces a complete subgraph of G, and such that the collection of sets of the form T is an edge clique cover of G.

D18: The ***p-intersection number*** of a graph G is the minimum cardinality of a set S such that G is a p-intersection graph on S.

D19: A ***p-competition graph*** is a graph isomorphic to the p-intersection graph of the family of out-sets of the vertices of some digraph.

FACTS

F14: [JaMcSc91] Every graph is a p-intersection graph on some set, for any positive integer p.

F15: [IsKiMcMcRo92] A graph G is the p-competition graph of an arbitrary digraph if and only if G has a p-edge clique cover of cardinality $|V(G)|$.

F16: For every graph G, the p-intersection number of G is equal to the minimum cardinality of a p-edge clique cover of G.

F17: [KiMcMcRo95] The p-intersection number of G is less than or equal to $i(G)+p-1$.

F18: [IsKiMcMcRo92] $K_{2,n}$ is the 2-competition graph of an arbitrary digraph if and only if $n = 1$ or $n \geq 9$.

F19: [Ja92] $K_{n,n}$ is not a 2-competition graph for $n \geq 2$.

REMARKS

R4: Clearly when $p = 1$, the p-intersection graphs and p-edge clique covers are the ordinary intersection graphs and edge clique covers, respectively.

R5: Determining p-intersection numbers is difficult, even for $p = 2$ (see [ChWe94, JaKeWe95, Ea97]).

R6: By restricting the type of sets in $\mathcal{F}$ to arbitrary intervals or subtrees, nothing interesting occurs in p-intersection graphs. For example, it is easy to show that for every $p \geq 1$, a graph is the p-intersection graph of a family of subtrees of some tree if and only if it is chordal. [McMc99]

R7: Fact 11 tells us that the complete bipartite graph $K_{m,n}$ is the competition graph of an arbitrary digraph if and only if $mn \leq n + m$.

Tolerance and Intervals

The notion of *tolerance* was introduced in [GoMo82, GoMoTr84].

NOTATION: We now consider $\mathcal{F} = \{I_1, \ldots, I_n\}$ to be a family of intervals of the real line.

DEFINITIONS

D20: A ***min-tolerance interval graph*** is a ϕ-tolerance intersection graph of a family $\mathcal{F} = \{I_1, \ldots, I_n\}$ of intervals of R in which $\mu(I_i) = |I_i|$, the length of the interval I_i, and $\phi(t_i, t_j) = \mathrm{min(t_i, t_j)}$.

TERMINOLOGY: Min-tolerance interval graphs were (and are often) referred to simply as *tolerance graphs*. Other types of ϕ-tolerance interval graphs, are ***max-tolerance interval graphs***, where $\phi(t_i, t_j) = \max(t_i, t_j)$, ***sum-tolerance interval graphs***, where $\phi(t_i, t_j) = t_i + t_j$, and so on.

D21: A graph is ***weakly chordal*** if it does not contain, as an induced subgraph, any C_n or the complement of C_n for any $n \geq 5$.

D22: A min-tolerance interval graph is ***bounded*** if it can be represented by tolerances where $t_i \leq |I_i|$ for all i.

EXAMPLE

E4: Clearly every interval graph is a min-tolerance interval graph, but the converse is false. The intervals and their assigned tolerances shown in Figure 9.3.4 below show that C_4 is a min-tolerance interval graph, but it is straightforward to show that C_4 is not an interval graph.

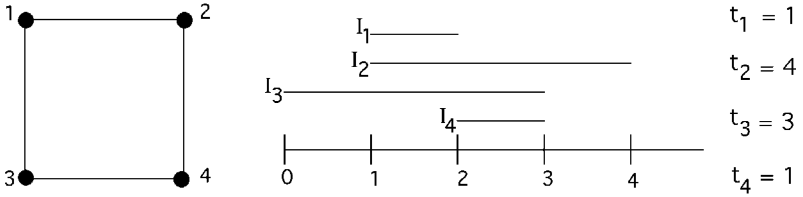


Figure 9.3.4 C_4 **is a min-tolerance interval graph.**

FACTS

F20: [GoMo82] A graph is an interval graph if and only if it is a min-tolerance interval graph with $t_i = t_j$ for all i, j if and only if it is a bounded min-tolerance graph with $t_i = t_j$ for all i, j.

F21: [GoMoTr84] Min-tolerance interval graphs are weakly chordal.

F22: [GoMoTr84] Min-tolerance interval graphs are perfect.

Some Special ϕ-Tolerance Interval Graphs

DEFINITIONS

D23: A ***ϕ-tolerance unit interval graph*** is a ϕ-tolerance interval graph using intervals all of the same length.

D24: A ***ϕ-tolerance proper interval graph*** is a ϕ-tolerance interval graph using intervals such that no interval is properly contained in another.

D25: The function ϕ is ***Archimedean*** if for every $c > 0$, $\lim_{x\to\infty} \phi(x, c) = \infty$ and $\lim_{x\to\infty} \phi(c, x) = \infty$.

D26: A graph G is an ***Archimedean ϕ-tolerance interval graph*** if G is a ϕ-tolerance interval graph for every Archimedean function ϕ.

FACTS

F23: [BoFiIsLa95] There exist min-tolerance proper interval graphs that are not min-tolerance unit interval graphs.

F24: [JaMc91] A graph is a sum-tolerance proper interval graph if and only if it is a sum-tolerance unit interval graph.

F25: [JaMcSc91] All trees are Archimedean ϕ-tolerance interval graphs.

F26: [GoJaTr03] For every k, $K_{2,k}$ is an Archimedean ϕ-tolerance interval graph.

REMARKS

R8: Results about when a ϕ-tolerance proper interval graph is a ϕ-tolerance unit interval graph remain elusive for functions ϕ other than *min* and *sum*.

R9: For ϕ-tolerance intersection graphs, the most frequently studied is when $\phi = \min$. However, min is clearly not Archimedean, whereas other natural functions such as sum, max and product are Archimedean. Indeed, as shown in [GoMoTr84], not all trees are min-tolerance interval graphs.

EXAMPLE

E5: The tree shown in Figure 9.3.5 is not a min-tolerance interval graph.

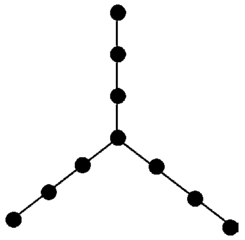


Figure 9.3.5 **A tree that is not a min-tolerance interval graph.**

Tolerance and Subtrees

From Remark 7, we note that restrictions must be placed on the host tree and/or subtrees in order to get nontrivial results.

REMARKS

R10: If we allow our trees to be continuous in the sense that a tree is a connected collection of curves in the plane without cycles, then interesting tolerance intersection graph results can be found in [BiDe93].

R11: We now see that with finite trees, degree restrictions on the trees present us with some challenging problems [JaMu00a, JaMu00b].

DEFINITIONS

D27: A graph has an ***(h, s, t)-representation*** if it is the t-intersection graph of a family of subtrees $\{S_1, \ldots, S_n\}$ of a tree T, where each S_i has maximum degree at most s and T has maximum degree at most h.

D28: An (h, s, t)-representation of a graph is ***leaf generated*** if every leaf of S_i is a leaf of T, where S_i and T are as above. If in addition, two subtrees S_i and S_j share a leaf if and only if they represent adjacent vertices in the graph, then the representation is ***orthodox***.

FACTS

F27: [JaMu00a] If the graph G has an (h, s, t)-representation for some $t \geq 2$, then G has an (h, s, t)-representation for all $u \geq t^2 - 4t + 6$.

F28: [JaMu00a] For all $h \geq s \geq 3$ and all $t \geq 1$, a graph G has an orthodox (h, s, t)-representation if and only if each block of G has an orthodox (h, s, t)-representation.

F29: [JaMu00a] Every graph has an orthodox $(3, 3, t)$-representation for every $t \geq 2n + n\log(n)$.

Tolerance and Competition graphs

DEFINITIONS

D29: Let ϕ be a symmetric function mapping pairs of non-negative integers to non-negative integers. A graph G is a ***ϕ-tolerance competition graph*** if and only if G is the ϕ-tolerance intersection graph of the family of out-sets of the vertices of some digraph.

D30: Let ϕ be as above and $T = (t_1, \ldots, t_n)$ be an n-tuple of not necessarily distinct non-negative integers. A ***ϕ-T-edge clique cover*** of the graph G is a family $\{V_1, \ldots, V_n\}$ such that $v_i v_j \in E(G)$ if and only if at least $\phi(t_i, t_j)$ of the sets V_k contain both v_i and v_j.

FACTS

F30: [BrMcVi95] A graph G is a ϕ-tolerance competition graph if and only if there exists a ϕ-T-edge clique cover of G with $|V(G)|$ elements.

F31: [BrMcVi95] (a) Every bipartite graph is a min-tolerance competition graph.
(b) $K_{2,n}$ is a max-tolerance competition graph for all $n \geq 1$.
(c) $K_{3,3}$ is not a sum-tolerance competition graph but is a sum-tolerance competition graph if non-integer tolerances are allowed.

F32: [BrCaVi00] If $\phi(t_i, t_j) = |t_i - t_j|$, then $K_{1,n}$ is a ϕ-tolerance competition graph if and only if $n \leq 17$ or $n = 19$.

REMARKS

R12: Interesting problems occur when the set of possible tolerances is restricted, even to $\{0, q\}$ (see [BrMcVi96]).

R13: In [BrCaVi00], the absolute-difference function is considered for ϕ, and the complete bipartite graphs that are ϕ-tolerance competition graphs are characterized.

R14: Many min-tolerance competition graphs are constructed in [AnLaLuMcMe94], but it is still an open problem to find a graph that is not a min-tolerance competition graph.

R15: More results on the various forms of ϕ-tolerance intersection graphs can be found in [GoTr03], which is the first book devoted exclusively to the topic of tolerance graphs. In this section we have only touched on three of many possible restrictions on the family $\mathcal{F}$: intervals, subtrees of a tree, outneighbor sets in a digraph.

References

[AnLaLuMcMe94] C. A. Anderson, L. Langley, J. R. Lundgren, P. A. McKenna and S. K. Merz, New classes of p-competition graphs and ϕ-tolerance competition graphs, *Congr. Numer.* 100 (1994), 97–107.

[Be59] S. Benzer, On the topology of the genetic fine structure, *Proc. Natl. Acad. Sci. U.S.A.* 45 (1959), 1607–1620.

[BiDe93] E. Bibelnieks and P. M. Dearing, Neighborhood subtree tolerance graphs, *Discrete Appl. Math.* 43 (1993), 13–26.

[BoFiIsLa95] K. P. Bogart, P. C. Fishburn, G. Isaak and L. J. Langley, Proper and unit tolerance graphs, *Discrete Appl. Math.* 60 (1995), 99–117.

[BrLeSp99] A. Brandstädt, V. B. Le and J. Spinrad, *Graph classes: A survey*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 1999.

[BrCaVi00] R. C. Brigham, J. R. Carrington and R. P. Vitray, Bipartite graphs and absolute difference tolerances, *Ars Combin.* 54 (2000), 3–27.

[BrMcVi95] R. C. Brigham, F. R. McMorris and R. P. Vitray, Tolerance competition graphs, *Linear Algebra Appl.* 217 (1995), 41–52.

[BrMcVi96] R. C. Brigham, F. R. McMorris and R. P. Vitray, Two-ϕ-tolerance competition graphs, *Discrete Appl. Math* 66 (1996), 101–108.

[Bu74] P. Buneman, A characterisation of rigid circuit graphs, *Discrete Math.* 9 (1974), 205–212.

[ChWe94] M. S. Chung and D. B. West, The p-intersection number of a complete bipartite graph and orthogonal double coverings of a clique, *Combinatorica* 14 (1994), 453–461.

[Co78] J. E. Cohen, *Food Webs and Niche Space*, Princeton University Press, Princeton, NJ, 1978.

[DuBr83] R. D. Dutton and R. C. Brigham, A characterization of competition graphs, *Discrete Appl. Math.* 6 (1983), 315–317.

[Ea97] N. Eaton, Intersection representations of complete unbalanced bipartite graphs, *J. Combin. Theory Ser. B* 71 (1997), 123–129.

[ErGoPo66] P. Erdős, A. W. Goodman and L. Pósa, The representation of a graph by set intersections, *Canad. J. Math.* 18 (1966), 106–112.

[Fi85] P. C. Fishburn, *Interval Orders and Interval Graphs*, Wiley, New York, 1985.

[FuGr65] D. R. Fulkerson and O. A. Gross, Incidence matrices and interval graphs, *Pacific J. Math.* 15 (1965), 835–855.

[Ga74] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, *J. Combin. Theory Ser. B* 16 (1974), 47–56.

[GiHo64] P. C. Gilmore and A. J. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math.* 16 (1964), 539–548.

[Go80] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, San Diego, 1980.

[GoJaTr03] M. C. Golumbic, R. E. Jamison and A. N. Trenk, Archimedean ϕ-tolerance graphs, manuscript, 2003.

[GoMo82] M. C. Golumbic and C. L. Monma, A generalization of interval graphs with tolerances, *Congr. Numer.* 35 (1982), 321–331.

[GoMoTr84] M. C. Golumbic, C. L. Monma and W. T. Trotter, Tolerance graphs, *Discrete Appl. Math.* 9 (1984), 157–170.

[GoTr03] M. C. Golumbic and A. N. Trenk, *Tolerance Graphs*, Cambridge University Press, to appear, 2003.

[Gu98] D. R. Guichard, Competition graphs of hamiltonian digraphs, *SIAM J. Discrete Math.* 11 (1998), 128–134.

[Ha57] G. Hajós, Über eine Art von Graphen, *Intern. Math. Nachr.* 11 (1957), Problem 65.

[IsKiMcMcRo92] G. Isaak, S.-K. Kim, T. A. McKee, F. R. McMorris and F. S. Roberts, 2-competition graphs, *SIAM J. Discrete Math.* 5 (1992), 524–538.

[Ja92] M. S. Jacobson, On the p-edge clique cover number of complete bipartite graphs, *SIAM J. Discrete Math.* 5 (1992), 539–544.

[JaKeWe95] M. S. Jacobson, A. E. Kézdy and D. B. West, The 2-intersection number of paths and bounded-degree trees, *J. Graph Theory* 19 (1995), 461–469.

[JaMc91] M. S. Jacobson and F. R. McMorris, Sum-tolerance proper interval graphs are precisely sum-tolerance unit interval graphs, *J. Combin. Inform. System Sci.* 16 (1991), 25–28.

[JaMcMu91] M. S. Jacobson, F. R. McMorris and H. M. Mulder, An introduction to tolerance intersection graphs, In *Graph Theory, Combinatorics and Applications* (Y. Alavi, et al., Eds.) Wiley Interscience, New York, vol. 2 (1991), pp. 705–723.

[JaMcSc91] M. S. Jacobson, F. R. McMorris and E. R. Scheinerman, General results on tolerance intersection graphs, *J. Graph Theory* 15 (1991), 573–577.

[JaMu00a] R. E. Jamison and H. M. Mulder, Constant tolerance intersection graphs in trees, *Congressus Numer.* 143 (2000), 175–192.

[JaMu00b] R. E. Jamison and H. M. Mulder, Tolerance intersection graphs on binary trees with constant tolerance 3, *Discrete Math.* 215 (2000), 115–131.

[Ki93] S.-R. Kim, The competition number and its variants, In *Quo Vadis, Graph Theory?* (J. Gimbel, et al., Eds.), [*Ann. Discrete Math.* vol. 55], North-Holland, Amsterdam, 1993, pp. 313–326.

[KiMcMcRo95] S.-R. Kim, T. A. McKee, F. R. McMorris and F. S. Roberts, p-competition graphs, *Linear Algebra Appl.* 217 (1995), 167–178.

[KoStWo78] L. T. Kou, L. J. Stockmeyer and C. K. Wong, Covering graphs by cliques with regard to keyword conflicts and intersection graphs, *Comm. ACM* 21 (1978), 135–139.

[LeBo62] C. G. Lekkerkerker and J. C. Boland, Representation of a finite graph by a set of intervals on the real line, *Fund. Math.* 51 (1962), 45–64.

[Lu89] J. R. Lundgren, Food webs, competition graphs, competition-common enemy graphs, and niche graphs. In *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences* (F. S. Roberts, Ed.), Springer, New York, 1989, pp. 221–243.

[LuMa83] J. R. Lundgren and J. S. Maybee, A characterization of graphs of competition number m, *Discrete Appl. Math.* 6 (1983), 319–322.

[Ma45] E. (Szpilrajn-) Marczewski, Sur deux propriétés des classes d'ensembles, *Fund. Math.* 33 (1945), 303–307.

[McMc99] T. A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, PA, 1999.

[MiRo84] B. G. Mirkin and S. N. Rodin, *Graphs and Genes*, Springer, Berlin, 1984.

[Ro69] F. S. Roberts, Indifference graphs, In *Proof Techniques in Graph Theory* (F. Harary, Ed.), Academic Press, New York, 1969, pp. 139–146.

[Ro78] F. S. Roberts, Food webs, competition graphs, and the boxicity of ecological phase space, In *Theory and Applications of Graphs* (Y. Alavi & D. Lick, Eds.) [*Lecture Notes in Mathematics* vol. 642], Springer, New York, 1968, pp. 477–490.

[Ro70] D. J. Rose, Triangulated graphs and the elimination process, *J. Math. Anal. Appl.* 32 (1970), 597–609.

[Wa78] J. R. Walter, Representations of chordal graphs as subtrees of a tree, *J. Graph Theory* 2 (1978), 265–267.

9.4 BANDWIDTH

Robert C. Brigham, University of Central Florida

9.4.1 Fundamentals
9.4.2 Elementary Results
9.4.3 Bounds on Bandwidth
9.4.4 On the Bandwidth of Combinations of Graphs
9.4.5 Bandwidth and Its Relationship to Other Invariants
9.4.6 Related Concepts
References

Introduction

Harper [Ha64] discusses a coding problem in which the integers $1, 2, \ldots, 2^n$ form the code words, each assigned to a vertex of the n-dimensional hypercube. If code words i and j are assigned to adjacent vertices, then Δ_{ij} is defined to be $|i - j|$. The paper determines the minimum value of $\sum \Delta_{ij}$ over all possible assignments. In concluding remarks Harper says "Another problem, as yet not solved, is this: how to number the vertices of an n-cube so that $max\Delta_{ij}$ ... is minimized." This latter problem is precisely that of determining the bandwidth of a hypercube, and this is the first known reference in graph theoretic terms.

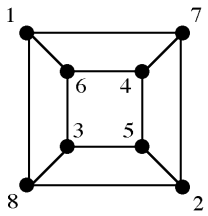
[ChChDeGi82] and [LaWi99], each with extensive bibliographies, provide comprehensive surveys of bandwidth. Further results can be found in [Ch88] and [Mi91]. All graphs discussed in this section are assumed to be simple and finite.

9.4.1 Fundamentals

The Bandwidth Concept

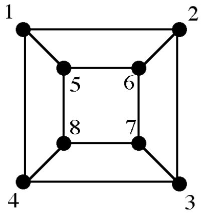
EXAMPLE

E1: Figures 9.4.1 and 9.4.2 show different vertex labelings of the three-dimensional hypercube Q_3 and the corresponding adjacency matrices.



$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 9.4.1 **Hypercube Q_3 with labeling and associated adjacency matrix.**



$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Figure 9.4.2 A second labeling of Q_3 and associated adjacency matrix.

For the labeling of Figure 9.4.1, all ones in the adjacency matrix lie in the seven diagonals above and below the main diagonal. Since the matrix is symmetric, we can restrict our attention to the diagonals in the upper triangular portion. The matrix corresponding to the labeling of Figure 9.4.2 has ones in only the four ("upper") diagonals closest to the main diagonal. The bandwidth of a graph corresponds to the minimum number of such diagonals, taken over all possible labelings. For Q_3 this number is four. Thus the labeling of Figure 9.4.2 yields the minimum. This concept is formalized as follows.

DEFINITIONS

NOTATION: An edge with endpoints (end vertices) u and v is denoted uv.

D1: A ***proper numbering*** of G is a bijection $f : V \rightarrow \{1, 2, \ldots, n\}$.

D2: Let f be a proper numbering of a graph G. The ***bandwidth of*** f, denoted $B_f(G)$, is given by $B_f(G) = max\{|f(u) - f(v)| : uv \in E\}$.

D3: The ***bandwidth*** of G is $B(G) = min\{B_f(G) : f$ is a proper numbering of $G\}$.

D4: A ***bandwidth numbering*** of G is a proper numbering f such that $B(G) = B_f(G)$ (i.e., a proper numbering that achieves $B(G)$).

Applications

The introduction mentions an application of bandwidth related to coding theory. [LaWi99] includes a survey, along with references, of other applications.

EFFICIENT MATRIX STORAGE

Storing the entire upper triangular portion of the adjacency matrix is one of several computer representations of a graph. However, only the $B(G)$ diagonals above the main diagonal need be stored. If $B(G)$ is small, this can represent a significant savings.

VLSI LAYOUT

The ***placement problem*** for modules of a VLSI design is the location of the modules on a two-dimensional grid so that certain criteria are met. Modules that must communicate with each other should be as close as possible. A simplified model of the geometry can be given by a graph G whose vertices correspond to modules and whose edges correspond to wires between modules. Then $B(G)$ represents the maximum distance between communicating modules.

INTERCONNECTION NETWORKS

An ***interconnection*** or ***parallel computation network*** is a collection of processors with links between them. This can be modeled by a graph G where the vertices represent the processors and edges correspond to the links. Sometimes it is desirable to simulate the network represented by G on a second network modeled by graph H. This can be done by a one-to-one mapping $f : V(G) \to V(H)$, where processor u in G is simulated by processor $f(u)$ in H, and link uv in G is simulated by a shortest path between $f(u)$ and $f(v)$ in H. If t is the communication time for a link uv in G, then the corresponding time in H is dt where d is the distance between $f(u)$ and $f(v)$ in H. If $t = 1$ and H is a path, the greatest possible delay in the simulation is $B(G)$.

BINARY CONSTRAINT SATISFACTION PROBLEM

A ***binary constraint satisfaction problem*** involves a collection of variables, a set of possible values for each, and constraints between them. The problem is to assign to each variable a permissible value such that all constraints are satisfied. The associated ***constraint graph*** G has vertices representing the variables, with an edge between two vertices if the corresponding variables share a nontrivial constraint. If $B(G)$ is small, the problem may be more easily solvable than otherwise since then it might be possible to deal with only a small number of variables at a time.

Algorithms

DEFINITIONS

D5: The ***bandwidth decision problem*** is the problem which accepts as input an arbitrary graph G and an arbitrary integer K and returns "YES" if $B(G) \leq K$ and "NO" otherwise.

D6: For a given fixed positive integer k, the ***bandwidth-k decision problem*** is the problem which accepts as input an arbitrary graph G and returns "YES" if $B(G) \leq k$ and "NO" otherwise.

D7: A ***polynomial algorithm*** for graphs is one whose execution time is bounded by a polynomial in some parameter of the problem, often the number of vertices.

D8: An ***NP-complete problem*** is a problem having a "YES" or "NO" answer that can be solved nondeterministically in polynomial time, and all other such problems can be transformed to it in polynomial time. Such problems are generally accepted as being computationally difficult.

FACTS

F1: [Pa76] The bandwidth decision problem is NP-complete.

F2: [GaGrJoKn78] The bandwidth decision problem is NP-complete for trees with maximum degree three.

F3: [Sa80] The bandwidth-k problem is solvable in polynomial time for any fixed positive integer k.

REMARKS

R1: The important distinction between the bandwidth decision problem and the bandwidth-k decision problem is that the integer K in the former is an input variable while the integer k in the latter is fixed for all graphs and does not appear as an input.

R2: In view of Facts 1 and 2 above, it is highly unlikely that a polynomial algorithm can be found for computing the bandwidth of all graphs or even for trees with maximum degree three. Sometimes, though, it is possible to restate an NP-complete problem, by limiting its generality, so that the revised problem becomes polynomial. This has been done for bandwidth, as illustrated in Fact 3.

R3: Many approximation algorithms have been developed, some dealing with a matrix equivalent of the bandwidth problem, and a listing of several of them is given in [LaWi99]. Further surveys and references are in [Ev79, HeGr79, Sm85]. While it certainly would be advantageous to determine bandwidth exactly, approximate values remain useful in practical applications, since any reduction in the number of nonzero diagonals of the adjacency matrix provides benefit.

R4: Apparently the first attempt to develop an approximate algorithm was reported in [AlMa65], and its effective use was limited to small matrices. The first algorithm to receive wide acceptance is discussed in [Ro68]. [CuMc69] describes an algorithm which took center stage during the 1970's, despite several limitations. [GiPoSt76] presents a greatly improved version. Details can be found in [ChChDeGi82].

R5: More recent work on both approximate algorithms for general graphs and exact algorithms for specific classes of graphs includes [GoOp90, KaSh96, HaMa97, Ya98, BlKoRaVe00, Fe00, Gu01, KlTa01, CaMaPr02, KrSt02].

9.4.2 Elementary Results

The Bandwidth of Some Common Families of Graphs

DEFINITIONS

D9: The ***complete k-partite graph*** $K_{n_1,n_2,\ldots,n_k}$ is the graph whose vertex set is partitioned into sets A_i of n_i vertices, $1 \le i \le k$, with two vertices adjacent if and only if they are in distinct sets.

D10: The ***n-dimensional hypercube*** Q_n is the graph having 2^n vertices, each labeled with a distinct n-digit binary sequence, and two vertices are adjacent if and only if their labels differ in exactly one position.

FACTS

F4: $B(P_n) = 1$, where P_n is the path having n vertices.

F5: $B(C_n) = 2$, where C_n is the cycle having n vertices.

F6: $B(K_n) = n - 1$, where K_n is the complete graph having n vertices.

F7: [Ei79] Let $n_1 \geq n_2 \geq \ldots \geq n_k$ be positive integers. Then $B(K_{n_1,n_2,\ldots,n_k}) = |V(K_{n_1,n_2,\ldots,n_k})| - \lceil (n_1+1)/2 \rceil$. Thus, $B(K_{n_1,n_2}) = \lceil n_1/2 \rceil + n_2 - 1$ [Ch70].

F8: [Ha66] $B(Q_n) = \sum_{k=0}^{n-1} \binom{k}{\lfloor k/2 \rfloor}$.

EXAMPLE

E2: Figure 9.4.2 shows a bandwidth numbering of Q_3 and Figure 9.4.3 presents bandwidth numberings for C_7 and $K_{5,3}$.

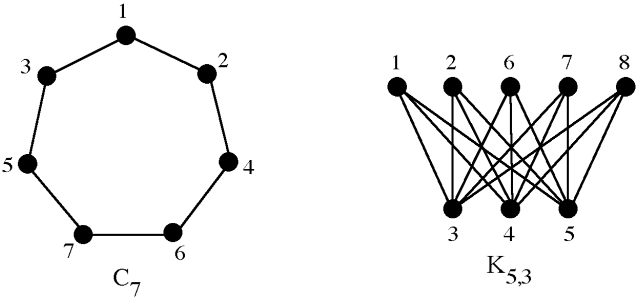


Figure 9.4.3 **Bandwidth numberings of C_7 and $K_{5,3}$.**

A Few Basic Relations

NOTATION: The *minimum degree* and *maximum degree* of the vertices in a graph G are denoted $\delta_{\min}(G)$ and $\delta_{\max}(G)$, respectively.

TERMINOLOGY NOTE: Several authors use $\delta(G)$ and $\Delta(G)$ instead.

FACTS

F9: [ChDeGiKo75] $B(G) \geq \lceil \delta_{\max}(G)/2 \rceil$.

F10: If H is a subgraph of G, then $B(H) \leq B(G)$.

F11: If graph G has components $G_1, G_2, \ldots, G_k$, then $B(G) = max\{B(G_1), B(G_2), \ldots, B(G_k)\}$.

F12: [ChDeGiKo75] If G is a nonplanar graph, then $B(G) \geq 4$.

On the Bandwidth of Trees

DEFINITIONS

D11: An (edge) ***subdivision*** of edge $e = uv$ in graph G is the graph obtained from G by replacing e by the path $< u, w, v >$ where w is a new vertex of degree two. A ***refinement*** of G is a graph obtained from G by a finite number of subdivisions.

D12: The ***complete k-ary tree*** $T_{k,d}$ ***of depth*** d is the rooted tree in which all vertices at level $d-1$ or less have exactly k children, and all vertices at level d are leaves.

FACTS

F13: [ChDeGiKo75] For any tree T, $B(T) \leq \lfloor |V(T)|/2 \rfloor$. Equality holds if and only if $|V(T)|$ is even and T is the *star* $K_{1,|V(T)|-1}$.

F14: [WaYa95, AnKaGe96] Let T be a tree with k univalent vertices. Then $B(T) \leq \lceil k/2 \rceil$.

F15: [Sm95] Let $T_{k,d}$ be the complete k-ary tree of depth d. Then $B(T_{k,d}) = \lceil k(k^d - 1)/(2d(k-1)) \rceil$.

F16: [Ch88] If tree T contains a refinement of the complete binary tree $T_{2,d}$, then $B(T) \geq \lceil d/2 \rceil$.

EXAMPLE

E3: Figure 9.4.4 shows bandwidth numberings of two trees, one for which $B(T) < \lfloor n/2 \rfloor$ and $K_{1,7}$ for which equality occurs.

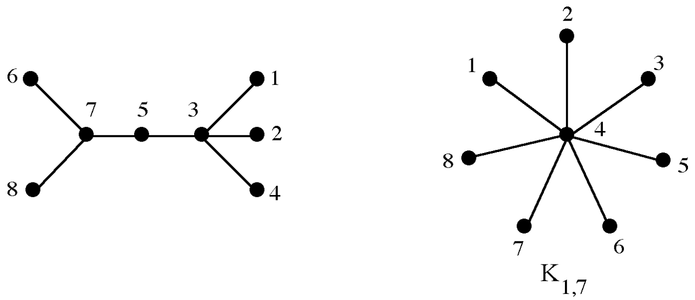


Figure 9.4.4 **Bandwidth numberings of two trees.**

Alternative Interpretations of Bandwidth

Three alternative interpretations of bandwidth are shown below. Others are given in [Li00].

DEFINITIONS

D13: The k^{th} ***power*** of graph G, denoted G^k, is the graph having the same vertex set as G and an edge between two vertices if and only if the distance between them is at most k in G.

D14: The ***complementary numbering*** f_c ***of proper numbering*** f of G is defined by $f_c(v) = n + 1 - f(v)$ for each vertex v of G.

FACTS

F17: [ChChDeGi82] For a real symmetric matrix M, let m_{ij} be the value in position (i, j). Consider the problem of finding a symmetric permutation of the rows and columns of M such that the maximum of $|i - j|$, taken over all pairs (i, j) for which m_{ij} is nonzero, is minimized. This problem's equivalence to the bandwidth problem follows by replacing each nonzero entry of M by 1 and considering the resulting matrix as an adjacency matrix of a graph.

F18: [ChChDeGi82] G has bandwidth k if and only if k is the smallest integer such that G can be embedded in P_n^k where P_n^k is the kth power of the path P_n on n vertices.

F19: $B_{f_c}(G) = B_f(G)$ so the complementary numbering of a bandwidth numbering also is a bandwidth numbering.

9.4.3 Bounds on Bandwidth

Two General Bounds

FACTS

F20: [Ha66] For $S \subseteq V$, let ∂S be the subset of S with at least one neighbor outside of S. Then $B(G) \geq \max_k \min\{|\partial S| : |S| = k\}$.

F21: [Ch80-a] For $S \subseteq V$, let ΔS be the subset of edges of G with exactly one endpoint in S. Then $B \geq \max_k \min\left\{\frac{(1+8\Delta|S|)^{1/2}-1}{2} : |S| = k\right\}$.

Subdivisions, Mergers, Contractions, and Edge Additions

DEFINITION

D15: The ***merger of two vertices*** u ***and*** v ***of graph*** G is the graph, denoted $G|_{u,v}$, obtained from G by identifying u and v and then eliminating any loops and duplicate edges. If $e = uv$, the merger $G|_{u,v}$ is called a ***contraction*** of G along e and denoted $G|_e$.

FACTS

F22: [ChOp86] If H is obtained from G by a subdivision of an edge, then

$$B(H) \geq \lceil (3B(G) - 1)/4 \rceil$$

and this result is sharp.

F23: [ChOp86] For any graph G and vertices $u, v \in V(G)$,

$$B(G) - 1 \leq B(G|_{u,v}) \leq 2B(G)$$

and both bounds are sharp.

F24: [Ch80-a, ChOp86] For any graph G and edge $e \in E(G)$,

$$B(G) - 1 \leq B(G|_e) \leq \lceil [3B(G) - 1]/2 \rceil$$

and both bounds are sharp.

F25: [WaWeYa95] Let $B(G) = b$ and $g(b, |V(G)|)$ be the maximum possible value of $B(G + e)$. Then $g(b, |V(G)|) = \begin{cases} b+1 & \text{if } |V(G)| \leq 3b+4 \\ \lceil (|V(G)| - 1)/3 \rceil & \text{if } 3b+5 \leq |V(G)| \leq 6b-2 \\ 2b & \text{if } |V(G)| \geq 6b-1 \end{cases}$.

EXAMPLE

E4: Figure 9.4.5 illustrates Fact 25 by showing a graph G having bandwidth 2 and a corresponding $G+e$ having bandwidth 4, both shown with bandwidth numberings.

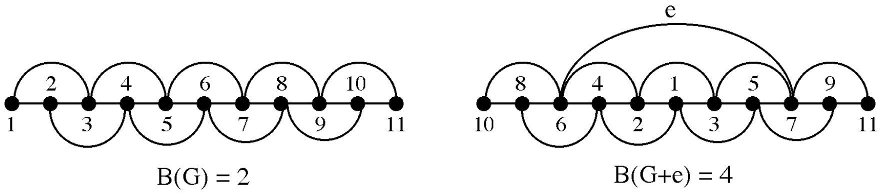


Figure 9.4.5 A graph whose bandwidth doubles when an edge is added.

REMARK

R6: Fact 25 gives a complete solution to a question originally posed by Erdös: whether $B(G+e) - B(G) \leq 1$, where $G+e$ is a graph obtained from G by adding an edge e not originally in G. This was first shown not to be the case in [Ch80-a].

Nordhaus-Gaddum Types of Bounds

DEFINITIONS

D16: The ***complement of graph*** G, denoted $\overline{G}$, is the graph with the same vertex set as G and $e \in \overline{G}$ if and only if $e \notin G$.

D17: A property P of a graph holds for ***almost all graphs*** if the ratio of the number of n-vertex graphs possessing P divided by the number of n-vertex graphs approaches one as n approaches infinity.

FACTS

F26: [ChErChGr81] For any graph G, $|V(G)| - 2 \leq B(G) + B(\overline{G})$.

F27: [ChErChGr81] There is a positive constant c_1 such that $B(G)+B(\overline{G}) \leq 2|V(G)| - c_1 \log |V(G)|$ for any graph.

F28: [ChErChGr81] There is a positive constant c_2 such that $2|V(G)| - c_1 \log |V(G)| \leq B(G) + B(\overline{G})$ for almost all graphs.

F29: [FuWe01] Let $f(n) = \max\{B(G) + B(\overline{G}) : G$ an n-vertex graph$\}$. Then $2n - \lceil (4+2\sqrt{2}) \log_2 n \rceil \leq f(n) \leq 2n - 4\log_2 n + o(\log n)$.

Other Bounds

FACTS

F30: [ChDeGiKo75] Let G be a graph and G^k be its k^{th} power. Then $B(G^k) \leq kB(G)$.

F31: [Ch80-a] For graph G, $B(G) \leq |V(G)| - 3$ if and only if $\overline{G}$ contains a P_4.

F32: [Ch80-a] Let G be a graph such that $d(x,y) \leq 2$ for every pair of vertices x, y. Then $B(G) = |V(G)| - 2$ if and only if every component of $\overline{G}$ is a vertex, a $K_{1,n}$, or a K_3.

9.4.4 On the Bandwidth of Combinations of Graphs

Two or more graphs can be combined in a variety of ways to form a new graph, and information about the bandwidth of the new graph often can be gleaned from the bandwidths of the original graphs.

Cartesian Product

DEFINITION

D18: The ***Cartesian product*** of graphs G and H, denoted $G \times H$, is the graph where $V(G \times H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G \times H)$ if and only if either (i) $g_1 = g_2$ and $h_1 h_2 \in E(H)$ or (ii) $h_1 = h_2$ and $g_1 g_2 \in E(G)$.

FACTS

F33: [Ch75, ChDeGiKo75] For graphs G and H,

$$B(G \times H) \leq \min\{|V(H)|B(G), |V(G)|B(H)\}.$$

F34: [Ch75] For paths P_m and P_n, where $\max\{m, n\} \geq 2$, $B(P_m \times P_n) = \min\{m, n\}$.

F35: If $m \geq 2$ and $n \geq 3$, then $B(P_m \times C_n) = \min\{2m, n\}$.

EXAMPLE

E5: Figure 9.4.6 shows a bandwidth numbering of $P_3 \times C_7$ which, by Fact 35, has bandwidth 6.

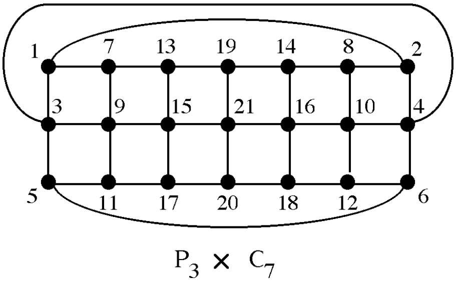


Figure 9.4.6 **A bandwidth numbering of $P_3 \times C_7$.**

Sum of Two Graphs

DEFINITION

D19: The ***sum*** of graphs G and H, denoted $G + H$, is the graph with $V(G + H) = V(G) \cup V(H)$ and $E(G + H) = E(G) \cup E(H) \cup \{uv : u \in V(G) \text{ and } v \in V(H)\}$.

FACTS

F36: [LiWiWa91] Let G and H be graphs such that $|V(G)| \geq |V(H)|$ and $B(G) < \lceil |V(G)|/2 \rceil$. Then $B(G + H) = \lceil |V(G)|/2 \rceil + |V(H)| - 1$.

F37: [LiWiWa91] Let G and H be graphs such that $|V(G)| \geq |V(H)|$ and $B(G) > \lceil |V(G)|/2 \rceil$. Then $|V(G)|/2 + |V(H)| - 1 \leq B(G+H) \leq \min\{B(G) + |V(H)|, \max\{B(H) + |V(G)|, \lceil |V(H)|/2 \rceil + |V(G)| - 1\}\}$.

F38: [LiWiWa91] For paths P_n and P_m with $n \geq m$, $B(P_n + P_m) = \lceil n/2 \rceil + m - 1$.

F39: [LiWiWa91] For cycles C_n and C_m with $n \geq m$,

$$B(C_n + C_m) = \begin{cases} \lceil n/2 \rceil + m - 1 & \text{if } n \geq 5 \\ 5 & \text{if } n \leq 4 \text{ and } m = 3 \\ 6 & \text{if } n = m = 4 \end{cases}$$

EXAMPLE

E6: Figure 9.4.7 shows a bandwidth numbering of $C_5 + C_3$ which, by Fact 39, has bandwidth 5.

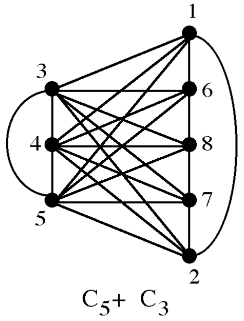


Figure 9.4.7 **A bandwidth numbering of $C_5 + C_3$.**

Corona and Composition

DEFINITIONS

D20: The ***corona*** of graphs G and H, denoted $G \circ H$, is the graph constructed from one copy of G and $|V(G)|$ copies of H, one associated with each vertex of G. If $v \in V(G)$ and H_v is the copy of H associated with v, there are the additional edges vh for every $h \in V(H_v)$.

D21: The ***composition*** of graphs G and H, denoted $G(H)$, is the graph where $V(G(H)) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(H))$ if and only if either (i) $g_1 g_2 \in E(G)$ or (ii) $g_1 = g_2$ and $h_1 h_2 \in E(H)$.

FACTS

F40: [Ch80-b] For graphs G and H, $B(G \circ H) \leq B(G)(|V(H)| + 1)$, and this bound is sharp.

F41: [Ch80-b] For graphs G and H, $B(G(H)) \leq (B(G)+1)|V(H)| - 1$, and this bound is sharp.

Strong Product and Tensor Product

DEFINITIONS

D22: The ***strong product*** of graphs G and H, denoted $G(Sp)H$, is the graph with $V(G(Sp)H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(Sp)H)$ if and only if either $g_1g_2 \in E(G)$ and $h_1h_2 \in E(H)$ or $g_1 = g_2$ and $h_1h_2 \in E(H)$ or $h_1 = h_2$ and $g_1g_2 \in E(G)$.

D23: The ***tensor product*** of graphs G and H, denoted $G(Tp)H$, is the graph with $V(G(Tp)H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(Tp)H)$ if and only if $g_1g_2 \in E(G)$ and $h_1h_2 \in E(H)$.

FACTS

F42: [LaWi95] (a) If $m \geq n \geq 2$, $B(P_m(S_p)P_n) = n + 1$.

(b) If $m \geq 3$ and $n \geq 2$, $B(C_m(S_p)P_n) = \begin{cases} m+2 & \text{if } n \geq \lfloor n/2 \rfloor + 1 \\ 2n+1 & \text{otherwise} \end{cases}$

(c) If $m \geq n$, $B(C_m(S_p)C_n) = 2n + 2$.

F43: [LaWi97-b]

$$B(C_m(T_p)C_n) = \begin{cases} n+1 & \text{if } m \geq n \geq 4 \text{ and } m, n \text{ even} \\ \min\{n+1, 2m+1\} & \text{if } m \text{ odd, } n \text{ even} \\ 2n+1 & \text{if } m \geq n \geq 3 \text{ and } m, n \text{ odd} \end{cases}$$

9.4.5 Bandwidth and Its Relationship to Other Invariants

Many bounds for bandwidth in terms of other graphical invariants have been found. Several are listed in the previously mentioned survey papers and in [BrDu85, BrDu91].

NOTATION: Throughout this subsection, G is a graph with $V = V(G)$, $E = E(G)$, $B = B(G)$, $\delta_{\max} = \delta_{\max}(G)$, and $\delta_{\min} = \delta_{\min}(G)$.

Vertex Degree

DEFINITION

D24: The ***degree sequence*** of graph G is a listing of the degrees of the vertices of G, usually in monotonic order.

FACTS

F44: [Ch70] If the graph G has the degree sequence $d_1 \leq d_2 \leq \ldots \leq d_n$, then $B \geq \max_j\{d_j - \lfloor (j-1)/2 \rfloor, d_j/2\}$. Setting $j = 1$ yields $B \geq \delta_{\min}$.

F45: [ChChDeGi82] If G contains no copies of K_3, $B \geq \lfloor (3\delta_{\min} - 1)/2 \rfloor$.

EXAMPLE

E7: Two graphs that show Fact 45 is sharp are the 3-dimensional hypercube Q_3 and the graph along with the bandwidth numbering shown in Figure 9.4.8. Both examples have no K_3, $\delta_{\min} = 3$, and bandwidth $B = \lfloor(3(3) - 1)/2\rfloor = 4$.

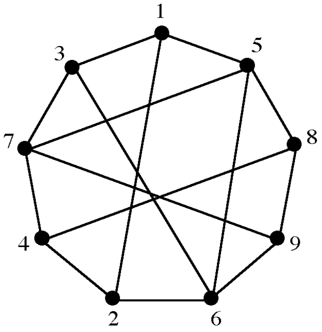


Figure 9.4.8 **Bandwidth numbering of a graph with no K_3, $\delta_{\min} = 3$, $B = 4$.**

Number of Vertices and Edges for Arbitrary Graphs

FACTS

F46: [DuBr89] $B \leq (|E| + 1)/2$.

F47: [BrDu85] $B \geq \left\lceil 2|V| - 1 - \sqrt{(2|V| - 1)^2 - 8|E|}\right\rceil /2$.

F48: [LaWi97-a] If G is connected, then $B \geq |V| - s$, where s is the largest integer such that $s(s - 1) \leq |V|(|V| - 1) - 2|E|$, and the bound is sharp.

F49: [DuBr89] If $B \geq |V|/2$, then $|E| \geq |V|(|V| - 1)/[2(|V| - B)]$.

F50: [DuBr89] If $B \geq |V|/2$, then $|E| \geq (2\lfloor|V|/2\rfloor - 1)\,[|V|/(|V| - 2)]^{B - \lfloor|V|/2\rfloor}$.

F51: [AlLiMcEr92] Let $B = \lceil(1 - \epsilon)|V|\rceil$ with $0 < \epsilon < 1$. Then there are positive constants c_1 (which depends on ϵ) and c_2 such that $c_1|V|/\epsilon \leq m(|V|, B) \leq c_2(\log(2/\epsilon)|V|/\epsilon$ where $m(|V|, B)$ is the minimum possible number of edges in a graph with $|V|$ vertices and bandwidth B.

Number of Vertices and Edges for Graphs with no K_3

FACTS

F52: [ChTr84] Let $t(n, B)$ be the maximum number of edges that an n-vertex graph having no K_3 and bandwidth at most B can have. Then $\left(2 - \sqrt{2}\right) nB \leq t(n, B) \leq \frac{5+\sqrt{3}}{11} nB$.

F53: [BrCaDuFiVi00] Let G be a bipartite graph with partite set sizes m and n, $m \leq n$, bandwidth B, and $(\lceil m/2\rceil + 1)B \leq n \leq (m + 1)B - 1$. Then $|E| \leq 2mB - 2m - 3 + \lceil(n + 1)/B\rceil + \lfloor(n + 1)/B\rfloor$, and this bound is sharp.

F54: [BrCaDuFiVi00] Let G be a bipartite graph with partite set sizes m and n, $m \leq n = (t+1)B + \lfloor \alpha B \rfloor$ where α is a fixed constant such that $0 \leq \alpha < 1$, and $B \geq (m + t + 4\lfloor m/t \rfloor - 5/2)/(1-\alpha)$. Then $|E| \leq 2mB - \lfloor m/t \rfloor(2m - t\lfloor m/t \rfloor - t)$, and this bound is sharp.

Radius and Diameter

DEFINITIONS

D25: The ***radius*** of graph G, denoted $rad(G)$, is the smallest number r such that there is a vertex u of G with distance at most r from every other vertex of G.

D26: The ***diameter*** of graph G, denoted $diam(G)$, is the maximum distance between any two vertices of G.

FACTS

F55: [ChChDeGi82] For any graph G, $B \leq \delta_{\max}(\delta_{\max} - 1)^{rad(G)-1}$.

F56: [Ch70, ChDeGiKo75] For any graph G, $\lceil (|V| - 1)/diam(G) \rceil \leq B \leq |V| - diam(G)$.

F57: [ChSe89] For any graph G, $B \geq \max\{(|V(G')| - 1)/diam(G')\}$ where the maximum is taken over all connected subgraphs G' of G that have at least two vertices.

REMARK

R7: Paths and cycles achieve the lower bound of Fact 56. Figure 9.4.9 shows a bandwidth numbering of a graph having $|V| = 9$, diameter $= 4$, and $B = 5$, so the graph achieves the upper bound.

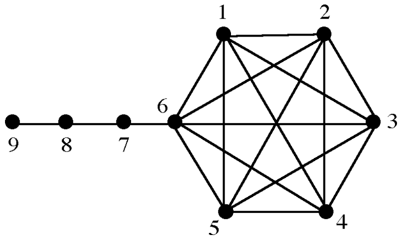


Figure 9.4.9 **Bandwidth numbering of a graph with $|V| = 9$, $d = 4$, $B = 5$.**

Vertex and Edge Chromatic Number

DEFINITIONS

D27: The (vertex) ***chromatic number*** of graph G, denoted $\chi(G)$, is the smallest number k such that there is a function $f : V(G) \to \{1, 2, \ldots, k\}$ with the property that, if uv is an edge, then $f(u) \neq f(v)$.

D28: The ***edge-chromatic number*** of graph G, denoted $\chi'(G)$, is the smallest number k such that there is a function $f : E(G) \to \{1, 2, \ldots, k\}$ with the property that, if edges e_1 and e_2 share a common vertex, then $f(e_1) \neq f(e_2)$.

FACTS

F58: [ChDeGiKo75] For any graph G, $B \geq \chi(G) - 1$.

F59: [BrDu85] For any graph G, $B \geq \chi'(G)/2$.

Vertex Independence and Vertex Cover Numbers

DEFINITIONS

D29: The ***vertex independence number*** of graph G, denoted $ind(G)$, is the largest cardinality of a set of vertices which induces a graph with no edges.

D30: The ***vertex cover number*** of graph G, denoted $\alpha_0(G)$, is the smallest cardinality of a set of vertices such that every edge is incident to at least one of the vertices in the set.

FACTS

F60: [Ch70, ChDeGiKo75] For any graph G,

$$\lceil |V|/ind(G) \rceil - 1 \leq B \leq |V| - \lfloor ind(G)/2 \rfloor - 1$$

F61: [De76] For any graph G, $B \geq \alpha_0(G)/ind(G)$.

Girth, Vertex Arboricity, and Thickness

DEFINITIONS

D31: The ***girth*** of graph G, denoted $girth(G)$, is the size of a smallest induced cycle of G.

D32: The ***vertex arboricity*** of graph G, denoted $arbor(G)$, is the minimum number of subsets into which $V(G)$ can be partitioned such that the vertices of each subset induce an acyclic subgraph.

D33: The ***thickness*** of graph G, denoted $thick(G)$, is the smallest number of planar subgraphs of G whose union is G.

FACTS

F62: [BrDu91] If G is not a forest, then $B \geq (girth(G) - 1)(arbor(G) - 2) + 2$.

F63: [BrDu91] If G is not a forest, then

$$B \geq [(girth(G) - 1)|V|/(2 \cdot ind(G)] - girth(G) + 2$$

F64: [BrDu85] For any graph G, $thick(G) \leq \max(B/2, 1)$.

9.4.6 Related Concepts

The study of bandwidth has spawned investigations into a variety of related ideas.

Bandsize

Bandsize is discussed briefly in [LaWi99].

DEFINITION

D34: Let f be a proper numbering of a graph G. The ***bandsize of*** f, denoted $bs_f(G)$, is the number of distinct edge differences produced by f. The ***bandsize of*** G is given by $bs(G) = \min\{bs_f(G) : f$ is a proper numbering of $G\}$.

FACT

F65: [ErHeWi89] For any graph G, $B(G) \geq bs(G)$.

Edgesum (Bandwidth Sum)

The edgesum first appeared in [Ha64]. Edgesums are discussed in [Se70, Io74, Io76, ChChDeGi82, Ch88, YaWa95, YuHu95, YuHu96, LaWi99].

DEFINITION

D35: Let f be a proper numbering of G. The ***edgesum generated by*** f is $s_f(G) = \sum_{uv \in E(G)} |f(u) - f(v)|$. The ***edgesum of*** G is given by $s(G) = \min\{s_f(G) : f$ is a proper numbering of $G\}$.

FACTS

F66: For the n-dimensional hypercube Q_n, $s(Q_n) = 2^{n-1}(2^n - 1)$.

F67: Like bandwidth, the edgesum decision problem is NP-complete.

Cyclic Bandwidth

Cyclic bandwidth is discussed in [Li94, Li97, HaKaRi99, LaShCh02].

DEFINITION

D36: Let f be a proper numbering of G. The ***cyclic bandwidth of*** f is $B_{cf}(G) = \max\{||f(u) - f(v)||_c : uv \in E\}$ where$||x||_c = \min\{|x|, n - |x|\}$. The ***cyclic bandwidth*** of G is given by $B_c(G) = \min\{B_{cf}(G) : f$ is a proper numbering of $G\}$.

Edge-Bandwidth

Edge-bandwidth is introduced and several results are presented in [JiMuShWe99].

DEFINITIONS

D37: An ***edge-numbering*** f of a graph G is a bijection from $E(G)$ to the set of integers.

D38: Let f be an edge-numbering of G. The ***edge-bandwidth of*** f is $B'_f(G) = \max\{|f(e_1) - f(e_2)| : \text{edges } e_1 \text{ and } e_2 \text{ adjacent in } G\}$. The ***edge-bandwidth of graph*** G is given by $B'(G) = \min\{B'_f(G) : f \text{ an edge numbering of } G\}$.

D39: The ***line graph*** of a graph G is the graph $L(G)$ such that $V(L(G)) = E(G)$ and two vertices in $L(G)$ are adjacent if and only if the corresponding edges are adjacent in G.

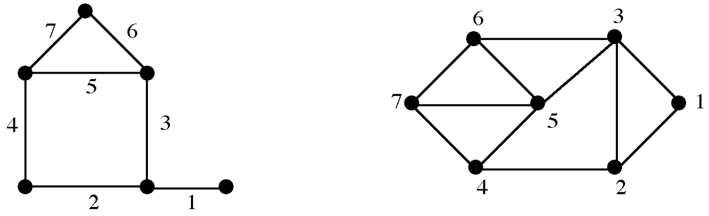
FACTS

F68: For any graph G, $B'(G) = B(L(G))$.

F69: For any graph G, $B(G) \leq B'(G)$ and, if G is a forest, $B'(G) \leq 2B(G)$.

EXAMPLE

E8: Figure 9.4.10 shows an edge-bandwidth numbering of a graph G and a bandwidth numbering of the line graph $L(G)$. Notice that the edge-bandwidth numbering of an edge of G is identical to the bandwidth numbering of the corresponding vertex in $L(G)$. It is not difficult to see that $B(L(G)) = 3$.



G, B'(G) = 3

L(G), B(L(G)) = 3

Figure 9.4.10 $B(L(G)) = B'(G) = 3$.

Profile

The *profile* of a graph is discussed in [LiYu94, LaWi99].

DEFINITIONS

D40: Let f be a proper numbering of a graph G, and let vertex $v \in V(G)$. The ***profile width*** is $w_f(v) = \max_{x \in N[v]}(f(v) - f(x))$ where $N[v]$ is the closed neighborhood of v.

D41: Let f be a proper numbering of a graph G. The ***profile of*** f is $P_f(G) = \sum_{v \in V} w_f(v)$. The ***profile of*** G, denoted $P(G)$, is given by $P(G) = \min\{P_f(G) : f \text{ is a proper numbering of } G\}$.

FACTS

F70: $P(P_n) = n - 1$.

F71: $P(C_n) = 2n - 3$.

F72: If $m \leq n$, then $P(K_{m,n}) = mn + m(m-1)/2$.

Cutwidth

References for cutwidth include [GuSu82, Le82, Ch85, MaPaSu85, Ya85, Ch88].

DEFINITION

D42: Let f be a proper numbering of a graph G. The ***cutwidth of*** f is $c_f(G) = \max_i |\{vw \in E(G) : f(v) \leq i < f(w)\}|$. The ***cutwidth of*** G is $cutw(G) = \min\{c_f(G) : f$ is a proper numbering of $G\}$.

FACTS

F73: $cutw(P_n) = 1$.

F74: $cutw(C_n) = 2$.

F75: $cutw(K_n) = \lfloor n^2/4 \rfloor$.

F76: $cutw(K_{1,n}) = \lfloor n/2 \rfloor$.

F77: $cutw(T_{k,d}) = \lceil (d-1)(kt-1)/2 \rceil + 1$ if $d \geq 3$.

Topological Bandwidth

Some references for topological bandwidth are [Ch80-a, MaPaSu85, Ch88].

DEFINITION

D43: The ***topological bandwidth*** of graph G, denoted $B^*(G)$, is given by $B^*(G) = \min\{B(G') : G'$ is a refinement of $G\}$.

FACTS

F78: For any graph G, $B^*(G) \leq cutw(G)$.

F79: For any tree T, $B^*(T) \leq cutw(T) \leq B^*(T) + \log_2 B^*(T) + 2$.

F80: If G is P_n, C_n, or $K_{1,n}$, then $B^*(G) = cutw(G)$.

F81: $B^*(K_n) = n - 1 < cutw(K_n)$.

Additive Bandwidth

The proper numbering of Q_3 given in Section 9.4.1 corresponds to an adjacency matrix with ones on all but one diagonal above the main diagonal, including the diagonal farthest away. Since $B(Q_3) = 4$, this is far from a bandwidth numbering. However, consider the main *contradiagonal* (running from the lower left corner of the adjacency matrix to the upper right). All ones are on the main contradiagonal and the two contradiagonals above and below it. This concept, recognized and elucidated in [BaRuSl92], is the basis for a second type of bandwidth called *additive bandwidth*. Since the ones of the adjacency matrix are all on contradiagonals within two of the main contradiagonal, the additive bandwidth for Q_3 is at most two, and it is easy to see that equality holds.

DEFINITION

D44: Let f be a proper numbering of a graph G. The ***additive bandwidth of*** f is $B_f^+(G) = max\{|f(u) + f(v) - (n+1)| : uv \in E\}$. The ***additive bandwidth of*** G, denoted $B^+(G)$, is given by $B^+(G) = min\{B_f^+(G) : f$ is a proper numbering of $G\}$.

REMARKS

R8: The expression $|f(u) + f(v) - (n+1)|$ indicates, in the adjacency matrix, the number of contradiagonals (perhaps zero) from the main contradiagonal which contains the one corresponding to edge uv, and the summation involved motivated the name "additive bandwidth."

R9: Many of the investigations which have been made into bandwidth have been repeated for this new concept. However, while it is believed that the corresponding decision problem is NP-complete, this had not been proven nor have any algorithms yet been developed.

R10: Since $B^+(Q_3) = 2 < B(Q_3) = 4$, we see that additive bandwidth can be smaller than bandwidth. The graphs $P_m \times P_n$ represent an infinite family for which this is true by the same factor of two. In fact, this factor of two is best possible. On the other hand, additive bandwidth can be arbitrarily larger than bandwidth (by Fact 34 and Fact 82 below).

R11: In addition to the three results given below, the values of $B^+(G)$ for other families of graphs have been determined, as have several relationships between it and other invariants. The complete k-ary tree has proven difficult, and only partial results are available for it. We have seen that adding an edge to a graph can double its original bandwidth. In fact, the addition of an edge can triple the original additive bandwidth [BrCaViWiYe]. Additional references for additive bandwidth are [Ha93, HaCaCa94, VoBr94, BrCaRoVi96, Ca96, DuBr97].

FACTS

F82: [BaRuSl92] $B^+(P_m \times P_n) = \lceil \min\{m, n\}/2 \rceil$.

F83: [BaRuSl92] If $B^+(G) \geq 1$, then $B(G) \leq 2B^+(G)$.

F84: [BaRuBrCaSlVi95] $B(tK_3) = 2$ and $B^+(tK_3) = t$.

References

[AlLiMcEr92] Y. Alavi, J. Liu, J. McCanna, and P. Erdös, On the minimum size of graphs with a given bandwidth, *Bulletin of the Institute of Combinatorics and its Applications* 6 (1992), 22–32.

[AlMa65] G. G. Alway and D. W. Martin, An algorithm for reducing the bandwidth of a matrix of symmetrical configuration, *The Computer Journal* 8 (1965), 264–272.

[AnKaGe96] K. Ando, A. Kaneko, and S. Gervacio, The bandwidth of a tree with k leaves is at most $\lceil k/2 \rceil$, *Discrete Mathematics* 150 (1996), 403–406.

[BaRuBrCaSlVi95] M. E. Bascuñán, S. Ruiz, R. C. Brigham, R. M. Caron, P. J. Slater, and R. P. Vitray, On the additive bandwidth of graphs, *Journal of Combinatorial Mathematics and Combinatorial Computing* 18 (1995), 129–144.

[BaRuSl92] M. E. Bascuñán, S. Ruiz, and P. J. Slater, The additive bandwidth of grids and complete bipartite graphs, *Congressus Numerantium* 88 (1992), 245–254.

[BlKoRaVe00] A. Blum, G. Konjevod, R. Ravi, and S. Vempala, Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems, *Theoretical Computer Science* 235 (2000), 25–42.

[BrCaDuFiVi00] R. C. Brigham, J. R. Carrington, R. D. Dutton, J. Fiedler, and R. P. Vitray, An extremal bandwidth problem for bipartite graphs, *Journal of Graph Theory* 35 (2000), 278–289.

[BrCaRoVi96] R. C. Brigham, J. R. Carrington, D. G. Rogers, and R. P. Vitray, On the additive bandwidth of the complete k-ary tree, *Congressus Numerantium* 118 (1996), 209–214.

[BrCaViWiYe] R. C. Brigham, J. R. Carrington, R. P. Vitray, D. J. Williams, and J. Yellen, Change in additive bandwidth when an edge is added, to appear in *Ars Combinatoria.*

[BrDu85] R. C. Brigham and R. D. Dutton, A compilation of relations between graph invariants, *Networks* 15 (1985), 73–107.

[BrDu91] R. C. Brigham and R. D. Dutton, A compilation of relations between graph invariants–supplement I, *Networks* 21 (1991), 421–455.

[Ca96] R. M. Caron, Free additive bandwidth of a graph, *Congressus Numerantium* 121 (1996), 49–57.

[CaMaPr02] A. Caprara, F. Malucelli, and D. Pretolani, On bandwidth-2 graphs, *Discrete Applied Mathematics* 117 (2002), 1–13.

[Ch70] V. Chvátal, A remark on a problem of Harary, *Czechoslovak Mathematical Journal* 20 (1970), 109–111.

[Ch75] J. Chvátalová, Optimal labeling of a product of two paths, *Discrete Mathematics* 11 (1975), 249–253.

[Ch80-a] J. Chvátalová, On the bandwidth problem for graphs, Ph.D. dissertation, Department of Combinatorics and Optimization, University of Waterloo, Ontario (1980).

[Ch80-b] P. Z. Chinn, The bandwidth of the corona and composition of two graphs, Department of Mathematics, Humboldt State University, Arcata, California (1980).

[Ch85] F. R. K. Chung, On the cutwidth and the topological bandwidth of a tree, *SIAM Journal on Algebraic and Discrete Methods* 6 (1985), 268–277.

[Ch88] F. R. K. Chung, Labelings of graphs, *Selected Topics in Graph Theory* 3, Academic Press Limited, San Diego, CA (1988), 151–168.

[ChChDeGi82] P. Z. Chinn, J. Chvátalová, A. K. Dewdney, and N. E. Gibbs, The bandwidth problem for graphs and matrices–a survey, *Journal of Graph Theory* 6 (1982), 223–254.

[ChDeGiKo75] J. Chvátalová, A. K. Dewdney, N. E. Gibbs, and R. R. Korfhage, The bandwidth problem for graphs: a collection of recent results, Research Report 24, Department of Computer Science, University of Western Ontario, London, Ontario (1975).

[ChErChGr81] P. Z. Chinn, P. Erdös, F. R. K. Chung, and R. L. Graham, On the bandwidth of a graph and its complement, *The Theory and Applications of Graphs*, G. Chartrand, Ed., Wiley, New York (1981), 243–253.

[ChOp86] J. Chvátalová and J. Opatrný, The bandwidth problem and operations on graphs, *Discrete Mathematics* 61 (1986), 141–150.

[ChSe89] F. R. K. Chung and P. D. Seymour, Graphs with small bandwidth and cutwidth, *Discrete Mathematics* 75 (1989), 113–119.

[ChTr84] F. R. K. Chung and W. T Trotter, Jr., Triangle-free graphs with restricted bandwidth, *Progress in Graph Theory*, Academic Press, Toronto, Ontario (1984), 175–190.

[CuMc69] E. Cuthill and J. McKee, Reducing the bandwidth of sparse symmetric matrices, *Proc. 24th National Conference of the ACM* (1969), 157–172.

[De76] A. K. Dewdney, The bandwidth of a graph–some recent results, *Congressus Numerantium* 17 (1976), 273–288.

[DuBr89] R. D. Dutton and R. C. Brigham, On the size of graphs of a given bandwidth, *Discrete Mathematics* 76 (1989), 191–195.

[DuBr97] R. D. Dutton and R. C. Brigham, Invariant relations involving the additive bandwidth, *Journal of Combinatorial Mathematics and Combinatorial Computing* 23 (1997), 77–85.

[Ei79] P. G. Eitner, The bandwidth of the complete multipartite graph, presented at the *Toledo Symposium on Applications of Graph Theory* (1979).

[ErHeWi89] P. Erdös, P Hell, and P. Winkler, Bandwidth versus bandsize, *Annals of Discrete Mathematics* 41 (1989), 117–129.

[Ev79] G. C. Everstine, A comparison of three resequencing algorithms for the reduction of matrix profile and wavefront, *International Journal for Numerical Methods in Engineering* 14 (1979), 837–853.

[Fe00] U. Feige, Approximating the bandwidth via volume respecting embeddings, *Journal of Computer and System Sciences* 60 (2000), 510–539.

[FuWe01] Z. Füredi and D. B. West, Ramsey theory and bandwidth of graphs, *Graphs and Combinatorics* 17 (2001), 463–471.

[GaGrJoKn78] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth, Complexity results for bandwidth minimization, *SIAM Journal on Applied Mathematics* 34 (1978), 477–495.

[GiPoSt76] N. E. Gibbs, W. G. Poole, Jr., and P. K. Stockmeyer, An algorithm for reducing the bandwidth and profile of a sparse matrix, *SIAM Journal on Numerical Analysis* 13 (1976), 236–250.

[GoOp90] C. GowriSankaran and J. Opatrný, New bandwidth reduction algorithms, *Congressus Numerantium* 76 (1990), 77–88.

[Gu01] A. Gupta, Improved bandwidth approximation for trees and chordal graphs, *Journal of Algorithms* 40 (2001), 24–36.

[GuSu82] E. M. Gurari and I. H. Sudborough, Improved dynamic programming algorithms for the bandwidth minimization problem and the min cut linear arrangement problem, Technical Report, Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL (1982).

[Ha64] L. H. Harper, Optimal assignment of numbers to vertices, *Journal of SIAM* 12 (1964), 131–135.

[Ha66] L. H. Harper, Optimal numberings and isoperimetric problems on graphs, *Journal of Combinatorial Theory* 1 (1966), 385–393.

[Ha93] F. W. Hackett, The additive bandwidth of a union of star graphs, Master's report, Department of Mathematics, University of Central Florida (1993).

[HaCaCa94] F. W. Hackett, R. M. Caron, and J. R. Carrington, The additive bandwidth of a union of stars, *Congressus Numerantium* 101 (1994), 155–160.

[HaKaRi99] F. Harary, P. C. Kainen, and A. Riskin, Every graph of cyclic bandwidth 3 is toroidal, *Bulletin of the Institute of Combinatorics and its Applications* 27 (1999), 81–84.

[HaMa97] J. Haralambides and F. Makedon, Approximation algorithms for the bandwidth minimization problem for a large class of trees, *Theory of Computing Systems* 30 (1997), 67–90.

[HeGr79] G. Hein and E. Groten, On the use of bandwidth and profile reduction in combination solutions of satellite altimetry, *Acta Geodaetia, Geophys Montanistica Acad. Sci. Hung.* 14 (1979), 59–69.

[Io74] M. A. Iordanskiĭ, Minimal numerations of the vertices of trees, *Soviet Mathematics Doklady* 15 (1974), 1311–1315.

[Io76] M. A. Iordanskiĭ, Minimal numerations of the vertices of trees, *Problemy Kibernetiki* 31 (1976), 109–132.

[JiMuShWe99] T. Jiang, D. Mubayi, A. Shastri, and D. B. West, Edge-bandwidth of graphs, *SIAM Journal on Discrete Mathematics* 12 (1999), 307–316.

[KaSh96] H. Kaplan and R. Shamir, Pathwidth, bandwidth, and completion problems to proper integral graphs with small cliques, *SIAM Journal on Computing* 25 (1996), 540–561.

[KlTa01] T. Kloks and R. B. Tan, Bandwidth and topological bandwidth of graphs with few P_4's, *Discrete Applied Mathematics* 115 (1997), 117–133.

[KrSt02] D. Kratsch and L. Stewart, Approximating bandwidth by mixing layouts of interval graphs, *SIAM Journal on Discrete Mathematics* 15 (2002), 435–449.

[LaShCh02] P. C. B. Lam, W. C. Shiu, and W. H. Chan, Characterization of graphs with equal bandwidth and cyclic bandwidth, *Discrete Mathematics* 242 (2002), 283–289.

[LaWi95] Y. Lai and K. Williams, Bandwidth of the strong product of paths and cycles, *Congressus Numerantium* 109 (1995), 123–128.

[LaWi97-a] Y. Lai and K. Williams, Some bounds on bandwidth, edgesum, and profile of graphs, *Congressus Numerantium* 125 (1997), 25–31.

[LaWi97-b] Y. Lai and K. Williams, On bandwidth for the tensor product of paths and cycles, *Discrete Applied Mathematics* 73 (1997), 133–141.

[LaWi99] Y. Lai and K. Williams, A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs, *Journal of Graph Theory* 31 (1999), 75–94.

[Le82] T. Lengauer, Upper and lower bounds on the complexity of the min-cut linear arrangement problem on trees, *SIAM Journal on Algebraic and Discrete Methods* 3 (1982), 99–113.

[Li94] Y. Lin, The cyclic bandwidth problem, *Systems Science and Mathematical Sciences* 7 (1994), 282–288.

[Li97] Y. Lin, Minimum bandwidth problem for embedding graphs in cycles, *Networks* 29 (1997), 135–140.

[Li00] Y. Lin, On characterizations of graph bandwidth, *OR Transactions* 4 (2000), 1–6.

[LiWiWa91] J. Liu, K. Williams, and J. F. Wang, Bandwidth for the sum of two graphs, *Congressus Numerantium* 82 (1991), 79–85.

[LiYu94] Y. X. Lin and J. J. Yuan, Minimum profile of grid networks, *Systems Science and Mathematical Sciences* 7 (1994), 56–66.

[MaPaSu85] F. S. Makedon, C. H. Papadimitriou, and I. H. Sudborough, Topological bandwidth, *SIAM Journal on Algebraic and Discrete Methods* 6 (1985), 418–444.

[Mi91] Z. Miller, Graph layouts, *Applications of Discrete Mathematics*, J. G. Michaels and K. H. Rosen (Editors), McGraw-Hill, New York (1991), 365–393.

[Pa76] C. H. Papadimitriou, The NP-completeness of the bandwidth minimization problem, *Computing* 16 (1976), 263–270.

[Ro68] R. Rosen, Matrix bandwidth minimization, *Proc. 23rd National Conference of the ACM*, Brandon Systems, Princeton, New Jersey (1968), 585–595.

[Sa80] J. B. Saxe, Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time, *SIAM Journal on Algebraic and Discrete Methods* 1 (1980), 363–369.

[Se70] M. A. Seĭdvasser, The optimal numbering of the vertices of a tree, *Diskretnyĭ Analiz* 17 (1970), 56–74.

[Sm85] W. F Smyth, Algorithms for the reduction of matrix bandwidth and profile, *Journal of Computational and Applied Mathematics* 12-13 (1985), 551–561.

[Sm95] L. Smithline, Bandwidth of the complete k-ary tree, *Discrete Mathematics* 142 (1995), 203–212.

[VoBr94] M. P. Vogt and R. C. Brigham, On the additive bandwidth of simple trees, *Congressus Numerantium* 103 (1994), 155–160.

[WaWeYa95] J. F. Wang, D. B. West, and B. Yao, Maximum bandwidth under edge addition, *Journal of Graph Theory* 20 (1995), 87–90.

[WaYa95] J. F. Wang and B. Yao, On upper bounds of bandwidths of trees, *Acta Mathematicae Applicatae Sinica (English Series)* 11 (1995), 152–159.

[Ya85] M. Yannakakis, A polynomial algorithm for the min-cut linear arrangement of trees, *Journal of the Association for Computing Machinery* 32 (1985), 950–988.

[Ya98] J. Yan, Algorithm aspects of the bandwidth problem on P_4-sparse graphs, *Tamsui Oxford Journal of Mathematical Sciences* 14 (1998), 11–18.

[YaWa95] B. Yao and J. F. Wang, On bandwidth sums of graphs, *Acta Mathematicae Applicatae Sinica (English Series)* 11 (1995), 69–78.

[YuHu95] J. Yuan and Q. Huang, A note on the bandwidth sum of complete multipartite graphs, *Journal of Mathematical Study* 28 (1995), 19–22.

[YuHu96] J. Yuan and Q. Huang, Some lower bounds of bandwidth sum of graphs with applications, *Mathematica Applicata* 9 (1996), 536–538.

GLOSSARY FOR CHAPTER 9

additive bandwidth $B^+(G)$ – of graph G:

$$B^+(G) = min\left\{B_f^+(G) \mid f\text{is proper numbering of } G\right\}$$

additive bandwidth $B_f^+(G)$ of proper numbering f – of graph G:

$$B_f^+(G) = max\left\{|f(u) + f(v) - (n+1)| \;\middle|\; uv \in E(G)\right\}$$

almost all graphs – property P holds for: if the ratio of the number of n-vertex graphs possessing P divided by the number of n-vertex graphs approaches one as n approaches infinity.

archimedean function: a function ϕ such that for every $c > 0$, $\lim_{x\to\infty} \phi(x,c) = \infty$ and $\lim_{x\to\infty} \phi(c,x) = \infty$.

asteroidal triple: three vertices in a graph such that, for any two of them, there is a path containing those two but no neighor of the third.

automorphism group: the set of all automorphisms of a graph under the operation of composition.

automorphism: an isomorphism between a graph and itself.

bandsize$_1$ – of proper numbering f in a graph G: the number of distinct edge differences produced by f; denoted $bs_f(G)$.

bandsize$_2$ – of graph G: $bs(G) = \min\{bs_f(G) : f$ is proper numbering of $G\}$.

bandwidth$_1$ $B_f(G)$ – of proper numbering f in a graph G:

$$B_f(G) = max\left\{|f(u) - f(v)| \;\middle|\; uv \in E(G)\right\}$$

bandwidth$_2$ $B(G)$ – of graph G: $B(G) = min\{B_f(G) : f$ is aproper numbering of $G\}$.

bandwidth decision problem: the problem which has answer "YES" if $B(G) \le K$ and "NO" otherwise when presented with given graph G and positive integer K.

bandwidth numbering – of graph G: proper numbering f such that $B(G) = B_f(G)$.

bandwidth-k decision problem: for fixed integer k, the problem which has answer "YES" if $B(G) \le k$ and "NO" otherwise when presented with given graph G.

cartesian product $G \times H$ – of graphs G and H: a graph where $V(G \times H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G \times H)$ if and only if either (i) $g_1 = g_2$ and $h_1h_2 \in E(H)$ or (ii) $h_1 = h_2$ and $g_1g_2 \in E(G)$.

center – of a connected graph: the subgraph induced by the central vertices.

___, k-: the subgraph induced by the k-central vertices.

central vertex – in a connected graph: a vertex whose eccentricity equals the radius of the graph.

___, k-: a vertex whose k-eccentricity is the k-radius.

chordal graph: a graph containing no induced cycle of length greater than three.

chromatic number $\chi(G)$ – of graph G: the smallest number k such that there is a function $f : V(G) \to \{1, 2, \ldots, k\}$ with the property that, if uv is an edge, then $f(u) \ne f(v)$.

closed interval$_1$ – between two vertices: a set consisting of these two vertices and all vertices lying in some geodesic between them.

closed interval$_2$ – of a set: the union of all closed intervals between every pair of vertices in the set.

closed neighborhood$_1$ – of a vertex v: $N(v) \cup \{v\}$, where $N(v)$ is the open neighborhood of v; denoted $N[v]$.

closed neighborhood$_2$ – of a set S: $\cup_{v \in S} N[v]$, where $N[v]$ is the closed neighborhood of v; denoted $N[S]$.

competition graph: an intersection graph of the family of outsets of the vertices in some digraph.

___, p-: a p-intersection graph of the family of outsets of the vertices of some digraph.

___, **ϕ-tolerance**: a graph that is the ϕ-tolerance intersection graph of the family of out-sets of the vertices of some digraph.

complement – of graph $G = (V, E)$: a graph $\overline{G}$ with $V(\overline{G}) = V$ and vertices u and v are adjacent in $\overline{G}$ if and only if they are not adjacent in G; also called *edge-complement*.

complementary numbering f_c – of proper numbering f: $f_c(v) = n + 1 - f(v)$ for each vertex v of G.

complete k-ary tree $T_{k,d}$ of depth d: a rooted tree in which all vertices at level $d-1$ or less have exactly k children, and all vertices at level d are leaves.

complete k-partite graph $K_{n_1,n_2,\ldots,n_k}$: a graph whose vertex-set is partitioned into sets A_i of n_i vertices, $1 \leq i \leq k$, with two vertices adjacent if and only if they are in distinct sets.

composition $G(H)$ – of graphs G and H: a graph where $V(G(H)) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(H))$ if and only if either (i) $g_1 g_2 \in E(G)$ or (ii) $g_1 = g_2$ and $h_1 h_2 \in E(H)$.

contraction $G|_e$ – of graph G along edge e: a merger where $e = uv$.

convex hull – of a set: the smallest convex set containing a given set.

convex set: a set of vertices in a graph whose closed interval is itself.

corona – of two graphs G_1 and G_2: the graph formed from one copy of G_1 and $|V(G_1)|$ copies of G_2 where the ith vertex of G_1 is adjacent to every vertex in the ith copy of G_2.

cutwidth$_1$ – of proper numbering f in a graph G: $c_f(G) = \max_i |\{vw \in E(G) : f(v) \leq i < f(w)\}|$.

cutwidth$_2$ – of a graph G: $cutw(G) = \min\{c_f(G) : f$ is a proper numbering of $G\}$.

cyclic bandwidth$_1$ $B_{cf}(G)$ – of proper numbering f in a graph G: $B_{cf}(G) = \max\{||f(u) - f(v)||_c : uv \in E\}$, where $||x||_c = \min\{|x|, n - |x|\}$.

cyclic bandwidth$_2$ $B_c(G)$ – of graph G: $B_c(G) = \min\{B_{cf}(G) : f$ is proper numbering of $G\}$.

daisy: a graph that can be constructed from $k \geq 2$ disjoint cycles by identifying a set of k vertices, one from each cycle, into one vertex.

decomposable graph: a graph G whose vertex-set can be partitioned into $\gamma(G)$ subsets $C_1, C_2, \ldots, C_{\gamma(G)}$ such that each of the induced subgraphs $G[C_i]$ is a complete subgraph of G.

degree sequence – of graph G: a listing of the degrees of the n vertices of G, usually in monotonic order.

diameter – of graph G: the maximum distance between any two vertices of G; denoted $diam(G)$.

___, k-: the maximum k-eccentricity among the vertices in a graph.

directed distance from u **to** v: the length of a shortest directed $u-v$ path.

distance$_1$ – between two vertices: the length of a shortest path between these two vertices.

distance$_2$ – between two subgraphs: the minimum distance between a vertex in one subgraph and a vertex in the other subgraph.

dominating function – of a graph G: a function $f : V \to \{0,1\}$ such that for each $v \in V$, $\sum_{u \in N[v]} f(u) \geq 1$.

dominating set – of a graph G: a set $S \subset V$ such that every vertex in V is either in S or adjacent to a vertex in S.

___, **connected** – of a graph G: a dominating set S of G such that $G[S]$ is connected.

___, **clique** – of a graph G: a dominating set S of G such that $G[S]$ is a clique.

___, **cycle** – of a graph G: a dominating set S of G such that $G[S]$ has a hamiltonian cycle.

___, **minimal**: a dominating set that contains no dominating set as a proper subset.

___, **paired** – of a graph G: a dominating set S of G such that the induced subgraph $G[S]$ has a perfect matching.

___, **total** – of a graph G: a set $S \subseteq V$ such that every vertex of V is adjacent to a vertex of S.

domination critical graph: a graph G with the property that adding an arbitrary edge not in G results in a graph with domination number less than the domination number of G.

___, k-γ: a domination critical graph with domination number k.

domination number – of a graph G: the minimum cardinality of a dominating set of G.

___, **clique** – of a graph G: the minimum cardinality of a clique dominating set of G.

___, **connected** – of a graph G: the minimum cardinality of a connected dominating set of G.

___, **cycle** – of a graph G: the minimum cardinality of a cycle dominating set of G.

___, **independent** – of a graph G: the minimum cardinality of a maximal independent set. (A maximal independent set must be a dominating set.)

___, **paired** – of a graph G: the minimum cardinality of a paired dominating set of G.

___, **total** – of a graph G: the minimum cardinality of a total dominating set of G.

___, **upper** – of a graph G: the maximum cardinality of a minimal dominating set of G.

eccentricity – of a vertex v: the distance from vertex v to a vertex farthest from it.

___, k-: the maximum Steiner distance among all k-element sets of vertices containing the vertex.

edge-bandwidth$_1$ – of edge-numbering f of graph G: $B'_f(G) = \max\{|f(e_1) - f(e_2)| :$ edges e_1 and e_2 adjacent in $G\}$.

edge-bandwidth$_2$ – of graph G: $B'(G) = \min\{B'_f(G) : f$ an edge numbering of $G\}$.

edge chromatic number $\chi_1(G)$ – of graph G: the smallest number k such that there is a function $f : E(G) \to \{1, 2, \ldots, k\}$ with the property that, if edges e_1 and e_2 share a common vertex, then $f(e_1) \neq f(e_2)$.

edge clique cover – of a graph G: a family $\mathcal{E} = \{C_1, \ldots, C_k\}$ of complete subgraphs of G such that every edge of G is in at least one of $E(C_1), \ldots, E(C_k)$. (Elsewhere, a clique is required to be a maximal complete subgraph.)

___, *p*- – of a graph G: a family $\{V_1, \ldots, V_m\}$ of not necessarily distinct subsets of $V(G)$ such that, for every set $\{i_1, \ldots, i_p\}$ of p distinct subscripts, $T = V_{i_1} \cap \cdots \cap V_{i_p}$ induces a complete subgraph of G, and such that the collection of sets of the form T is an edge clique cover of G.

___, **ϕ-T-edge** – of the graph G: a family $\{V_1, \ldots, V_n\}$ such that $v_i v_j \in E(G)$ if and only if at least $\phi(t_i, t_j)$ of the sets V_k contain both v_i and v_j.

edge cover – of a graph G: a set of edges covering all the vertices of G.

edge-numbering f – of graph G: a bijection $f : E(G) \to \{1, 2, \ldots, |E(G)|\}$.

edgesum$_1$ $s_f(G)$ – generated by proper numbering f of graph G:

$$s_f(G) = \sum_{uv \in E(G)} |f(u) - f(v)|$$

edgesum$_2$ $s(G)$ – of graph G: $s(G) = \min\{s_f(G) : f$ is a proper numbering of $G\}$.

geodesic: a shortest path between two vertices.

geodetic number: the minimum cardinality of a geodetic set in a graph.

geodetic set: a set of vertices of a graph whose closed interval is the vertex-set of the graph.

girth – of graph G: the size of a smallest induced cycle of G; denoted $girth(G)$.

hull number – of graph G: the minimum cardinality of a hull set in G

hull set: a set of vertices of a graph whose convex hull is the vertex-set of the graph.

hypercube – n-dimensional: the graph having 2^n vertices, each labeled with a distinct n-digit binary sequence, and two vertices adjacent if and only if their labels differ in exactly one position.

independent set – of vertices: a set of vertices in which no two vertices are adjacent.

intersection graph – of a family of sets $\mathcal{F} = \{S_1, \ldots, S_n\}$: the graph having $\mathcal{F}$ as vertex-set with S_i adjacent to S_j if and only if $i \neq j$ and $S_i \cap S_j \neq \emptyset$.

___, *p*-: a ϕ-intersection graph where ϕ is the constant function equal to p.

___, **ϕ-tolerance** – of the family $\mathcal{F} = \{S_1, \ldots, S_n\}$: the graph with vertices $\mathcal{F}$ and $S_i S_j$ an edge if and only if $i \neq j$ and $\mu(S_i \cap S_j) \geq \phi(t_i, t_j)$, where μ is a real-valued function on the subsets of S with $\mu(S_i) = t_i$, and ϕ is a non-negative, real-valued, symmetric function defined on pairs of positive real numbers.

intersection number – of a graph G: the size of a smallest set S for which G is the intersection graph of subsets of S.

___, *p*- – of a graph: the size of a smallest set S for which G is the p-intersection graph of subsets of S.

interval graph: an intersection graph of a family of intervals of the real line.

___, **Archimedean ϕ-tolerance**: a graph that is a ϕ-tolerance interval graph for every Archimedean function ϕ.

___, **ϕ-tolerance**: a ϕ-tolerance intersection graph of a family of intervals of the real line.

___, **max-tolerance**: a ϕ-tolerance interval graph where $\phi(t_i, t_j) = \max(t_i, t_j)$.

___, **min-tolerance**: a ϕ-tolerance interval graph where $\phi(t_i, t_j) = \min(t_i, t_j)$.

___, **proper**: an interval graph using intervals where no interval is properly contained in another.

___, **sum-tolerance**: a ϕ-tolerance interval graph where $\phi(t_i, t_j) = t_i + t_j$.

___, **unit**: an interval graph using intervals all of the same length.

irredundance number – of a graph G: the minimum cardinality of a maximal irredundant set of G.

___, **upper** – of a graph G: the maximum cardinality of an irredundant set of G.

irredundant set – of vertices: a set S of vertices such that for every vertex $v \in S$, $N[v] - N[S - \{v\}] \neq \emptyset$.

key: the graph obtained by joining with an edge a vertex in C_m to an end-vertex of P_n; denoted $L_{m,n}$.

line graph $L(G)$ – of graph G: a graph with vertex-set equal to the edges of G and two vertices are adjacent if and only if the corresponding edges are adjacent in G.

link edge: an edge joining two link vertices.

link vertex$_1$ – in a type (a) unit: two non-adjacent vertices in that unit.

link vertex$_2$ – in a type (b) unit: the vertex of degree 1 in that unit.

$\frac{m}{3}$-graph: a graph with m edges that satisfies the three conditions: (i) $\delta(G) \geq 2$, (ii) G is connected, and (iii) $\gamma(G) > m/3$.

median vertex: a vertex whose total distance is minimum among the vertices in a graph.

median subgraph: the subgraph induced by the median vertices.

merger $G|_{u,v}$ – of vertices u and v of graph G: a graph obtained from G by identifying u and v and then eliminating any loops and duplicate edges.

$\frac{2}{5}$-minimal graph: an n-vertex graph that is edge-minimal with respect to satisfying the three conditions: (i) $\delta(G) \geq 2$, (ii) G is connected, and (iii) $\gamma(G) \geq 2n/5$.

minimum hull subgraph: an induced subgraph whose vertex set is a hull set of minimum cardinality for some graph.

NP-complete – problem: a problem having a "YES" or "NO" answer that can be solved nondeterministically in polynomial time, and all other such problems can be transformed into it in a polynomial time.

open neighborhood$_1$ – of a vertex v: the set of vertices that are adjacent to v; denoted $N(v)$.

open neighborhood$_2$ – of a set S: $\cup_{v \in S} N(v)$, where $N(v)$ is the open neighborhood of vertex v; denoted $N(S)$.

out-set – of a vertex v in a digraph: the set of vertices x such that (v, x) is an arc in the digraph.

packing number – of a graph G: the maximum cardinality of a packing in G.

packing – in a graph G: a set S of vertices such that each pair of vertices in S are at a distance at least 3 apart in G.

perfect elimination ordering – of a graph G: an ordering $\langle v_1, \ldots, v_n \rangle$ of all the vertices of G such that, for each $i \in \{1, \ldots, n\}$, v_i is a simplicial vertex of the subgraph induced on the vertex subset $\{v_i, v_{i+1}, \ldots, v_n\}$.

peripheral vertex: a vertex in a connected graph whose eccentricity equals the diameter of the graph.

periphery: the subgraph induced by the peripheral vertices.

polynomial algorithm: an algorithm whose execution time is bounded by a polynomial in some parameter of the problem, often the number of vertices for graph problems.

power – k^{th} of graph G: the graph having the same vertex-set as G and an edge between two vertices if and only if the distance between them is at most k in G.

profile$_1$ $P_f(G)$ – of proper numbering f in a graph G: $P_f(G) = \sum_{v \in V} w_f(v)$.

profile$_2$ $P(G)$ – of graph G: $P(G) = \min\{P_f(G) : f \text{ is a proper numbering of } G\}$.

profile width $w_f(v)$ – for vertex $v \in V(G)$ and proper numbering f of graph G: $w_f(v) = \max_{x \in N[v]}(f(v) - f(x))$ where $N[v]$ is the closed neighborhood of v.

proper numbering – of graph G: a bijection $f : V(G) \to \{1, 2, \ldots, |V(G)|\}$.

radius $rad(G)$ – of graph G: the smallest number r such that there is a vertex u of G with distance at most r from every other vertex of G; equivalently, the minimum eccentricity among the vertices of a connected graph.

___, k-: the minimum k-eccentricity among the vertices in a graph.

refinement – of graph G: a graph obtained from G by a finite number of subdivisions.

self-centered graph: a graph whose center is itself.

simplicial vertex: a vertex whose neighbors induce a complete subgraph.

Steiner distance – of a set of vertices in a graph: the minimum size of a connected subgraph containing the set.

strong diameter – of a strong digraph: the maximum strong eccentricity among the vertices of the strong digraph.

strong eccentricity – of a vertex in a strong digraph: the greatest strong distance from the vertex to a vertex in the strong digraph.

strong radius – of a strong digraph: the minimum strong eccentricity among the vertices of the strong digraph.

strong (Steiner) distance – between two vertices in a strong digraph: the minimum size of a strong subdigraph containing these two vertices.

strong center – of a strong digraph: the subdigraph induced by the strong central vertices.

strong central vertex – in a strong digraph: a vertex whose strong eccentricity is the strong radius.

strong digraph: synonym for strongly connected digraph.

strong product $G(Sp)H$ – of graphs G and H: a graph with $V(G(Sp)H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(Sp)H)$ if and only if either $g_1g_2 \in E(G)$ and $h_1h_2 \in E(H)$ or $g_1 = g_2$ and $h_1h_2 \in E(H)$ or $h_1 = h_2$ and $g_1g_2 \in E(G)$.

strongly connected digraph: a digraph containing both a directed $u-v$ path and a directed $v-u$ path for every pair u, v of vertices in the digraph.

strongly self-centered digraph: a strong digraph whose strong center is itself.

subdivision – of edge uv: a graph obtained by replacing uv with path $< u, w, v >$ where w is a new vertex of degree two.

subtree graph: an intersection graph of a familiy of subtrees of a tree.

sum $G + H$ – of graphs G and H: a graph with $V(G+H) = V(G) \cup V(H)$ and $E(G+H) = E(G) \cup E(H) \cup \{uv : u \in V(G) \text{ and } v \in V(H)\}$.

tensor product $G(Tp)H)$ – of graphs G and H: a graph with $V(G(Tp)H) = V(G) \times V(H)$ and $(g_1, h_1)(g_2, h_2) \in E(G(Tp)H)$ if and only if $g_1g_2 \in E(G)$ and $h_1h_2 \in E(H)$.

thickness $thick(G)$ – of graph G: a smallest number of planar subgraphs of G whose union is G.

topological bandwidth $B^*(G)$ – of graph G: $B^*(G) = \min\{B(G') : G'$ is a refinement of $G\}$.

total distance of a vertex: the sum of the distances from the vertex to all other vertices.

transitive orientation – of a graph G: an assignment of directions to the edges such that the resulting binary relation is transitive.

type-(a) **unit**: a graph that is isomorphic to a cycle C_5.

type-(b) **unit**: a graph that is isomorphic to a key $L_{4,1}$.

vertex arboricity $arbor(G)$ – of graph G: the minimum number of subsets into which $V(G)$ can be partitioned such that the vertices of each subset induce an acyclic subgraph.

vertex cover number $\alpha_0(G)$ – of graph G: the smallest cardinality of a set of vertices such that every edge is incident to at least one of the vertices in the set.

vertex independence number $ind(G)$ – of graph G: the largest cardinality of a set of vertices which induces a graph with no edges.

vertex independence number – of a graph G: the maximum cardinality of a maximal independent set of G.

weakly chordal: a graph that does not contain, as an induced subgraph, a cycle C_n or the complement of C_n for any $n \geq 5$.

Chapter 10

GRAPHS IN COMPUTER SCIENCE

10.1 SEARCHING

Harold N. Gabow, University of Colorado

10.2 DYNAMIC GRAPH ALGORITHMS

Camil Demetrescu, University of Rome "La Sapienza", Italy
Irene Finocchi, University of Rome "Tor Vergata", Italy
Giuseppe F. Italiano, University of Rome "Tor Vergata", Italy

10.3 DRAWINGS OF GRAPHS

Giuseppe Liotta, University of Perugia, Italy
Roberto Tamassia, Brown University

10.4 ALGORITHMS ON RECURSIVELY CONSTRUCTED GRAPHS

R. B. Borie, University of Alabama
R. Gary Parker, Georgia Institute of Technology
C. A. Tovey, Georgia Institute of Technology

GLOSSARY

10.1 SEARCHING

Harold N. Gabow, University of Colorado

10.1.1 Breadth-First Search
10.1.2 Depth-First Search
10.1.3 Topological Order
10.1.4 Connectivity Properties
10.1.5 DFS as a Proof Technique
10.1.6 More Graph Properties
10.1.7 Approximation Algorithms
References

Introduction

A ***search*** of a graph is a methodical exploration of all the vertices and edges. It must run in "linear time", i.e., in one pass (or a small number of passes) over the graph. Even with this restriction, a surprisingly large number of fundamental graph properties can be tested and identified.

This section examines the two most important search methods. *Breadth-first search* gives an efficient way to compute distances. *Depth-first search* is useful for checking many basic connectivity properties, for checking planarity, and also for data flow analysis for compilers. A treatment of at least some aspects of both these methods can be found in almost any algorithms text (some recent ones are [BrBr96, CLRS01, GoTa02, HSR98, Se02, We99]).

All the algorithms of this section (except for §10.1.7) run in linear time or very close to it. Since it takes linear time just to read the graph, the algorithms are essentially as efficient as possible (they are "asymptotically optimal").

NOTATION: Throughout this chapter, the number of vertices and edges of a graph $G = (V, E)$ are denoted n and m, respectively. Time bounds for algorithms are given using asymptotic notation, e.g., $O(n)$ denotes a quantity that, for sufficiently large values of n, is at most cn, for some constant c that is independent of n.

CONVENTION: In all algorithms, we assume that the graph G is given as an adjacency list representation. If G is undirected, this means that each vertex has a list of all its neighbors. The list can be sequentially allocated or linked. If G is directed, then each vertex has a list of all its out-neighbors.

10.1.1 Breadth-First Search

The ***breadth-first search*** method (abbr. ***bfs***) finds shortest paths from a given vertex of a graph to other vertices. It generalizes to Dijkstra's algorithm, which allows numerical (nonnegative) edge-lengths. Throughout this section, the given graph G can be directed or undirected.

DEFINITIONS

D1: A ***length function*** on a graph specifies the numerical length of each edge. Each edge is assumed to have length one, unless there is an explicitly declared length function.

D2: The ***distance*** from vertex u to vertex v in a graph, denoted $d(u,v)$, is the length of a shortest path from u to v.

D3: The ***diameter*** of a graph is the maximum value of $d(u,v)$ for $u \neq v$.

D4: A ***shortest-path tree*** T from a vertex s is a tree, rooted at s, that contains all the vertices that are reachable from s. The path in T from s to any vertex x is a shortest path in G, i.e., it has length $d(s,x)$.

EXAMPLES

E1: Figure 10.1.1 gives a shortest path tree from vertex s.

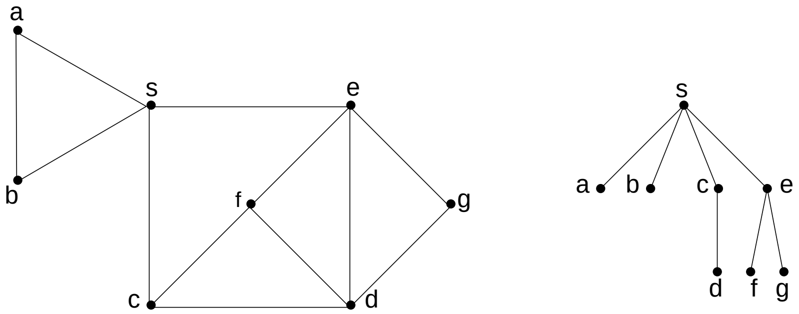


Figure 10.1.1 **Undirected graph and shortest path tree.**

E2: The *small-world phenomenon* [Mi67, Kl00] occurs when relatively sparse graphs have low diameter. Studies have shown that the graphs of movie actors, neural connections in the *c.∼elegans* worm, and the electric power grid of the western United States all exhibit the small-world phenomenon. The world-wide web is believed to have this structure too.

E3: For several decades, mathematicians have computed their *Erdös number* as their distance from the prolific mathematician Paul Erdös, in the graph where an edge joins two mathematicians who have coauthored a paper.

E4: The premise of the *6 Degrees of Kevin Bacon* game is that the graph whose vertices are movie actors and whose edges join two actors appearing in the same movie has diameter at most 6.

E5: In computer and communications networks, a message is typically broadcast from one site s to all others by passing it down a shortest path tree from s.

E6: To solve a puzzle like Sam Lloyd's "15 puzzle" [St96], we can represent each position by a vertex. A directed edge (i,j) exists if we can legally move from i to j. We seek a shortest path from the initial position to a winning position.

Ordered Trees

DEFINITIONS

D5: In a rooted tree, a vertex x is an ***ancestor*** of a vertex y, and y is a ***descendant*** of x, if there is a path from x to y whose edges all go from parent to child. By convention x is an ancestor and descendant of itself (e.g., in the tree of Figure 10.1.1 vertex e has 3 descendants).

D6: Vertex x is a ***proper ancestor (descendant)*** of vertex y if it is an ancestor (descendant) and $x \neq y$.

D7: An ***ordered tree*** is a rooted tree in which the children of each vertex are linearly ordered. In a plane drawing of such a tree, left-to-right order gives the order of the children. (The leftmost child is first.)

D8: Vertex x is ***to the left of*** vertex y if some vertex has children c and d, with c to the left of d, c an ancestor of x and d an ancestor of y.

D9: In a graph G, a ***breadth-first tree*** T from a vertex s contains the vertices that are reachable from s. It is an ordered tree, rooted at s. If x is a vertex at depth δ in the tree T, then the children of x in T are the vertices of G that are adjacent in G to x, but not adjacent (in G) to any vertex in T at depth less than δ, or to any vertex at depth δ in T that is at the left of x.

FACTS

F1: Any breadth-first tree is a shortest-path tree.

F2: A high level bfs algorithm is given below as Algorithm 10.1.1. It constructs a breadth-first tree. It starts from s, finds the vertices at distance 1 from s, then the vertices at distance 2, etc.

Algorithm 10.1.1: Breadth-first Search

```
Input: directed or undirected graph $G = (V, E)$, vertex $s$
Output: breadth-first tree $T$ from $s$
        $V_i = \{$all vertices at distance $i$ from $s\}$

$V_0 = \{s\}$
make $s$ the root of $T$
$i = 0$
while $V_i \neq \emptyset$ do /* construct $V_{i+1}$ */
    $V_{i+1} = \emptyset$
    for each vertex $v \in V_i$ do
    /* “scan” $v$ */
        for each edge $(v, w)$ do
            if $w \notin \bigcup_j V_j$ then
                make $w$ the next child of $v$ in $T$
                add $w$ to $V_{i+1}$
    $i = i + 1$
```

F3: The high-level algorithm can be implemented to run in total time $O(n+m)$. The main data structure is a queue of vertices that have been added to T, but whose children in T have not been computed.

F4: In general we verify that an algorithm takes time $O(n+m)$ by checking that it spends constant time (i.e., $O(1)$ time) on each vertex and edge of G.

F5: Not every shortest path tree is a breadth-first tree (e.g., the tree of Figure 10.1.1). This does not cause any problems in applications.

F6: The diameter can be found by doing a breadth-first search from each vertex.

F7: Dijkstra's algorithm computes a shortest path tree from s in a graph with a nonnegative length function. It generalizes breadth-first search. Like bfs it finds the set V_d of all vertices at distance d from s, for increasing values of d. An appropriate data structure implements the algorithm in time $O(m + n\log n)$ [FrTa87, CLRS01].

10.1.2 Depth-First Search

Depth-first search (abbr. ***dfs***) was investigated in the 19th century as a strategy for exploring a maze [Lu82, Tarr95]. The fundamental properties of the depth-first search tree were discovered by Hopcroft and Tarjan [HoTa73a, Ta72]. Tarjan also developed many other elegant and efficient dfs algorithms (see §10.1.6). The idea of depth-first search is to scan repeatedly an edge incident to the most recently discovered vertex that still has unscanned edges.

DEFINITIONS

D10: Two vertices in a tree are ***related*** if one is an ancestor of the other.

D11: In an undirected graph $G = (V, E)$, a ***depth-first tree*** (abbr. ***dfs tree***) from a vertex s is a tree subgraph T, rooted at s, that contains all the vertices of G that are reachable from s.

- Edges of $E(T)$ and $E(G) - E(T)$ are called ***tree edges*** and ***nontree edges***, respectively.
- Each nontree edge is also called a ***back edge***.

The crucial property is that the two endpoints of each back edge are related.

D12: In an undirected graph G, a ***depth-first spanning forest*** is a collection of depth-first trees, one for each connected component of G. Each vertex of G belongs to exactly one tree of the forest.

D13: Let $G = (V, E)$ be a directed graph where every vertex is reachable from a designated vertex s. A ***depth-first tree*** from s is an ordered tree in G, rooted at s that contains all vertices V. Each edge of T is called a ***tree edge***. Each ***nontree edge*** $(x, y) \in E - T$ can be classified into one of three types:

- A ***back edge*** has y an ancestor of x.
- A ***forward edge*** has y a descendant of x.
- A ***cross edge*** joins two unrelated vertices.

The crucial property is that each cross edge (x, y) has x to the right of y.

D14: Let G be a directed graph, in which we no longer assume that some vertex can reach all others. A ***depth-first forest*** is an ordered collection of trees in G so that each vertex of G belongs to exactly one tree. The edges of G are classified into the 4 types of edges in Definition 13 with one additional possibility:

- A ***cross edge*** can join 2 vertices in different trees as long as it goes from right to left (i.e., from a higher numbered tree to a lower numbered tree).

EXAMPLES

E7: Figure 10.1.2 illustrates a depth-first search of an undirected graph. In drawings of depth-first spanning trees, tree edges are solid and nontree edges are dashed. There can be many depth-first trees with the same root. For instance the tree edge $(5, 6)$ could be replaced by $(5, 7)$.

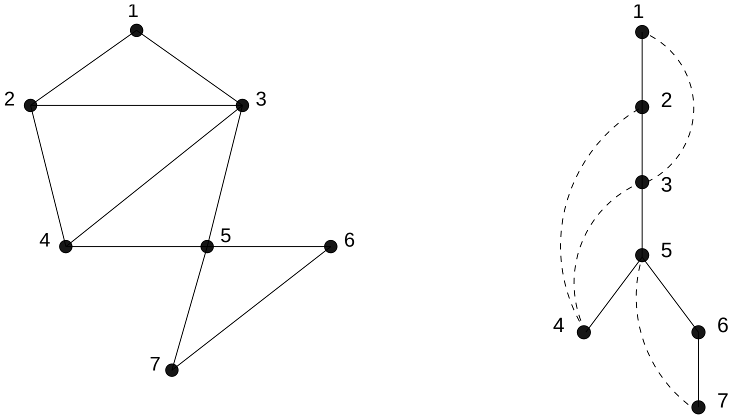


Figure 10.1.2 **Undirected graph and depth-first spanning tree.**

E8: Figure 10.1.3 illustrates a depth-first search of a directed graph. There is 1 forward edge, 2 back edges and 2 cross edges.

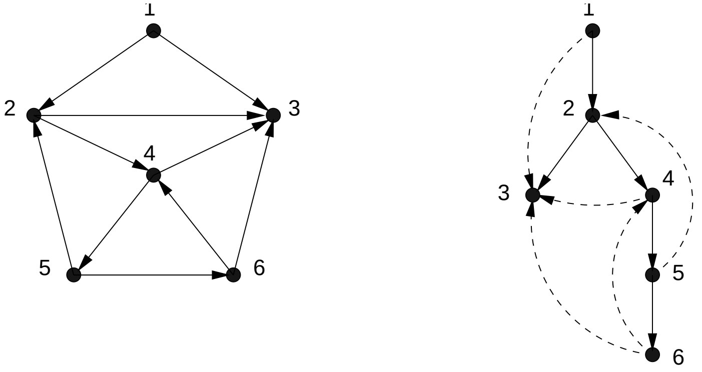


Figure 10.1.3 **Directed graph and depth-first spanning tree.**

FACTS

F8: Any vertex s of an undirected graph has a depth-first tree from s. Any vertex s of a directed graph has a depth-first tree of the subgraph induced by the vertices reachable from s. A high level algorithm to find such a tree is the following.

Algorithm 10.1.2: Depth-First Search

```
Input: directed or undirected graph $G = (V, E)$, vertex $s$
Output: depth-first tree $T$ from $s$

make $s$ the root of $T$
DFS($s$)

procedure DFS($v$)
/* vertex $v$ is discovered at this point */
for each edge $(v, w)$ do
    /* edge $(v, w)$ is scanned (from $v$) at this point */
    if $w$ has not been discovered then
        make $w$ the next child of $v$
        DFS($w$)
/* vertex $v$ is finished at this point */
```

F9: The procedure DFS is recursive, i.e., it calls itself. The overhead for a recursive call is $O(1)$. Algorithm 10.1.2 uses linear time, $O(n + m)$.

F10: If scanning edge (v, w) from the vertex v results in the discovery of the vertex w, then (v, w) is a tree edge.

F11: Suppose that the graph G is undirected. For the tree T produced by Algorithm 10.1.2 to be a valid depth-first tree, any edge $(v, w) \in E - T$ must have v and w related vertices. Why does T have this property? By symmetry suppose v gets discovered before w. Then w will either be made a child of v (like edge $(3, 5)$ in Figure 10.1.2) or a nonchild descendant of v (like edge $(3, 4)$ in Figure 10.1.2).

F12: Suppose that the graph G is directed. For T to be a valid depth-first tree, any edge (v, w) must be one of the 4 possible types. Why does T have this property? First suppose v gets discovered before w. In that case w will be a descendant of v and (v, w) will be a tree or forward edge (as in Fact 11). Next suppose v is discovered after w. Then either v descends from w or v is to the right of w. In the former case (v, w) is a back edge and in the latter case (v, w) is a cross edge.

F13: Algorithm 10.1.2 can be extended to a procedure that constructs a depth-first forest F: The procedure starts with $F = \emptyset$. It repeatedly chooses a vertex $s \notin F$, uses DFS(s) to grow a depth-first tree T from s, and adds T to F.

F14: Algorithm 10.1.2 uses linear time. (For directed graphs a point to note is that a vertex w gets added to only 1 tree of F. This is because once discovered, vertex w remains "discovered" throughout the whole procedure.)

REMARKS

R1: We can test whether an undirected graph is connected in linear time, by using a depth-first search. The trees of a depth-first search spanning forest give the connected components.

R2: We can test whether all vertices of a directed graph are reachable from a vertex s in linear time, by a depth-first search.

Discovery Order

DEFINITIONS

D15: ***Discovery order*** is a numbering of the vertices from 1 to n in the order they are discovered. This is also called the ***preorder*** of the dfs tree.

D16: In ***finish time order*** the vertices are numbered from 1 to n by increasing finish time. This is the ***postorder*** of the dfs tree.

FACTS

F15: Most algorithms based on the depth-first search tree use discovery order. These algorithms identify each vertex v with its discovery number, also called v. This is how the vertices are named in Figure 10.1.3.

F16: In discovery order, the descendants of a vertex v are numbered consecutively, with v first, followed by all its proper descendants. This gives a quick way to test if a given vertex w descends from another given vertex v: Let v have d descendants. w is a descendant of v exactly when $v \leq w < v + d$. This method can be implemented to run in $O(1)$ (i.e., constant) time.

REMARKS

R3: The power of depth-first search comes from its simplification of the edge structure — the absence of cross edges in undirected graphs, and the absence of left-to-right edges in directed graphs. Depth-first search algorithms work by propagating information up or down the dfs tree(s).

R4: Many simple properties of graphs can be analyzed without using the full power of depth-first search. The algorithm always works with a path in the dfs tree, rather than with the entire dfs tree. The algorithm propagates information along the path.

R5: As a simple example of Remark 4 we give a procedure that shows an undirected graph with minimum degree δ has a path of length $> \delta$: execute $\texttt{DFS}(s)$ (for any s), stopping at the first vertex t that becomes finished. The portion of tree T constructed by this procedure is a path from s to t of length $> \delta$. The reason is that all of t's neighbors must be in the path for t to be finished.

R6: Sections 10.1.3–5 deal with simpler graph properties that can be handled by the path view of depth-first search. §10.1.6 covers deeper properties whose algorithms require the full power of the depth-first search tree. §10.1.7 deals with both views of depth-first search.

10.1.3 Topological Order

Topological order is the fundamental property of directed acyclic graphs. In conjunction with dynamic programming, topological order leads to efficient algorithms for many fundamental properites of directed acyclic graphs — even properties that are NP-complete in general graphs.

DEFINITIONS

D17: A ***dag*** is a directed acyclic graph, i.e., it has no cycles.

D18: A ***source*** of a dag is a vertex with indegree 0, and a ***sink*** is a vertex with outdegree 0.

D19: A ***topological numbering (topological order, topological sort)*** of a directed graph assigns an integer to each vertex so that each edge is directed from lower number to higher number.

EXAMPLES

E9: The dag of Figure 10.1.4 has source a and sink f. Alphabetic order is a valid topological ordering. In general a dag has many topological numberings. In this figure 12 are possible.

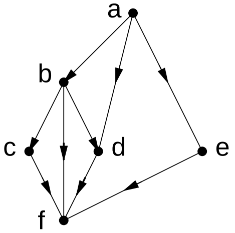


Figure 10.1.4 **Dag and topological order.**

E10: A dag can always be drawn so that all edges are directed downwards, as in Figure 10.1.4. Topological numbers guide the vertical placement of the vertices. This principle is useful in algorithms for drawing graphs (see Section 10.3).

E11: Prerequisite graphs in a university department are dags: if course X is a prerequisite to course Y, then an arrow is drawn from X to Y. There cannot be a cycle, else no one could graduate! The course numbering is a topological numbering: a prerequisite to a course always has a lower number.

E12: A combinational circuit is a collection of logic gates and interconnecting wires, with no feedback. The no-feedback property makes it a dag.

E13: A graph of program dependencies is a dag (assuming no recursion is allowed). For instance the dependencies specified by a `makefile` is a dag. The `make` utility always ensures that a file's timestamp is no later than the timestamp of any dependent file. Thus the timestamps form a topological numbering.

E14: The formulas in a spreadsheet depend on one another, and this dependence relation is a dag. When the value of a cell is changed, the values of dependent cells are recalculated in topological order.

E15: In ecology, a *food web* is a graph whose vertices are the species of an ecosystem. An arrow is drawn from one species to all the other species it preys upon. This model is commonly assumed to be a dag, to disallow cycles in the food chain.

FACTS

F17: Every dag has one or more sources and one or more sinks. This can be seen by examining a path of maximal length. The first (last) vertex must be a source (sink), since otherwise the path could be extended at the beginning (end).

F18: A graph with a topological numbering is a dag. To see this observe that topological numbers increase along a path, so a path cannot return to its starting vertex. Thus no cycle exists.

F19: Any dag has a topological numbering. To construct such a numbering with lowest number 1, assign the lowest number to a source s. Then proceed recursively on the dag $G - s$, using lowest number 2.

F20: One can similarly construct a topological numbering by repeatedly numbering a sink s with the highest number, and proceeding recursively on dag $G - s$.

F21: The strategy of Fact 20 can be implemented efficiently by depth-first search. The reason is that as we grow a depth-first path in a dag, the first vertex to become finished is a sink. More succinctly, we can grow a depth-first path until a sink is reached. This gives the following high-level algorithm.

Algorithm 10.1.3: Topological Numbering (High Level)

```
Input: dag $G = (V, E)$
Output: topological numbering of $G$: vertex $v$ has number $I[v]$

repeat until $G$ has no vertices:
    grow a dfs path $P$ until a sink $s$ is reached
    set $I[s] = n$, decrease $n$ by 1 and delete $s$ from $P$ & $G$
```

To make this algorithm efficient, each iteration grows the dfs-path P by starting with the previous P and extending it, if possible.

F22: A lower level implementation of Algorithm 10.1.3 runs in linear time. The idea is to use array $I[1..n]$ for 2 purposes:

$$I[v] = \begin{cases} 0 & \text{if } v \text{ has never been in } P \\ t & \text{if } v \text{ has been deleted and assigned topological number } t \end{cases}$$

Algorithm 10.1.4: Topological Numbering (Lower Level)

```
Input: dag $G = (V, E)$
Output: topological numbering of $G$: vertex $v$ has number $I[v]$

$num = n$;
for each vertex $v$ do $I[v] = 0$
for each vertex $v$ do if $I[v] = 0$ then DFS($v$)

procedure DFS($v$)
for each edge $(v, w)$ do
    if $I[w] = 0$ then DFS($w$)
/* $v$ is now a sink in the high level algorithm */
$I[v] = num$; decrease $num$ by 1
/* $v$ is now deleted in the high level algorithm */
```

F23: Algorithm 10.1.4 runs in linear time. It spends $O(1)$ time on each vertex and edge.

EXAMPLE

E16: Figure 10.1.5 illustrates how the algorithm numbers the dag of Figure 10.1.4.

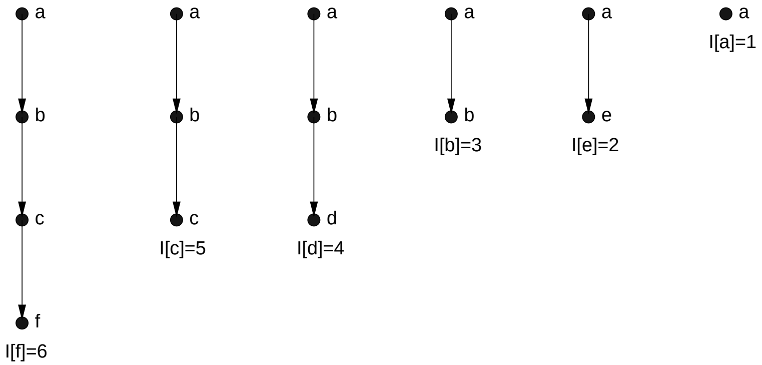


Figure 10.1.5 **Execution of topological numbering algorithm.**

FACTS

F24: Listing the vertices in order of decreasing finish time (Definition 16) is a valid topological order.

F25: Tarjan's algorithm for topological order [Ta74b, CLRS01] is based on Fact 24. Algorithm 10.1.4 is a reinterpretation of Tarjan's algorithm.

To illustrate this, Figure 10.1.6 shows a dfs tree for Figure 10.1.4. Each vertex is labelled by its name and finish number. Subtracting each finish number from 7 gives the topological number of Figure 10.1.5.

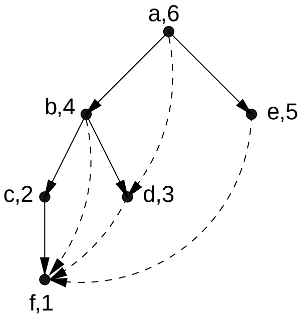


Figure 10.1.6 **Topologically numbering by finish times.**

F26: Another linear-time topological numbering algorithm [Kn73] works by repeatedly deleting a source. The algorithm maintains a queue of sources, as well as the in-degree of each vertex. If the in-degrees are not initially available this algorithm can do more work than Algorithm 10.1.4, since it makes two passes over the graph.

F27: Dag algorithms often propagate information from higher topological numbers to lower, either after scanning each edge (v, w) or at the end of $\texttt{DFS}(v)$. Propagating information in the opposite direction is also possible.

F28: As an example suppose each edge e of a dag G has a real-valued length $\ell[e]$. We can find the longest path in G in linear time. The idea is to set $d[v]$ to the length of a longest path starting at v. These values $d[v]$ can be computed in reverse topological order, using the recurrence

$$d[v] = \max\{0,\ \ell[v,w] + d[w] : (v,w) \in E\}$$

It is easy to modify `DFS` to calculate these values.

The algorithm can recover the longest path from the $d[\,]$ values in a second pass. The second pass can be faster if the first pass stores a pointer for each vertex indicating its successor on its longest path. Longest paths are useful in critical path scheduling. Finding the longest path in a general graph is NP-complete.

F29: Similar algorithms can be used to calculate the longest path from s to t, shortest paths from a vertex s, etc.

F30: More generally Fact 28 illustrates how the technique of dynamic programming can be used to solve problems on dags. Dynamic programming is based on similar recurrences [CLRS01].

EXAMPLE

E17: Figure 10.1.7 illustrates how the algorithm finds a longest path in a dag. Edges are labelled with their length, and vertices are labelled with their $d[\,]$ values. The longest path corresponds to the largest $d[\,]$ value, which is 5; it is the upper path from source to sink.

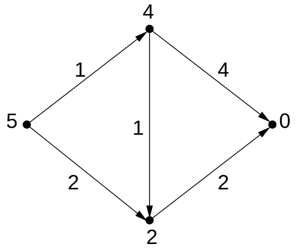


Figure 10.1.7 **Longest path algorithm.**

10.1.4 Connectivity Properties

Depth-first search is the method of choice to calculate low order connectivity information. This section surveys notions of 1- and 2-connectivity. It starts with 1-connectivity of directed graphs, and it then examines 2-connectivity of undirected graphs. These connectivity algorithms are originally due to Tarjan [Ta72]. This section follows the path-based development of [Ga00], which simplifies the algorithms to eliminate the depth-first spanning tree.

Strong Components of a Directed Graph

In this section, $G = (V, E)$ is a directed graph.

DEFINITIONS

D20: For two vertices u and v, a ***uv-path*** is a path starting at u and ending at v.

D21: A directed graph $G = (V, E)$ is ***strongly connected*** if for every two distinct vertices u and v, there is a uv-path and a vu-path.

D22: In general, a directed graph will not be strongly connected. But the vertices can be partitioned into blocks that are strongly connected, according to this definition: two vertices u & v are in the same ***strong component (SC)*** if and only if they can reach each other, i.e., there is a uv-path and a vu-path. This defines a partition of V since it is an equivalence relation.

D23: For any directed graph G, contracting each SC to a vertex gives the ***strong component graph (SC graph)*** (also called the *condensation* of G).

D24: A ***tournament*** is a directed graph G such that each pair of vertices is joined by exactly one edge. This models a round robin tournament, where edge (x, y) represents the fact that player x beat player y.

FACTS

F31: Let C be a cycle in a graph G. All vertices of C are in the same SC. Contracting the vertices of cycle C to a single vertex yields a graph with the same SC graph as G.

F32: The SC graph is always a dag. This follows from Fact 31.

F33: A topological numbering of the SC graph of a tournament gives a ranking of the players. To see why, note that if player x is in an SC with lower topological number than y, then the tournament contains the edge (x, y) (not (y, x)). Thus SC number 1 contains the players that are unequivocally in the top tier — they all beat all other players. SC number 2 contains the 2nd tier players — they all beat all other players except those in tier 1, etc.

F34: All the vertices on a cycle belong to the same SC. In fact the SC graph is formed by repeatedly contracting cycles, until no cycle remains.

F35: A sink s is a vertex of the SC graph. In fact the SC's are $\{s\}$ and the SC's of $G - s$.

F36: Facts 34–35 justify the following high-level algorithm for finding the SC graph. It repeatedly contracts a cycle or deletes a sink.

F37: Next we present a linear-time depth-first search algorithm for finding the strong components and the SC graph of a given directed graph.

```
Algorithm 10.1.5:  Strong Components

  Input: directed graph $G = (V, E)$
  Output: strong components of $G$

  repeat until $G$ has no vertices:
      grow a dfs path $P$ until a sink or a cycle is found
          sink $s$: mark $\{s\}$ as an SC & delete $s$ from $P$ & $G$
          cycle $C$: contract the vertices of $C$
```

Like Algorithm 10.1.3, for efficiency each iteration grows P by starting with the previous P and extending it, if possible.

F38: The algorithm has a low-level implementation that finds the SC graph in linear time [Ga00]. Sinks are deleted similar to Algorithm 10.1.3. Cycles are contracted using a stack to represent P and another stack to give the boundaries of contracted vertices in P.

F39: The algorithm discovers each SC as a sink of the SC graph. So the SC's can be numbered in topological order by the method of Algorithm 10.1.3.

F40: The first linear-time algorithm for strong components is due to Tarjan [Ta72]. It computes a value called $lowpoint(v)$ for each vertex v. $lowpoint(v)$ is the lowest-numbered vertex (in preorder) in v's SC that is reachable from v by a path of (0 or more) tree edges followed by a back or cross edge ($lowpoint(v)$ equals v if no smaller numbered vertex can be reached). The vertices with $lowpoint(v) = v$ are the "roots" of the strong components.

F41: A third linear-time strong component algorithm is due to Sharir [Sh81] and Kosaraju (unpublished; see also [CLRS01]). It does a depth-first search, followed by a second depth-first search on the reverse graph. This makes good sense — the first search discovers which vertices can reach which others, and the second search discovers which vertices can be reached by which others.

EXAMPLES

E18: Figure 10.1.8 shows a directed graph, its three strong components, and its SC graph. Each strong component is strongly connected.

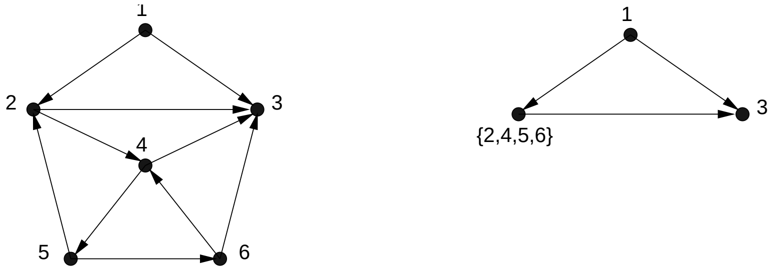


Figure 10.1.8 **Strong components of a directed graph.**

An elementary misperception is that a strongly connected graph has a Hamiltonian cycle. The component $\{2, 4, 5, 6\}$ illustrates that this is not always true.

E19: Figure 10.1.9 gives a dfs tree of Figure 10.1.8. (To better illustrate the algorithm a different dfs from Figure 10.1.8 is used.) Each vertex is labelled by its preorder number followed by its lowpoint value.

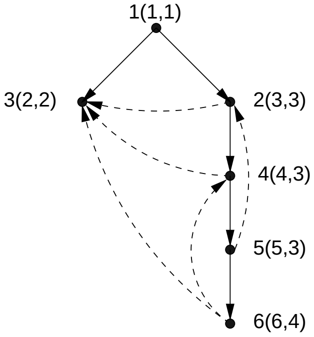


Figure 10.1.9 **Execution of strong component algorithm.**

E20: Suppose we number the vertices of an arbitrary directed graph by topologically numbering the SC graph, and then listing first the vertices in SC number 1, then the vertices in SC number 2, etc. The adjacency matrix of the graph with new vertex numbers is upper block triangular. This is because no edge goes from a higher numbered SC to a lower numbered SC. For instance Figure 10.1.10 gives the adjacency matrix. It is upper triangular except for the block corresponding to SC $\{b, d, e\}$.

$$\begin{array}{c|ccccc|}
 & \text{a} & \text{b} & \text{d} & \text{e} & \text{c} \\ \hline
\text{a} & 0 & 1 & 1 & 1 & 1 \\
\text{b} & 0 & 0 & 0 & 1 & 1 \\
\text{d} & 0 & 1 & 0 & 0 & 1 \\
\text{e} & 0 & 0 & 1 & 0 & 1 \\
\text{c} & 0 & 0 & 0 & 0 & 0 \\ \hline
\end{array}$$

Figure 10.1.10 Upper block triangular adjacency matrix.

E21: Example 20 shows how the SC graph is used to speed up operations on sparse matrices like Gaussian elimination, matrix inversion, finding eigenvalues, etc. The given matrix M is interpreted as a directed graph, with m_{ij} corresponding to edge (i, j). The adjacency matrix of Example 20 is constructed, and the 1 for each edge (i, j) is replaced by the value m_{ij}. The resulting block upper triangular matrix has less fill-in for Gaussian elimination and nice properties for other matrix operations [Ha69].

E22: Figure 10.1.11 below illustrates the execution of the algorithm on the graph of Figure 10.1.8.

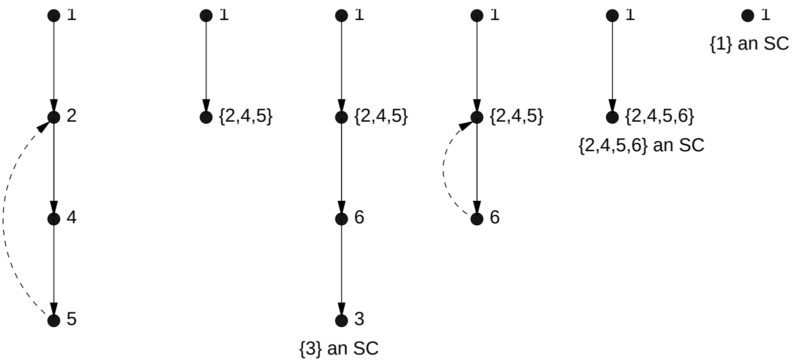


Figure 10.1.11 Execution of strong component algorithm.

E23: Figure 10.1.12 below shows a tournament and its SC graph. Player a is first, players b, d, e are in the 2nd tier, and player c is last.

E24: A Markov chain is *irreducible* if the graph of its (nonzero) transition probabilities is strongly connected.

REMARK

R7: The algorithm of Fact 41 is very simple to code and is covered in many textbooks. It can be appreciably slower than the other two algorithms, because it makes two passes over the graph and has larger memory requirement.

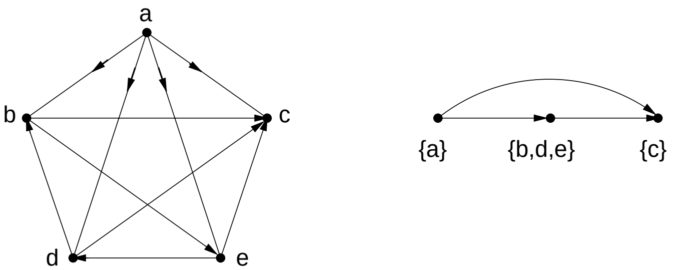


Figure 10.1.12 **Tournament and its SC graph.**

Bridges and Cutpoints of an Undirected Graph

In this section $G = (V, E)$ is a connected undirected graph.

DEFINITIONS

D25: A vertex v is an ***cutpoint (articulation point)*** if $G - v$ is not connected. A graph is ***biconnected*** if it has no cutpoint.

D26: A ***biconnected component*** is a maximal subgraph that has no cutpoint.

D27: An edge e is a ***bridge*** if $G - e$ is not connected. An edge is a bridge if and only if it's not in any cycle. A graph is ***bridgeless*** if it has no bridges.

D28: Let B be the set of all bridges of G. The ***bridge components (BCs)*** of G are the connected components of $G - B$. Equivalently a BC is the induced subgraph on a maximal set of vertices, any of which can reach any other without crossing a bridge.

D29: Contracting each BC to a vertex gives a tree, the ***bridge tree***.

D30: An ***orientation*** of an undirected graph assigns a unique direction to each edge.

D31: A ***perfect matching*** of an undirected graph G is a spanning subgraph in which every vertex has degree exactly 1.

TWO EXAMPLES

E25: Figure 10.1.13 shows a graph with 3 bridges, 6 cutpoints, and 7 biconnected components. It illustrates that an end of a bridge is a cutpoint unless it has degree one. However, a cutpoint need not be the end of a bridge.

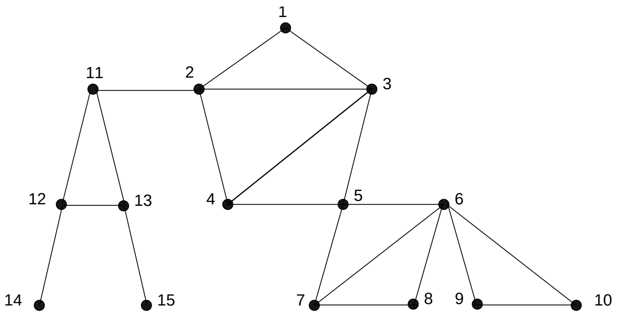


Figure 10.1.13 **Undirected graph with bridges and cutpoints.**

E26: If a communications network (e.g., Internet) has a bridge, that link's failure disables communication, i.e., there are sites that cannot send messages to each other. If the network has an articulation point, that site's failure also disables communication.

FACTS

F42: All vertices on a cycle are in the same BC. In fact the bridge tree is formed by repeatedly contracting cycles.

F43: A vertex x of degree ≤ 1 is a vertex of the bridge tree. In fact the BC's are $\{x\}$ and the BC's of $G - x$.

F44: Facts 42 and 43 justify the following high level algorithm for finding the bridges and bridge tree. It has a linear-time implementation almost identical to Algorithm 10.1.5, the strong component algorithm. We call the last vertex x of a dfs path a ***dead end*** if x has degree ≤ 1.

Algorithm 10.1.6: Bridges

Input: connected undirected graph $G = (V, E)$
Output: bridge components and bridges of G

```
repeat until $G$ has no vertices:
    grow a dfs path $P$ until a cycle is found or a dead end is reached
        cycle $C$: contract the vertices of $C$
        dead end $x$: mark $\{x\}$ as a BC
                    if $x$ has degree 1, then mark its edge as a bridge (of $G$)
```

F45: A similar linear-time algorithm finds the cutpoints and biconnected components of an undirected graph [Ga00].

F46: The original linear-time dfs algorithm of Hopcroft and Tarjan for cutpoints and biconnected components [Ta72] is based on the idea of lowpoints (recall Fact 40).

Start with a dfs tree T. Assume that the vertices are numbered in discovery order and that each vertex is identified with its discovery number. Define

$$lowpoint(v) = \min\{v\} \cup \{w : \text{some back edge goes from a descendant of } v \text{ to } w\}$$

Hopcroft and Tarjan proved that G is biconnected if and only if

(i) vertex 1 has exactly one child (which must be vertex 2);

(ii) $lowpoint(2) = 1$;

(iii) each vertex $w > 2$ has $lowpoint(w) < v$, where v is the parent of w.

The cutpoints have a similar characterization.

Lowpoint is easy to compute in a bottom-up pass over T, since

$$lowpoint(v) = \min\{v\} \cup \{lowpoint(w) : w \text{ a child of } v\} \cup \{w : (v, w) \text{ a back edge}\}$$

MORE EXAMPLES

E27: Figure 10.1.14 below illustrates the execution of the Bridges algorithm on the graph of Figure 10.1.13.

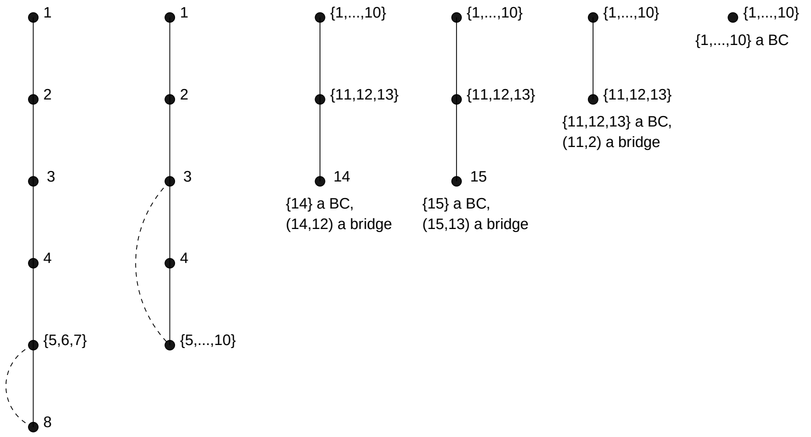


Figure 10.1.14 Execution of bridge algorithm.

E28: Figure 10.1.15 below illustrates **Robbins's Theorem** that a connected undirected graph has a strongly connected orientation if and only if it is bridgeless [Ro39]. If one of the horizontal edges is deleted, making the other a bridge, then the graph has no strongly connected orientation.

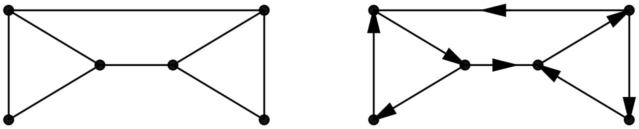


Figure 10.1.15 Undirected graph and strongly connected orientation.

E29: **Kotzig's Theorem** [Ko79] states that a unique perfect matching must contain a bridge of G. Figure 10.1.16 shows a graph with a unique perfect matching — matched edges are drawn heavy. Note that deleting the bridge of the matching gives another graph with a unique perfect matching. This idea can be used to efficiently find a unique perfect matching or show it does not exist [GaKaTa01].

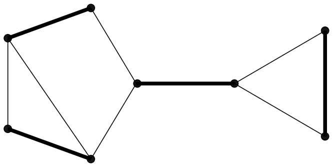


Figure 10.1.16 Graph with a unique perfect matching.

E30: **Whitney's Flipping Theorem** asserts that a graph is planar if and only if each biconnected component is planar [Wh32a].

10.1.5 DFS as a Proof Technique

In addition to being a powerful algorithmic tool, depth-first search can be used to easily prove many theorems of graph theory. (It's a handy way to remember the theorems too!) This subsection gives several examples.

DEFINITIONS

D32: A ***mixed graph*** G can have both directed and undirected edges.

D33: A mixed graph G is ***traversable*** if every ordered pair of vertices u, v has a uv-path with all its directed edges pointing in the forward direction. (Traversability is equivalent to connectedness if G is undirected and to strong connectedness if G is directed.)

D34: A ***bridge*** in a mixed graph is an undirected edge that is a bridge of G when edge directions are ignored.

D35: An ***orientation*** of a mixed graph assigns a unique direction to each undirected edge.

EXAMPLES

E31: Robbins's Theorem can be proved using the high-level bridge algorithm (Algorithm 10.1.6) and the strong components algorithm (Algorithm 10.1.5). When the BC algorithm is executed on a bridgeless graph G, it ends with G contracted to a single vertex. But if the SC algorithm ends with the entire graph contracted to a single vertex, then the initial graph is strongly connected. So orient the given undirected graph G to make the execution of the SC algorithm on the orientation mimic the execution of the BC algorithm on G. To do this orient edges that extend the dfs path or cause contractions (in the BC algorithm) so they do the same in the SC algorithm.

This is illustrated in Figure 10.1.17, which shows how a depth-first search executed on the undirected graph of Figure 10.1.15 gives the orientation shown in that figure. Enlarged hollow vertices are contractions of original vertices.

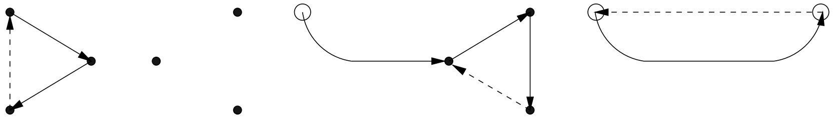


Figure 10.1.17 **Dfs proof of Robbins's Theorem.**

E32: The same approach proves a generalization of Robbins's theorem by Boesch and Tindell [BoTi80] that a traversable graph has a strongly-connected orientation if and only if it has no bridge. It can be proved using Algorithm 10.1.5, with the sink rule replaced by a rule for a "1-sink", i.e., a vertex with no leaving directed edge and only one incident undirected edge.

E33: Kotzig's Theorem can be proved by dfs [Ga79]. We illustrate by proving a simple special case: a bipartite graph with a unique perfect matching has a vertex of degree one. The idea is to grow a dfs path P two edges at a time, repeatedly adding an unmatched edge (x, y) and the matched edge containing y. When the path cannot be extended the last vertex y has degree 1. If not a back edge from y creates an even length cycle, whose edges yield another perfect matching as shown in Figure 10.1.18 below.

A linear-time dfs algorithm for testing if a perfect matching is unique is given in [GaKaTa01].

E34: ***Rédei's Theorem*** [Re34] states that any tournament has a Hamiltonian path, i.e., a simple path through all its vertices. This is easy to see by dfs: listing the vertices in order of decreasing finish time gives a Hamiltonian path.

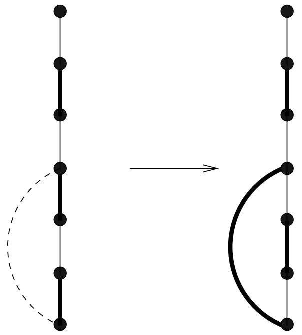


Figure 10.1.18 **Dfs proof of Kotzig's Theorem.**

10.1.6 More Graph Properties

The basic properties of depth-first search were developed by Hopcroft and Tarjan as stepping-stones to their goal of an efficient planarity algorithm. This subsection starts by surveying the high-level principles of the planarity algorithm. It then surveys other important properties that can be decided by efficient dfs algorithms. The depth-first tree plays a central role in all these algorithms.

Planarity Testing

The first complete linear-time algorithm to decide whether or not a graph is planar is due to Hopcroft and Tarjan. This property has obvious applications to graph drawing, circuit layout, etc. This section gives the high-level depth-first approach.

DEFINITIONS

D36: Let G be a biconnected graph with a cycle C. The edge set $E - E(C)$ can be partitioned into a family of subgraphs called ***segments*** as follows:

(i) An edge not in C that joins 2 vertices of C is a segment.

(ii) The remaining segments each consist of a connected component of $G - V(C)$, plus all edges joining that component to C.

D37: Two segments S, T of a cycle C in a graph ***interlace*** either if $|V(S) \cap V(T) \cap V(C)| \geq 3$, or if there are 4 distinct vertices u, v, w, x that occur along cycle C (not necessarily consecutively) in that order such that $u, w \in S$ and $v, x \in T$.

EXAMPLE

E35: Figure 10.1.19 below shows a cycle C (dotted) with 5 segments. Segments S_1 and S_2 interlace, and S_4 interlaces with both S_3 and S_5.

FACTS

F47: By Whitney's Flipping Theorem (Example 30), one can test planarity by treating each biconnected component separately.

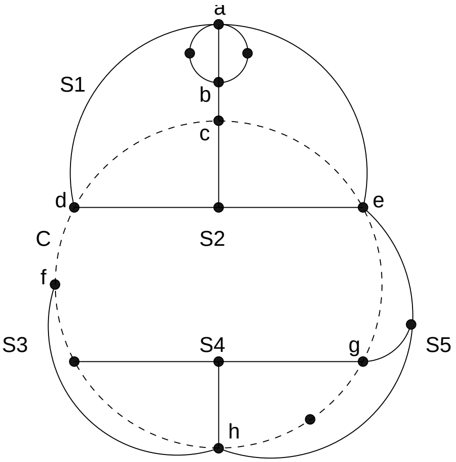


Figure 10.1.19 **Planar graph with interlacing segments.**

F48: The graph theoretic approach used by Hopcroft and Tarjan is the following theorem of Auslander and Parter [AuPa61]: a biconnected graph G is planar if and only if

(a) $C \cup S$ is planar for every segment S;

(b) the segments can be partitioned into two families such that no two segments in the same family interlace.

The necessity of both (a) and (b) is clear. An outline of a complete proof of this theorem is given in [Ev79].

F49: Here is the overall structure of the algorithm of Hopcroft and Tarjan [HoTa74] which decides in linear time whether or not a graph is planar. Each biconnected component is processed separately.

A depth-first spanning tree of the component is found.

A cycle C is chosen, consisting of a path in the dfs tree plus one back edge.

Then segments are found:

(i) each back edge that joins two vertices of C is a segment;

(ii) each remaining segment S is determined by a vertex $w \notin C$ whose parent is in C. The edges of S are those edges with at least one endpoint descending from w. (Specifically, this amounts to the tree edge joining w to its parent, plus all edges of the subtree rooted at w, plus all back edges that join two descendants of w or join a descendant of w with a vertex of C.)

The algorithm processes each segment S recursively, checking that $C \cup S$ is planar and S can be added to an imbedding of all subgraphs processed so far. (The latter uses the interlacing criterion.)

F50: A number of additional ideas are used to achieve linear time. The *lowpoint* values (Fact 46) are used to guide the construction of cycles C. In fact the "second lowpoint" is also used. A second depth-first search is done for cycle generation. The planarity algorithm is intricate, but is very fast in practice.

Triconnectivity

Hopcroft and Tarjan show how to find the triconnected components in linear time [HoTa73b]. Like their planarity algorithm the approach is based on segments.

DEFINITIONS

D38: An undirected graph is ***triconnected*** if it is connected and remains so whenever any two or fewer vertices are deleted.

D39: Two vertices in a biconnected graph form a ***separation pair*** if deleting them leaves a disconnected graph.

There is a natural definition of the *triconnected components* of a graph.

EXAMPLE

E36: In Figure 10.1.19 above there are 5 separation pairs: a, b; a, c; d, e; e, f; and g, h.

FACTS

F51: The following characterization of the separation pairs is easy to prove. Let G be a biconnected graph with a cycle C. Let a, b be a separation pair. Then a and b either both belong to C or both belong to a common segment. Moreover, suppose a and b both belong to C. Then either

(a) some segment S has $V(S) \cap V(C) = \{a, b\} \subset V(S)$; or

(b) $C - \{a, b\}$ consists of two nonempty paths, and no segment contains a vertex of both paths.

(The symbol "$\subset$" denotes proper set containment.)

F52: The triconnectivity algorithm applies the characterization of Fact 51 recursively. Hopcroft and Tarjan's triconnectivity algorithm shares algorithmic ideas with their planarity algorithm.

F53: Another useful fact is that the two vertices of a separation pair are related (Definition 10).

Ear Decomposition and st-numbering

DEFINITIONS

D40: An ***open ear decomposition*** of an undirected graph is a partition of the edges into a simple cycle P_0 and simple paths $P_1, \ldots, P_k$ such that for each $i > 0$, P_i is joined to previous paths only at its (2 distinct) ends, i.e., $V(P_i) \cap V(\cup_{j<i} P_j)$ consists of the 2 ends of P_i. (The concept, but not the name, is due to Whitney.)

D41: Let (s, t) be any edge of a biconnected graph. An ***st-numbering*** numbers the vertices from 1 to n so that s is numbered 1, t is numbered n, and every other vertex has both a higher-numbered neighbor and a lower-numbered neighbor.

EXAMPLE

E37: Figure 10.1.20 shows an ear decomposition consisting of cycle P_0 and simple paths $P_1, \ldots, P_6$. The 15 vertices are numbered in an *st*-numbering (corresponding to the ear decomposition).

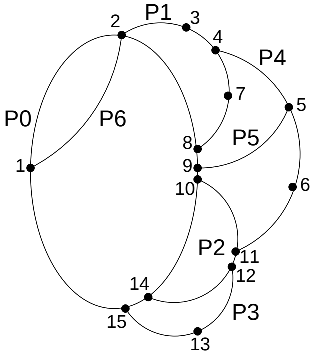


Figure 10.1.20 Ear decomposition and *st*-numbering of a biconnected graph.

FACTS

F54: Whitney [Wh32b] proved that an undirected graph is biconnected if and only if it has an open ear decomposition.

F55: An algorithm based on *lowpoint* values can be used to find an open ear decomposition of a biconnected graph in linear time (*pathfinder* in [EvTa76], although the term "ear decomposition" is not used).

F56: An open ear decomposition with $(s, t) \in P_0$ can be used to give an *st*-numbering in linear time [EvTa76].

REMARKS

R8: *st*-numbering is the basis of the linear-time planarity algorithm of Lempel, Even and Cederbaum [LeEvCe67]. It constructs a planar imbedding by repeatedly adding a vertex. More precisely it starts with an imbedding of one vertex and its incident edges. Then it repeatedly adds all edges incident to the next vertex, updating the imbedding. The vertices are added in *st*-order.

R9: Ear decomposition is closely related to depth-first search. An open ear decomposition can be found efficiently on parallel computers with large numbers of processors; the same cannot be said for doing a depth-first search. Efficient parallel algorithms for bi- and triconnectivity and planarity are based on ear decomposition [Ra93].

Reducibility

DEFINITIONS

D42: A ***(program) flow graph*** is a directed graph with a distinguished vertex r, the ***start vertex***, that can reach every vertex.

D43: A flow graph is ***reducible*** if it can be transformed into the single vertex r by a sequence of operations of the following type:

> if (v, w) is the only edge entering w and if $w \neq r$, then contract edge (v, w) to vertex v.

(The contraction operation discards parallel edges and self-loops.)

D44: We define a problem in data structures that arises in many dfs algorithms (and other contexts). A universe of n elements is given. The problem is to maintain a partition $\mathcal{P}$ of this universe into sets.

> Initially each element forms a singleton set of $\mathcal{P}$.
>
> Partition $\mathcal{P}$ is updated by the operation $union(A, B)$, which replaces two sets A and B of $\mathcal{P}$ by their union $A \cup B$.
>
> A second operation $find(x)$ computes the name of the set currently containing element x.

The ***set-merging problem*** is to process a sequence of m intermixed *union* and *find* operations.

EXAMPLE

E38: Figure 10.1.21 shows an irreducible flow graph. In fact, this flow graph gives a forbidden subgraph characterization of reduciblity: a flow graph is reducible if and only if it does not contain a subgraph consisting of 4 vertices r, a, b, c (where r and a may coincide but otherwise the vertices are distinct) joined by vertex disjoint paths from r to a, a to b, a to c, b to c and c to b.

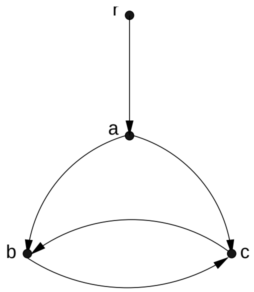


Figure 10.1.21 **Irreducible flow graph.**

REMARK

R10: Flow graphs model the stucture of computer programs. Any program without goto's has a reducible flow graph. Many methods for code optimization (e.g., eliminating common subexpressions, identifying active variables, finding useless definitions, etc.) depend on the graph being reducible [AhSeUl86].

FACTS

The starting point of a linear-time reducibility algorithm of Tarjan [Ta74a] for flow graphs is a reformulation of reducibility. It produces the sequence of contractions that reduce it to the start vertex. For each vertex w in a dfs tree of a flow graph, we define

$$I(w) = \{v : \text{there is a simple } vw\text{-path ending with a back edge (to } w)\}$$

F57: [Ta74a] A flow graph is reducible if and only if every set $I(w)$ consists only of descendants of w.

F58: [Ta74a] Assume that the vertices of the flow graph are indexed by preorder number. Let w be the largest vertex (in preorder) with an entering back edge. Suppose that $I(w)$ consists only of descendants of w. If we contract the vertices of $I(w)$ into vertex w, then the new graph is reducible if and only if the original was.

F59: Fact 58 specifies a sequence of contractions that substantiate the reducibility of a graph. The sets $I(w)$ can be computed simply by scanning edges in the backwards direction, starting at w. If a nondescendant of w is ever reached, then the graph is not reducible. The efficient descendance test of Fact 16 is used.

F60: The contractions performed by the algorithm change the vertex set of the graph. At all times the vertices of the current graph form a partition of the original vertex set. This partition is manipulated by the *union* and *find* operations (Definition 44).

F61: The best known algorithm for set-merging is based on the so-called weighted union and path compression rules. Tarjan showed that this algorithm solves the set-merging problem in time $O(m\alpha(m,n))$. Here α is an inverse of Ackermann's function and is very slowly growing [Ta75, CLRS01].

F62: Gabow and Tarjan [GaTa85] showed that a special case of the set-merging problem can be solved in linear time. Using this special case algorithm makes the reducibility algorithm run in linear time.

F63: Suppose the vertices are numbered in discovery order. If $v < w$ then any vw-path contains a common ancestor of v and w [Ta72]. This property holds in both directed and undirected graphs.

EXAMPLE

E39: Figure 10.1.22 shows a depth-first spanning tree of a flow graph with the vertices labelled by discovery number. Any path from vertex 5 to vertex 8 passes through one or both of the common ancestors of 5 and 8, i.e., vertices 1 and 2.

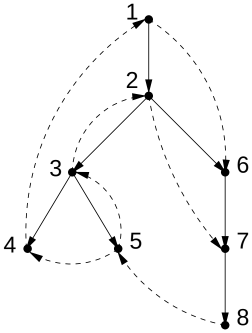


Figure 10.1.22 **Depth-first search with preorder numbers.**

Two Directed Spanning Trees

DEFINITIONS

D45: In a flow graph, an ***r*-tree** is a directed spanning tree rooted at start vertex r.

D46: Consider a dfs tree with root r. Edge (v, w) is a ***(directed) bridge*** in a flow graph if every rw-path includes (v, w).

EXAMPLE

E40: In Figure 10.1.21 edge (r, a) is a bridge. Duplicating it gives a graph with 2 edge-disjoint r-trees (the directed paths r, a, b, c and r, a, c, b).

FACTS

Consider a dfs tree with root r. For any vertex w define

$$I(w) = \{v : \text{there is a simple } vw\text{-path containing only descendants of } w\}$$

Clearly the path of this definition ends in a back edge to w (unless $v = w$). Note the similarity with Fact 57.

F64: A flow graph has two edge-disjoint r-trees if and only if each vertex $v \neq r$ has two edge-disjoint rv-paths (recall Definition 20). This is a special case of ***Edmonds's Branching Theorem***, which is the same statement generalized from 2 to any $k \geq 2$ [Ed72].

F65: A flow graph has 2 edge-disjoint r-trees if and only if there are no bridges.

F66: Tarjan presents a linear-time algorithm to find 2 edge-disjoint r-trees if they exist [Ta74a]. More generally in an arbitrary flow graph the algorithm finds 2 r-trees that contain the fewest possible number of common edges. The more general problem is solved by identifying the bridges and duplicating each of them. This gives a graph with 2 edge-disjoint r-trees (Fact 65).

F67: In terms of dfs, an edge (v, w) is a bridge if and only if (v, w) is a tree edge and is the only edge entering $I(w)$. Tarjan's algorithm identifies the bridges using techniques similar to the reducibility algorithm. Computing the trees is more involved.

F68: The algorithm performs set-merging to keep track of the contracted vertices, as in the reducibility algorithm. As in that algorithm the data structure of [GaTa85] is used to achieve linear time.

Dominators

DEFINITIONS

D47: In a flow graph with start vertex r, vertex v ***dominates*** vertex $w \neq v$ if every rw-path contains v.

D48: The ***immediate dominator*** of w, denoted $idom(w)$, is a vertex v that dominates w such that every other dominator of w dominates v.

D49: The ***dominator tree*** is a tree T whose nodes are the vertices of G. The root of T is the start vertex r. The parent of a vertex $v \neq r$ is $idom(v)$.

D50: The ***internal vertices*** of a path are all its vertices except the endpoints.

D51: Consider a dfs tree with root r. For every vertex $w \neq r$, the ***semidominator*** of w, $sdom(w)$, is defined by

$$sdom(w) = \min\{v : \text{some } vw\text{-path has all its internal vertices} > w\}$$

EXAMPLE

E41: Figure 10.1.23 shows the dominator tree for the graph of Figure 10.1.22. Note that vertex 2 does not dominate 3 because of path $1, 6, 7, 8, 5, 3$. The start vertex 1 is the semidominator of every vertex except two: $sdom(7) = 2$, $sdom(8) = 7$. Although vertex 1 is the immediate dominator of 7 it is not the semidominator of 7.

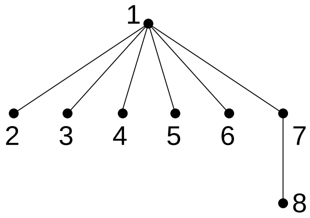


Figure 10.1.23 **Dominator tree for Figure 10.1.22.**

FACTS

F69: The basic properties of dominance are due to Lowry and Medlock [LoMe69]: Every vertex except r has a unique immediate dominator. This justifies the notion of dominator tree. A vertex v dominates w if and only if v is a proper ancestor of w in the dominator tree.

F70: Lengauer and Tarjan [LeTa79] give an efficient algorithm to find the dominator tree T. It is a refinement of an earlier dfs algorithm of Tarjan [Ta74b].

F71: For any vertex w, $sdom(w)$ is a proper ancestor of w. This follows from Fact 63.

FACTS ABOUT SEMIDOMINATORS

Semidominators are useful because of the next two facts proved by Lengauer and Tarjan:

F72: Take any vertex $w \neq r$. Let u be a vertex with minimum value $sdom(u)$ among all vertices in the tree path from $sdom(w)$ to w, excluding $sdom(w)$. Then

$$idom(w) = \begin{cases} sdom(w) & \text{if } sdom(w) = sdom(u) \\ idom(u) & \text{otherwise} \end{cases}$$

F73: Semidominators can be computed by a recursive definition:

$$\begin{aligned} sdom(w) = &\min\{v : (v, w) \text{ an edge}\} \cup \\ &\{sdom(u) : u > w \text{ and some edge goes from a descendant of } u \text{ to } w\} \end{aligned}$$

(Note the similarity with *lowpoint* in Fact 46.)

F74: The algorithm of Lengauer and Tarjan [LeTa79] computes semidominators using Fact 73 in a backwards pass (i.e., w is decreasing). Then it computes immediate dominators using Fact 72 in a forwards pass.

F75: The time for the algorithm is $O(m\alpha(m, n))$. An implementation of this algorithm in linear time is presented in [AlHaLaTh99].

10.1.7 Approximation Algorithms

Finding small spanning subgraphs with prespecified connectivity properties is usually a difficult (NP-hard) problem. For example, finding a bridgeless spanning subgraph with the fewest possible number of edges is NP-hard. (The reason is that this subgraph contains n edges if and only if there is a Hamiltonian cycle.)

Depth-first search has been used to design good approximation algorithms for such difficult problems. Here the goal is to find a subgraph that has all the desired properties except that instead of having the fewest possible number of edges, it is within a small constant factor of this goal. This section surveys the use of depth-first search in approximation algorithms for connectivity properties. Other dfs approximation algorithms are surveyed in [Kh97].

DEFINITIONS

D52: Consider an optimization problem that seeks to find a smallest feasible solution OPT. An ***α-approximation algorithm*** is a polynomial-time algorithm that is guaranteed to find a solution of size at most $\alpha|OPT|$ [CLRS01]. For the graph problems of this section, the size of the solution is the number of edges.

D53: The ***smallest bridgeless spanning subgraph*** of a connected bridgeless undirected graph is a bridgeless spanning subgraph with the minimum possible number of edges.

D54: An undirected graph is ***k-edge connected*** if it is connected and remains so when any set of fewer than k edges is deleted. This concept makes good sense for a multigraph. A ***k-ECSS*** is a k-edge connected spanning subgraph; the graph is assumed to be k-edge connected. So a bridgeless spanning subgraph is a 2-ECSS. A ***smallest k-ECSS*** has the fewest possible number of edges. From now on instead of “smallest bridgeless spanning subgraph”, we use the shorter equivalent phrasing, “smallest 2-ECSS”.

ALGORITHM

Approximation algorithms for the smallest 2-ECSS are our first concern. A 2-approximation can be designed from Algorithm 10.1.6 in a straightforward way. Khuller and Vishkin [KhVi94] were the first to go beyond this. They presented an elegant dfs algorithm based on a “tree carving” using the dfs tree. The following modification of Algorithm 10.1.6 is a path-based reinterpretation of their algorithm.

Algorithm 10.1.7: Smallest 2-ECSS Approximation

```
Input: bridgeless undirected graph $G = (V, E)$
Output: edge set $F \subseteq E$, a 3/2-approximation to the smallest 2-ECSS

$F = \emptyset$
repeat until $G$ has 1 vertex:
    grow a dfs path $P$ until its endpoint $x$ has all neighbors belonging to $P$
    let $y$ be the neighbor of $x$ closest to the start of $P$
    let $C$ be the cycle formed by edge $(x, y)$ & edges of $P$
    add all edges of $C$ to $F$
    contract the vertices of $C$
```

EXAMPLE

E42: Figure 10.1.24 below gives a sample execution of the algorithm. The given graph on top has a Hamiltonian cycle, so the smallest 2-ECSS has n edges.

The algorithm grows the depth-first path of solid edges shown in the middle, starting from r. It then adds the dashed edges.

A typical edge addition is illustrated in the bottom graph, where the enlarged hollow vertex is the contraction of the last vertices on the path.

As n approaches ∞, the algorithm's solution approaches $3n/2$ edges: n solid edges and $n/2$ dashed edges. So the approxmation ratio approaches $3/2$.

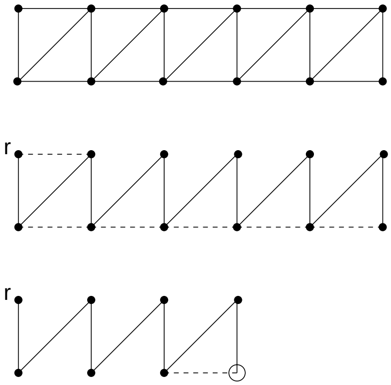


Figure 10.1.24 **Smallest 2-ECSS approximation algorithm.**

FACTS

Good approximation algorithms require good lower bounds on the size of the optimum solution. We analyze this algorithm using 2 lower bounds.

F76: The ***Degree Lower Bound*** says that any 2-ECSS has at least n edges. This results from the fact that every vertex must have degree at least 2.

F77: The ***Carving Lower Bound*** says that if Algorithm 10.1.7 contracts c cycles, then any 2-ECSS has at least $2c$ edges [KhVi94]. To see this let x be an endpoint of P giving a contraction in the algorithm. Any 2-ECSS contains ≥ 2 edges leaving x. These edges disappear in the contraction operation. So we can repeat this argument for every contraction, getting a lower bound of $2c$ edges.

F78: Algorithm 10.1.7 is a $3/2$ approximation. This follows because the edge set F consists of $n - 1$ edges from paths P and c edges that cause contractions. If OPT is the edge set of a 2-ECSS, then $|OPT| > n$ (Degree Lower Bound) and $|OPT|/2 \geq c$ (Carving Lower Bound). Thus $|F| < 3|OPT|/2$.

F79: Vempala and Vetta [VeVe00] present a $4/3$-approximation algorithm for the smallest 2-ECSS. Their algorithm is based on the idea of doing a depth-first search of objects of the graph, specifically cycles and paths. It uses the ***Matching Lower Bound***: Any 2-ECSS has at least as many edges as a smallest spanning subgraph where every vertex has degree ≥ 2. Vempala and Vetta give a similar $4/3$-approximation algorithm for the smallest biconnected subgraph of a biconnected graph.

F80: Jothi, Raghavachari & Varadrajan [JoRaVa03] use a stronger version of the Matching Lower Bound in a dfs algorithm that achieves performance ratio 5/4 for the smallest 2-ECSS. Vetta [Ve01] uses a version of the Matching Lower Bound in a dfs algorithm that approximates the smallest strongly connected subgraph of a strongly connected graph to within a factor 3/2.

F81: The Carving Lower Bound extends to k-ECSS: If Algorithm 10.1.7 contracts c cycles then any k-ECSS has at least kc edges [KhVi94]. This can be proved by simply changing the "2"'s to k's in Fact 77.

F82: Gabow [Ga02] gives a dfs algorithm that is a 3/2-approximation for the smallest 3-ECSS of a multigraph. It uses the above dfs approach of [KhVi94] for 2-ECSS, the Carving Lower Bound, and ear decomposition.

F83: Khuller and Raghavachari [KhRa96] present the first approximation algorithm that achieves ratio < 2 for the smallest k-ECSS of a multigraph. It boosts the edge-connectivity of the solution graph in steps of 2. Each of these steps is a slight variant of the above algorithm of [KhVi94]. The analysis is based on a refinement of the Carving Lower Bound. Gabow [Ga03] improves the analysis to show it is a 1.61-approximation.

References

[AhSeUl86] A. V. Aho, R. Sethi, and J. D. Ullman, *Compilers: Principles, Techniques and Tools*, Addison-Wesley, 1986.

[AlHaLaTh99] S. Alstrup, D. Harel, P. W. Lauridsen, and M. Thorup, Dominators in linear time, *SIAM J. Comput.* 28 (1999), 2117–2132.

[AuPa61] L. Auslander and S. V. Parter, On imbedding graphs in the plane, *J. Math. and Mech.* 10 (1961), 517–523.

[BoTi80] F. Boesch and R. Tindell, Robbins's theorem for mixed multigraphs, *Amer. Math. Monthly* 87 (1980), 716–719.

[BrBr96] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*, Prentice-Hall, 1996.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition, McGraw-Hill, 2001.

[Ed72] J. Edmonds, Edge-disjoint branchings, pp. 91–96 in *Combinatorial Algorithms*, ed. by R. Rustin, Algorithmics Press, New York (1972).

[Ev79] S. Even, *Graph Algorithms*, Computer Science Press, Baltimore, 1979.

[EvTa76] S. Even and R. E. Tarjan, Computing an *st*-numbering, *Theoret. Comp. Sci.* 2 (1976), 339–344.

[FrTa87] M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, *J. ACM* 34 (1987), 596–615.

[Ga79] H. N. Gabow, Algorithmic proofs of two relations between connectivity and the 1-factors of a graph, *Disc. Math.* 26 (1979), 33–40.

[Ga00] H. N. Gabow, Path-based depth-first search for strong and biconnected components, *Inf. Proc. Letters* 74 (2000), 107–114.

[Ga02] H. N. Gabow, An ear decomposition approach to approximating the smallest 3-edge connected spanning subgraph of a multigraph, *Proc. 13th Annual ACM-SIAM Symp. on Disc. Algorithms* (2002), 84–93.

[Ga03] H. N. Gabow, Better performance bounds for finding the smallest k-edge connected spanning subgraph of a multigraph, *Proc. 14th Annual ACM-SIAM Symp. on Disc. Algorithms* (2003), 460–469.

[GaTa85] H. N. Gabow and R. E. Tarjan, A linear-time algorithm for a special case of disjoint set union, *J. Comp. and Sys. Sci.* 30 (1985), 209–221.

[GaKaTa01] H. N. Gabow, H. Kaplan and R. E. Tarjan, Unique maximum matching algorithms, *J. Algorithms* 40 (2001), 159–183.

[GoTa02] M. T. Goodrich and R. Tamassia, *Algorithm Design: Foundations, Analysis and Internet Examples*, John Wiley & Sons, 2002.

[Ha69] F. Harary, *Graph Theory*, Addison-Wesley, Reading MA, 1969.

[HoTa73a] J. Hopcroft and R. E. Tarjan, Efficient algorithms for graph manipulation, *Comm. ACM* 16 (1973), 372–378.

[HoTa73b] J. E. Hopcroft and R. E. Tarjan, Dividing a graph into triconnected components, *SIAM J. Comput.* 2 (1973), 135–158.

[HoTa74] J. Hopcroft and R. Tarjan, Efficient planarity testing, *J. ACM* 21 (1974), 549–568.

[HSR98] E. Horowitz, S. Sahni and S. Rajasekaran, *Computer Algorithms*, Computer-Science Press, Baltimore, 1998.

[JoRaVa03] R. Jothi, B. Raghavachari and S. Varadarajan, A 5/4-approximation algorithm for minimum 2-edge-connectivity, *Proc. 14th Annual ACM-SIAM Symp. on Disc. Algorithms* (2003) 725–734.

[Kh97] S. Khuller, Approximation algorithms for finding highly connected subgraphs, in *Appoximation Algorithms for NP-hard Problems*, ed. by D. S. Hochbaum, PWS Publishing, 1997.

[KhRa96] S. Khuller and B. Raghavachari, Improved approximation algorithms for uniform connectivity problems, *J. Algorithms* 21 (1996), 434–450.

[KhVi94] S. Khuller and U. Vishkin, Biconnectivity approximations and graph carvings, *J. ACM* 41 (1994), 214–235.

[Kl00] J. Kleinberg, The small-world phenomenon: An algorithmic perspective, *Proc. 32nd Annual ACM Symp. on Th. Comput.* (2000), 163–170.

[Kn73] D. E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms*, Second Edition, Addison-Wesley, 1973.

[Ko59] A. Kotzig, On the theory of finite graphs with a linear factor I, *Mat.-Fyz. Časopis Slovensk. Akad. Vied* 9 (1959), 73–91.

[LeEvCe67] A. Lempel, S. Even and I. Cederbaum, An algorithm for planarity testing of graphs, pp. 215–232 in *Theory of Graphs: Int. Symp*, ed. by P. Rosenstiehl, Gordon and Breach, 1967.

[LeTa79] T. Lengauer and R. E. Tarjan, A fast algorithm for finding dominators in a flowgraph, *ACM Trans. on Prog. Lang. and Sys.* 1 (1979), 121–141.

[LoMe69] E. S. Lowry and C. W. Medlock, Object code optimization, *C. ACM* 12 (1969), 13–21.

[Lu:1882] E. Lucas, *Récreations Mathématiques*, Paris, 1882.

[Mi67] S. Milgram, The small world problem, *Psychology Today* 1 (1967), 60–67.

[Ra93] V. Ramachandran, Parallel open ear decomposition with applications to graph biconnectivity and triconnectivity, in *Synthesis of Parallel Algorithms*, ed. by J. H. Reif, Morgan Kaufmann, 1993.

[Re34] L. Rédei, Ein kombinatorischer Satz., *Acta Litt. Sci. Szeged* 7 (1934), 39–43.

[Ro39] H. E. Robbins, A theorem on graphs, with an application to a problem in traffic control, *Amer. Math. Monthly* 46 (1939), 281–283.

[Se02] R. Sedgewick, *Algorithms in C++, Part 5: Graph Algorithms*, Addison-Wesley, 2002.

[Sh81] M. Sharir, A strong-connectivity algorithm and its application in data flow analysis, *Comp. and Math. with Applications* 7 (1981), 67–72.

[St96] I. Stewart, "Cows in the maze", pp.116-118 in Mathematical Recreations, *Sci. American* Dec. 1996.

[Ta72] R. E. Tarjan, Depth-first search and linear graph algorithms, *SIAM J. Comput.* 1 (1972), 146–160.

[Ta74a] R. E. Tarjan, Testing flow graph reducibility, *J. Comput. Sys. Sci.* 9 (1974), 355–365.

[Ta74b] R. E. Tarjan, Finding dominators in directed graphs, *SIAM J. Comput.* 3 (1974), 62–89.

[Ta75] R. E. Tarjan, Efficiency of a good but not linear set union algorithm, *J. ACM* 22 (1975), 215–225.

[Ta76] R. E. Tarjan, Edge-disjoint spanning trees and depth-first search, *Acta Inf.* 6 (1976), 171–185.

[Tarr:1895] G. Tarry, Le probléme des labyrinthes, *Nouvelles Ann. de Math.* 14 (1895), 187.

[VeVe00] S. Vempala and A. Vetta, Factor 4/3 approximations for minimum 2-connected subgraphs, pp. 262–273 in *Appoximation Algorithms for Combinatorial Optimization*, ed. by K. Jansen and S. Khuller, Lecture Notes in Computer Science 1931, Springer-Verlag, 2000.

[Ve01] A. Vetta, Approximating the minimum strongly connected subgraph via a matching lower bound, *Proc. 12th Annual ACM-SIAM Symp. on Disc. Algorithms* (2001), 417–426.

[We99] M. A. Weiss, *Data Structures and Algorithm Analysis in C++*, Second Edition, Addison-Wesley, 1999.

[Wh32a] H. Whitney, Non-separable and planar graphs, *Trans. Amer. Math. Soc.* 34 (1932), 339–362.

[Wh32b] H. Whitney, Congruent graphs and the connectivity of graphs, *Amer. J. Math.* 54 (1932), 150–168.

10.2 DYNAMIC GRAPH ALGORITHMS

Camil Demetrescu, University of Rome "La Sapienza", Italy
Irene Finocchi, University of Rome "Tor Vergata", Italy
Giuseppe F. Italiano, University of Rome "Tor Vergata", Italy

Part 1. Dynamic Problems on Undirected Graphs
10.2.1 General Techniques for Undirected Graphs
10.2.2 Connectivity
10.2.3 Minimum Spanning Trees
Part 2. Dynamic Problems on Directed Graphs
10.2.4 General Techniques for Directed Graphs
10.2.5 Dynamic Transitive Closure
10.2.6 Dynamic Shortest Paths
Research Issues
References

Introduction

In many applications of graph algorithms, including communication networks, VLSI design, graphics, and assembly planning, graphs are subject to discrete changes, such as additions or deletions of edges or vertices. In the last two decades there has been a growing interest in such dynamically changing graphs, and a whole body of algorithms and data structures for dynamic graphs has been discovered. This section of the *Handbook* is intended as an overview of this field.

DEFINITIONS

D1: An ***update on a graph*** is an operation that inserts or deletes edges or vertices of the graph or changes attributes associated with edges or vertices, such as cost or color.

D2: A ***dynamic graph*** is a graph that is undergoing a sequence of updates.

REMARKS

R1: In a typical dynamic graph problem, one would like to answer queries on dynamic graphs, for instance, whether the graph is connected, or which is the shortest path between any two vertices.

R2: The goal of a dynamic graph algorithm is to update efficiently the solution of a problem after dynamic changes, rather than having to recompute it from scratch each time. Given their powerful versatility, it is not surprising that dynamic algorithms and dynamic data structures are often more difficult to design and to analyze than their static counterparts.

DEFINITIONS

We can classify dynamic graph problems according to the types of updates allowed.

D3: A dynamic graph problem is said to be ***fully dynamic*** if the update operations include unrestricted insertions and deletions of edges or vertices.

D4: A dynamic graph problem is said to be ***partially dynamic*** if only one type of update, either insertions or deletions, is allowed.

D5: A dynamic graph problem is said to be ***incremental*** if only insertions are allowed.

D6: A dynamic graph problem is said to be ***decremental*** if only deletions are allowed.

REMARKS

R3: In the first part of this work we will present the main algorithmic techniques used to solve dynamic problems on *undirected* graphs. To illustrate those techniques, we will focus particularly on dynamic minimum spanning trees and on connectivity problems.

R4: In the second part of this work we will deal with dynamic problems on *directed* graphs, and we will investigate as paradigmatic problems the dynamic maintenance of transitive closure and shortest paths.

R5: Interestingly enough, dynamic problems on directed graphs seem much harder to solve than their counterparts on undirected graphs, and they require completely different techniques and tools.

PART 1: DYNAMIC PROBLEMS ON UNDIRECTED GRAPHS

This part considers fully dynamic algorithms for undirected graphs. These algorithms maintain efficiently some property of a graph that is undergoing structural changes defined by insertion and deletion of edges, and/or updates of edge costs. To check the graph property throughout a sequence of these updates, the algorithms must be prepared to answer queries on the graph property efficiently.

EXAMPLES

E1: The ***fully dynamic minimum spanning tree*** problem consists of maintaining a minimum spanning forest of a graph during insertions of edges, deletions of edges, and edge cost changes.

E2: A ***fully dynamic connectivity*** algorithm must be able to insert edges, delete edges, and answer a query on whether the graph is connected, or whether two vertices are in the same connected component.

REMARKS

R6: The goal of a dynamic algorithm is to minimize the amount of recomputation required after each update.

R7: All the dynamic algorithms that we describe are able to maintain dynamically the graph property at a cost (per update operation) which is significantly smaller than the cost of recomputing the graph property from scratch.

In this part, first we present general techniques and tools used in designing dynamic algorithms on undirected graphs (§10.2.1), and then we survey the fastest algorithms for solving two of the most fundamental graph problems: connectivity (§10.2.2) and minimum spanning trees (§10.2.3).

10.2.1 General Techniques for Undirected Graphs

Many of the algorithms proposed in the literature use the same general techniques, and hence we begin by describing these techniques. As a common theme, most of these techniques use some sort of graph decomposition, and they partition either the vertices or the edges of the graph to be maintained. Moreover, data structures that maintain properties of dynamically changing trees are often used as building blocks by many dynamic graph algorithms. The basic update operations are edge insertions and edge deletions. Many properties of dynamically changing trees have been considered in the literature.

EXAMPLES

E3: The basic query operation is tree membership: while the forest of trees is dynamically changing, we would like to know at any time which tree contains a given vertex, or whether two vertices are in the same tree. Dynamic tree membership is a special case of dynamic connectivity in undirected graphs, and indeed in §10.2.2 and in §10.2.3 we will see that some of the data structures presented here for trees are useful for solving the more general problem on graphs.

E4: Other properties have also been considered: the parent of a vertex, the least common ancestor of two vertices, and the center or the diameter of a tree [AlHoDeTh97, AlHoTh00,SlTa83]. When costs are associated either to vertices or to edges, one could also ask what is the minimum or maximum cost in a given path.

In what follows, we first present three different data structures that maintain properties of dynamically changing trees: topology trees, ET trees, and top trees. Next, we discuss techniques that can be applied on general undirected graphs: clustering, sparsification, and randomization. In the course of the presentation, we also highlight how these techniques have been applied to solve the fully dynamic connectivity and/or minimum spanning tree problems, and which update and query bounds can be achieved when they are deployed.

Topology Trees

Topology trees have been introduced by Frederickson [Fr85] to maintain dynamic trees upon insertions and deletions of edges.

DEFINITIONS

D7: Given a tree T of a forest, a ***cluster*** is a connected subgraph of T.

D8: The ***cardinality*** of a cluster is the number of its vertices.

D9: The ***external degree*** of a cluster is the number of tree edges incident to it.

D10: A ***topology tree*** is a hierarchical representation of a tree T of the forest: each level of the topology tree partitions the vertices of T into clusters. Clusters at level 0 contain one vertex each. Clusters at level $\ell \geq 1$ form a partition of the vertices of the tree T' obtained by shrinking each cluster at level $\ell - 1$ into a single vertex. The basic partition must be suitably chosen so that the topology tree has depth $O(\log n)$ and, so that during edge insertions and deletions, each level of the topology tree can be updated by applying only a few local adjustments.

ASSUMPTION

In order to illustrate the solution proposed by Frederickson [Fr85,Fr97], we assume that the tree T has maximum vertex degree 3: this is without loss of generality, since a standard transformation can be applied if this is not the case [Ha69].

DEFINITION

D11: A ***restricted partition*** of a tree T is a partition of its vertex set V into clusters of external degree ≤ 3 and cardinality ≤ 2 such that:

(1) Each cluster of external degree 3 has cardinality 1.

(2) Each cluster of external degree < 3 has cardinality at most 2.

(3) No two adjacent clusters can be combined and still satisfy the above.

An example of topology tree, together with the restricted partitions used to obtain its levels, is given in Figure 10.2.1.

REMARKS

R8: There can be several restricted partitions for a given tree T, based upon different choices of the vertices to be unioned.

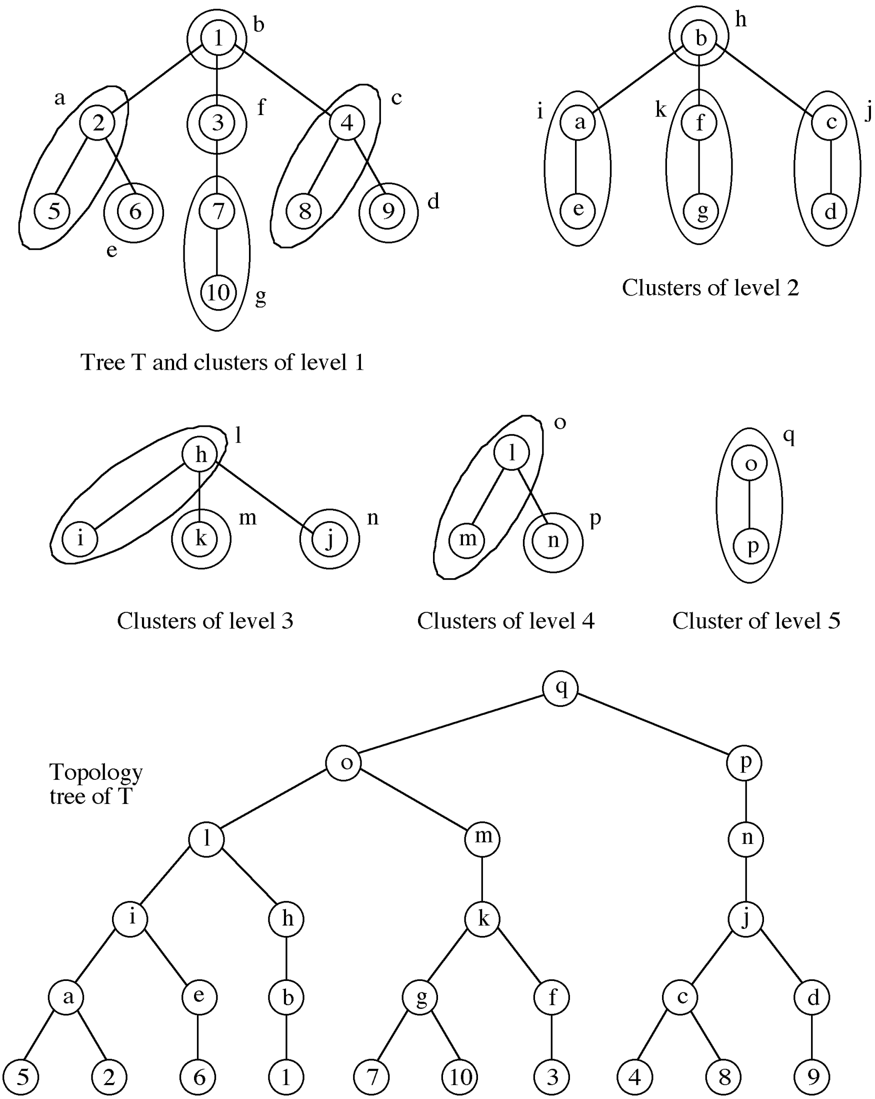
R9: Because of clause (3), the restricted partition implements a cluster-forming scheme according to a locally greedy heuristic, which does not always obtain the minimum number of clusters, but which has the advantage of requiring only local adjustments during updates.

APPROACH

Edge deletion. We sketch how to update the clusters of a restricted partition when an edge e is deleted from a tree T. First, removing e splits T into two trees, say T_1 and T_2, which inherit all of the clusters of T, possibly with the following exceptions.

◇ If e is entirely contained in a cluster, this cluster is no longer connected and therefore must be split. After the split, we must check whether each of the two resulting clusters is adjacent to a cluster of tree degree at most 2, and if these two adjacent clusters together have cardinality ≤ 2. If so, we combine these two clusters in order to maintain condition (3).

◇ If e is between two clusters, then no split is needed. However, since the tree degree of the clusters containing the endpoints of e has been decreased, we must check if each cluster should be combined with an adjacent cluster, again because of condition (3).



Tree T and clusters of level 1

Clusters of level 2

Clusters of level 3

Clusters of level 4

Cluster of level 5

Figure 10.2.1 **Restricted partitions and topology tree of a tree** T**.**

Edge insertion. Similar local manipulations can be applied to restore invariants (1) - (3) in Definition 11 in case of edge insertions.

Construction of the topology tree. The levels of the topology tree are built in a bottom up fashion by repeatedly applying the locally greedy heuristic.

Update of the topology tree. Each level can be updated upon insertions and deletions of edges in tree T by applying few locally greedy adjustments similar to the ones described before. In particular, a constant number of basic clusters (corresponding to leaves in the topology tree) are examined: the changes in these basic clusters percolate up in the topology tree, possibly causing vertex clusters to be regrouped in different ways.

FACTS

F1: The number of nodes at each level of the topology tree is a constant fraction of that at the previous level, and thus the number of levels is $O(\log n)$ (see [Fr85,Fr97]).

The fact that only a constant amount of work has to be done on $O(\log n)$ topology tree nodes implies a logarithmic bound on the update time.

F2: (Frederickson's Theorem) [Fr85] The update of a topology tree because of an edge insertion or deletion can be supported in $O(\log n)$ time.

ET Trees

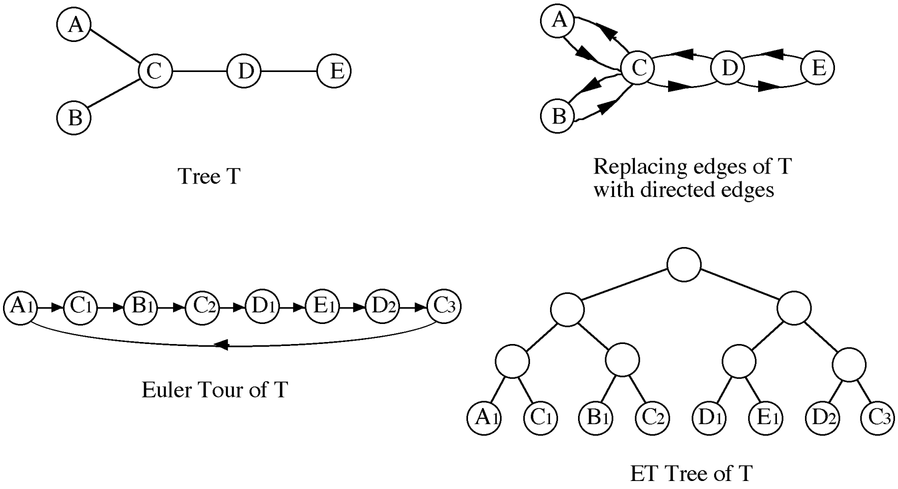
ET trees have been introduced by Henzinger and King [HeKi99] to work on dynamic forests whose vertices are associated with weighted or unweighted keys. Updates allow it to cut arbitrary edges, to insert edges linking different trees of the forest, and to add or remove the weighted key associated to a vertex. Supported queries are the following:

- ♢ `Connected`(u, v): tells whether vertices u and v are in the same tree.
- ♢ `Size`(v): returns the number of vertices in the tree that contains v.
- ♢ **`Minkey`**(v): returns a key of minimum weight in the tree that contains v; if keys are unweighted, an arbitrary key is returned.

DEFINITIONS

D12: An ***Euler tour*** of a tree T is a maximal closed walk over the graph obtained by replacing each edge of T by two directed edges with opposite direction. The walk traverses each edge exactly once; hence, if T has n vertices, the Euler tour has length $2n - 2$ (see Figure 10.2.2).

D13: An ***ET tree*** is a dynamic *balanced binary tree* (the number of nodes in the left and right subtrees of each node differs by at most one) over some Euler tour around T. Namely, leaves of the balanced binary tree are the nodes of the Euler Tour, in the same order in which they appear (see Figure 10.2.2).



Tree T

Replacing edges of T with directed edges

Euler Tour of T

ET Tree of T

Figure 10.2.2 Euler Tour and ET Tree of a tree T.

REMARK

R10: Although each vertex of T may occur several times in the Euler tour (an arbitrary occurrence is marked as *representative* of the vertex), an ET tree has $O(n)$ nodes.

APPROACH

Edge insertion and deletion. If trees in the forest are linked or cut, a constant number of splits and concatenations allow it to reconstruct the new Euler tour(s); the ET tree(s) can then be rebalanced by affecting only $O(\log n)$ nodes.

Connectivity queries. The query `Connected`(u, v) can be easily supported in $O(\log n)$ time by finding the roots of the ET trees containing u and v and checking if they coincide.

Size and Minkey queries. To support `Size` and `Minkey` queries, each node q of the ET tree maintains two additional values: the number $s(q)$ of representatives below it and the minimum weight key $k(q)$ attached to a representative below it. Such values can be maintained in $O(\log n)$ time per update and allow it to answer queries of the form `Size`(v) and `Minkey`(v) in $O(\log n)$ time for any vertex v of the forest: the root r of the ET tree containing v is found and values $s(r)$ and $k(r)$ are returned, respectively. See [HeKi99] for additional details of the method.

FACT

F3: Both updates and queries can be supported in $O(\log n)$ time using ET trees (see [HeKi99]).

Top Trees

Top trees have been introduced by Alstrup et al. [AlHoDeTh97] to maintain efficiently information about paths in trees, such as, e.g., the maximum weight on the path between any pair of vertices in a tree. The basic idea is taken from Frederickson's topology trees, but instead of partitioning vertices, top trees work by partitioning edges: the same vertex can then appear in more than one cluster.

DEFINITIONS

D14: Similarly to [Fr85,Fr97], a ***cluster*** is a connected subtree of tree T, with the additional constraint that at most two vertices, called ***boundary vertices***, have edges out of the subtree.

D15: Two clusters are said to be ***neighbors*** if their intersection contains exactly one vertex.

D16: A ***top tree*** of T is a binary tree such that:

◇ The leaves and the internal nodes represent edges and clusters of T, respectively.

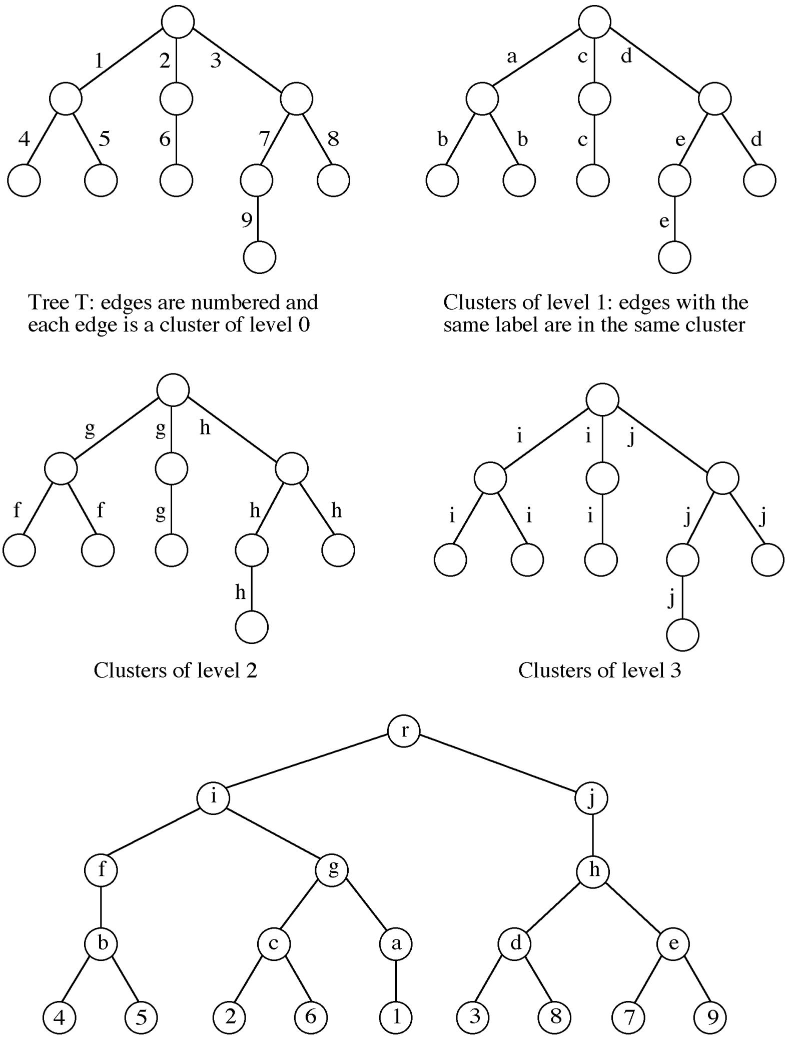
◇ The subtree represented by an internal node is the union of the subtrees represented by its two children, which must be neighbors.

◇ The root represents the entire tree T.

◇ The height is $O(\log n)$.

We refer to Figure 10.2.3 for an example of a top tree.

Top trees can be maintained under edge insert and delete operations in tree T by making use of two basic `Merge` and `Split` operations.



Tree T: edges are numbered and each edge is a cluster of level 0

Clusters of level 1: edges with the same label are in the same cluster

Clusters of level 2

Clusters of level 3

Top tree of T

Figure 10.2.3 Clusters and top tree of a tree T.

Merge. It takes two top trees whose roots are neighbor clusters and joins them to form a unique top tree.

Split. This is the reverse operation, deleting the root of a given top tree.

Edge insertion and deletion. The implementation of an edge insertion/deletion starts with a sequence of `Split` of all ancestor clusters of edges whose boundary changes and finishes with a sequence of `Merge`. Since an end-point v of an edge has to be already boundary vertex of the edge if v is not a leaf, each edge insert/delete can change the boundary of at most two edges, excluding the edge being inserted/deleted.

FACT

F4: [AlHoDeTh97] For a dynamic forest we can maintain top trees of height $O(\log n)$ supporting edge insertions and deletions with a sequence of $O(\log n)$ `Split` and `Merge`. The sequence itself is identified in $O(\log n)$ time.

REMARKS

R11: Top trees are typically used by attaching extra information to their nodes. A careful choice of the extra information makes it possible to maintain easily path properties of trees, such as the maximum weight of an edge in the unique path between any two vertices. See [AlHoDeTh97,AlHoTh00,HoDeTh01] for sample applications.

R12: Top trees are a natural generalization of standard balanced binary trees over dynamic collections of lists that may be concatenated and split, where each node of the balanced binary tree represents a segment of a list. In the terminology of top trees, this is just a special case of a cluster.

Clustering

The clustering technique of [Fr85] is based upon partitioning the graph into a suitable collection of *clusters*, such that each update involves only a small number of such clusters.

REMARKS

R13: Typically, the decomposition defined by the clusters is applied recursively, and the information about the subgraphs is combined with the topology trees described above.

R14: A refinement of the clustering technique appears in the idea of *ambivalent data structures* [Fr97], in which edges can belong to multiple groups, only one of which is actually selected depending on the topology of the given spanning tree.

EXAMPLE

E5: We briefly describe the application of clustering to the problem of maintaining a minimum spanning forest [Fr85]. Let $G = (V, E)$ be a graph with a designated spanning tree S. Clustering is used for partitioning the vertex set V into subtrees connected in S, so that each subtree is only adjacent to a few other subtrees. A topology tree is then used for representing a recursive partition of the tree S. Finally, a generalization of topology trees, called *2-dimensional topology trees*, is formed from pairs of nodes in the topology tree and allows it to maintain information about the edges in $E \setminus S$ [Fr85].

FACTS

F5: Fully dynamic algorithms based only on a single level of clustering obtain typically time bounds of the order of $O(m^{2/3})$ (see for instance [GaIt92,Ra95]).

When the partition can be applied recursively, better $O(m^{1/2})$ time bounds can be achieved by using 2-dimensional topology trees (see, for instance, [Fr85,Fr97]).

F6: (Frederickson's theorem) [Fr85] The minimum spanning forest of an undirected graph can be maintained in time $O(m^{1/2})$ per update, where m is the current number of edges in the graph.

REMARKS

R15: See [Fr85,Fr97] for details about Frederickson's algorithm. With the same technique, an $O(m^{1/2})$ time bound can be obtained also for fully dynamic connectivity and 2-edge connectivity [Fr85,Fr97].

R16: The type of clustering used can be very problem-dependent, however, which makes this technique difficult to be used as a black box.

Sparsification

Sparsification is a general technique due to Eppstein, et al. [EpGaItNi97] that can be used as a black box (without having to know the internal details), in order to design and dynamize graph algorithms. It is a divide-and-conquer technique that allows it to reduce the dependence on the number of edges in a graph, so that the time bounds for maintaining some property of the graph match the times for computing in sparse graphs. More precisely, when the technique is applicable, it speeds up a $T(n, m)$ time bound for a graph with n vertices and m edges to $T(n, O(n))$, i.e., to the time needed if the graph were sparse. For instance, if $T(n, m) = O(m^{1/2})$, we get a better bound of $O(n^{1/2})$. The technique itself is quite simple. A key concept is the notion of certificate.

DEFINITIONS

D17: For any graph property P and graph G, a ***certificate*** for G is a graph G' such that G has property P if and only if G' has the property.

D18: A subgraph on n vertices is ***sparse*** if it has $O(n)$ edges.

D19: A time bound $T(n)$ is ***well-behaved*** if, for some $c < 1$, $T(n/2) < cT(n)$. Well-behavedness eliminates strange situations in which a time bound fluctuates wildly with n. For instance, all polynomials are well-behaved.

APPROACH

Let G be a graph with m edges and n vertices. We partition the edges of G into a collection of $O(m/n)$ sparse subgraphs. The information relevant for each subgraph can be summarized in a sparse certificate. Certificates are then merged in pairs, producing larger subgraphs which are made sparse by again computing their certificate. The result is a balanced binary tree in which each node is represented by a sparse certificate. Each update involves $O(\log(m/n))$ graphs with $O(n)$ edges each, instead of one graph with m edges.

NOTATION

In the present context, $\log x$ stands for $\max(1, \log_2 x)$, so that $\log(m/n)$ is never smaller than 1, even if $m < 2n$.

REMARKS

There exist two variants of sparsification.

R17: The first variant is used in situations where no previous fully dynamic algorithm is known. A static algorithm is used for recomputing a sparse certificate in each tree node affected by an edge update. If the certificates can be found in time $O(m+n)$, this variant gives time bounds of $O(n)$ per update.

R18: In the second variant, certificates are maintained using a dynamic data structure. For this to work, a *stability* property of certificates is needed, to ensure that a small change in the input graph does not lead to a large change in the certificates. (We refer the interested reader to [EpGaItNi97] for a precise definition of stability.) This variant transforms time bounds of the form $O(m^p)$ into $O(n^p)$.

FACTS

F7: [EpGaItNi97] Let P be a property for which we can find sparse certificates in time $f(n,m)$ for some well-behaved f, and such that we can construct a data structure for testing property P in time $g(n,m)$ which can answer queries in time $q(n,m)$. Then there is a fully dynamic data structure for testing whether a graph has property P, for which edge insertions and deletions can be performed in time $O(f(n,O(n)))+g(n,O(n))$, and for which the query time is $q(n,O(n))$.

F8: [EpGaItNi97] Let P be a property for which stable sparse certificates can be maintained in time $f(n,m)$ per update, where f is well-behaved, and for which there is a data structure for property P with update time $g(n,m)$ and query time $q(n,m)$. Then P can be maintained in time $O(f(n,O(n)))+g(n,O(n))$ per update, with query time $q(n,O(n))$.

REMARKS

R19: Basically, the first version of sparsification (Fact 7) can be used to dynamize static algorithms, in which case we only need to *compute* efficiently *sparse* certificates, while the second version (Fact 8) can be used to speed up existing fully dynamic algorithms, in which case we need to *maintain* efficiently *stable sparse* certificates.

R20: Sparsification applies to a wide variety of dynamic graph problems, including minimum spanning forests, edge-connectivity, and vertex-connectivity. As an example, for the fully dynamic minimum spanning tree problem, it reduces the update time from $O(m^{1/2})$ [Fr85,Fr97] to $O(n^{1/2})$ [EpGaItNi97].

R21: Since sparsification works on top of a given algorithm, we need not know the internal details of this algorithm. Consequently, it can be applied orthogonally to other data structuring techniques: in a large number of situations both clustering and sparsification have been combined to produce an efficient dynamic graph algorithm.

Randomization

Clustering and sparsification allow one to design efficient deterministic algorithms for fully dynamic problems. The last technique we present in this section is due to Henzinger and King [HeKi99]; it achieves faster update times for some problems by exploiting the power of randomization.

APPROACH

We sketch how the randomization technique works, taking the fully dynamic connectivity problem as an example. In a graph $G=(V,E)$ to be maintained dynamically, the edges of a spanning forest F are called *tree edges*, and the edges in $E \setminus F$ are called *non-tree edges.*

Maintaining spanning forests. Trees in the spanning forests are maintained using the Euler Tours data structure (ET trees) described above, which allows one to obtain logarithmic updates and queries within the forest.

Random sampling. A key idea behind the technique of Henzinger and King is the following: when e is deleted from a tree T, use random sampling among the non-tree edges incident to T, in order to find quickly a replacement edge for e, if any.

Graph decomposition. The second key idea is to combine randomization with a suitable graph decomposition. We maintain an edge decomposition of the current graph G into $O(\log n)$ edge disjoint subgraphs $G_i = (V, E_i)$. These subgraphs are hierarchically ordered. The lower levels contain tightly-connected portions of G (i.e., dense edge cuts), while the higher levels contain loosely-connected portions of G (i.e., sparse cuts). For each level i, a spanning forest for the graph defined by all the edges in levels i or below is also maintained.

REMARKS

R22: Note that the hard operation is the deletion of a tree edge: indeed, a spanning forest is easily maintained throughout edge insertions, and deleting a non-tree edge does not change the forest.

R23: The goal is an update time of $O(\log^3 n)$: after an edge deletion, in the quest for a replacement edge, we can afford a number of sampled edges of $O(\log^2 n)$. However, if the candidate set of edge e is a small fraction of all non-tree edges which are adjacent to T, it is unlikely to find a replacement edge for e among this small sample. If we found no candidate among the sampled edges, we must check explicitly all the non-tree edges adjacent to T. After random sampling has failed to produce a replacement edge, we need to perform this check explicitly, otherwise we would not be guaranteed to provide correct answers to the queries.

R24: Since there might be numerous edges adjacent to T, this explicit check could be an expensive operation, so it should be made a low probability event for the randomized algorithm. This can produce pathological updates, however, since deleting all edges in a relatively small candidate set, reinserting them, deleting them again, and so on will almost surely produce many of those unfortunate events.

R25: The graph decomposition is used to prevent the undesirable behavior described above. If a spanning forest edge e is deleted from a tree at some level i, random sampling is used to quickly find a replacement for e at that level. If random sampling succeeds, the tree is reconnected at level i. If random sampling fails, the edges that can replace e in level i form with high probability a sparse cut. These edges are moved to level $i + 1$ and the same procedure is applied recursively on level $i + 1$.

FACT

F9: (Henzinger and King's Theorem) [HeKi99] Let G be a graph with n vertices and m_0 edges, subject to edge deletions only. A spanning forest of G can be maintained in $O(\log^3 n)$ expected amortized time per deletion, if there are at least $\Omega(m_0)$ deletions. The time per query is $O(\log n)$.

10.2.2 Connectivity

We now give a high level description of the fastest deterministic algorithm for the fully dynamic connectivity problem in undirected graphs [HoDeTh01]: the algorithm answers connectivity queries in $O(\log n/\log\log n)$ worst-case running time while supporting edge insertions and deletions in $O(\log^2 n)$ amortized time. Like the randomized algorithm in [HeKi99], the deterministic algorithm in [HoDeTh01] maintains a spanning forest F of the dynamically changing graph G.

FACTS

F10: Let e be a tree edge of forest F, and let T be the tree of F containing it. When e is deleted, the two trees T_1 and T_2 obtained from T after the deletion of e can be reconnected if and only if there is a non-tree edge in G with one endpoint in T_1 and the other endpoint in T_2. We call such an edge a *replacement edge* for e. In other words, if there is a replacement edge for e, then T is reconnected via this replacement edge; otherwise, the deletion of e creates a new connected component in G.

F11: To accommodate systematic search for replacement edges, the algorithm associates to each edge e a level $\ell(e)$ and, based on edge levels, maintains a set of sub-forests of the spanning forest F: for each level i, forest F_i is the sub-forest induced by tree edges of level $\geq i$.

F12: $F = F_0 \supseteq F_1 \supseteq F_2 \supseteq \ldots \supseteq F_L$, where L denotes the maximum edge level.

F13: Initially, all edges have level 0; levels are then progressively increased, but never decreased. The changes of edge levels are accomplished so as to maintain the following invariants, which obviously hold at the beginning.

INVARIANTS

Invariant (1): F is a maximum spanning forest of G if edge levels are regarded as weights.

Invariant (2): The number of nodes in each tree of F_i is at most $n/2^i$.

REMARKS

R26: Invariant (1) should be interpreted as follows. Let (u, v) be a non-tree edge of level $\ell(u, v)$, and let $u \cdots v$ be the unique path between u and v in F (such a path exists since F is a spanning forest of G). Let e be any edge in $u \cdots v$ and let $\ell(e)$ be its level. Due to invariant (1), $\ell(e) \geq \ell(u, v)$. Since this holds for each edge in the path, and by construction $F_{\ell(u,v)}$ contains all the tree edges of level $\geq \ell(u, v)$, the entire path is contained in $F_{\ell(u,v)}$, i.e., u and v are connected in $F_{\ell(u,v)}$.

R27: Invariant (2) implies that the maximum number of levels is $L \leq \lfloor \log_2 n \rfloor$.

FACTS

F14: When a new edge is inserted, it is given level 0. Its level can be then increased at most $\lfloor \log_2 n \rfloor$ times as a consequence of edge deletions.

F15: When a tree edge $e = (v, w)$ of level $\ell(e)$ is deleted, the algorithm looks for a replacement edge at the highest possible level, if any. Due to invariant (1), such a replacement edge has level $\ell \leq \ell(e)$. Hence, a replacement subroutine `Replace(`(u, w)`,`$\ell(e)$`)` is called with parameters e and $\ell(e)$. We now sketch the operations performed by this subroutine.

F16: `Replace(`(u, w)`,`ℓ`)` finds a replacement edge of the highest level $\leq \ell$, if any. If such a replacement does not exist in level ℓ, we have two cases: if $\ell > 0$, we recurse on level $\ell - 1$; otherwise, $\ell = 0$, and we can conclude that the deletion of (v, w) disconnects v and w in G.

F17: During the search at level ℓ, suitably chosen tree and non-tree edges may be promoted at higher levels as follows. Let T_v and T_w be the trees of forest F_ℓ obtained after deleting (v, w) and let, w.l.o.g., T_v be smaller than T_w. Then T_v contains at most $n/2^{\ell+1}$ vertices, since $T_v \cup T_w \cup \{(v, w)\}$ was a tree at level ℓ and due to invariant (2). Thus, edges in T_v of level ℓ can be promoted at level $\ell + 1$ by maintaining the invariants. Non-tree edges incident to T_v are finally visited one by one: if an edge does connect T_v and T_w, a replacement edge has been found and the search stops, otherwise its level is increased by 1.

F18: We maintain an ET-tree, as described in §10.2.1, for each tree of each forest. Consequently, all the basic operations needed to implement edge insertions and deletions can be supported in $O(\log n)$ time.

F19: [HoDeTh01] A dynamic graph G with n vertices can be maintained upon insertions and deletions of edges using $O(\log^2 n)$ amortized time per update and answering connectivity queries in $O(\log n / \log \log n)$ worst-case running time.

REMARKS

R28: In addition to inserting and deleting edges from a forest, ET-trees must also support operations such as finding the tree of a forest that contains a given vertex, computing the size of a tree, and, more importantly, finding tree edges of level ℓ in T_v and non-tree edges of level ℓ incident to T_v. This can be done by augmenting the ET-trees with a constant amount of information per node: see [HoDeTh01] for details.

R29: Using an amortization argument based on level changes, the claimed $O(\log^2 n)$ bound on the update time can be finally proved. Namely, inserting an edge costs $O(\log n)$, as well as increasing its level. Since this can happen $O(\log n)$ times, the total amortized insertion cost, inclusive of level increases, is $O(\log^2 n)$. With respect to edge deletions, cutting and linking $O(\log n)$ forest has a total cost $O(\log^2 n)$; moreover, there are $O(\log n)$ recursive calls to `Replace`, each of cost $O(\log n)$ plus the cost amortized over level increases. The ET-trees over $F_0 = F$ allows it to answer connectivity queries in $O(\log n)$ worst-case time. As shown in [HoDeTh01], this can be reduced to $O(\log n / \log \log n)$ by using a $\Theta(\log n)$-ary version of ET-trees.

10.2.3 Minimum Spanning Trees

A few simple changes to the connectivity algorithm presented in §10.2.2 are sufficient to maintain a minimum spanning forest of a weighted undirected graph upon deletions of edges [HoDeTh01]. A general reduction from [HeKi97] can then be applied to make the deletions-only algorithm fully dynamic.

Decremental Minimum Spanning Tree

APPROACH

In addition to starting from a *minimum* spanning forest, the only change concerns the function `Replace`, which should be implemented so as to consider candidate replacement edges of level ℓ in order of increasing weight, and not in arbitrary order. To do so, the ET-trees from §10.2.1 can be augmented so that each node maintains the minimum weight of a non-tree edge incident to the Euler tour segment below it. All the operations can still be supported in $O(\log n)$ time, yielding the same time bounds as for connectivity.

We now discuss the correctness of the algorithm. In particular, function `Replace` returns a replacement edge of minimum weight on the highest possible level: it is not immediate that such a replacement edge has the minimum weight among all levels. This can be proved by first showing that the following invariant, proved in [HoDeTh01], is maintained by the algorithm.

INVARIANT

Invariant (3): Every cycle $\mathcal{C}$ has a non-tree edge of maximum weight and minimum level among all the edges in $\mathcal{C}$.

FACTS

F20: Invariant (3) can be used to prove that, among all the replacement edges, the lightest edge is on the maximum level. Let e_1 and e_2 be two replacement edges with $w(e_1) < w(e_2)$, and let $\mathcal{C}_i$ be the cycle induced by e_i in F, $i = 1, 2$. Since F is a minimum spanning forest, e_i has maximum weight among all the edges in $\mathcal{C}_i$. In particular, since by hypothesis $w(e_1) < w(e_2)$, e_2 is also the heaviest edge in cycle $\mathcal{C} = (\mathcal{C}_1 \cup \mathcal{C}_2) \setminus (\mathcal{C}_1 \cap \mathcal{C}_2)$. Thanks to Invariant (3), e_2 has minimum level in $\mathcal{C}$, proving that $\ell(e_2) \leq \ell(e_1)$. Thus, considering non-tree edges from higher to lower levels is correct.

F21: [HoDeTh01] There exists a deletions-only minimum spanning forest algorithm that can be initialized on a graph with n vertices and m edges and supports any sequence of edge deletions in $O(m \log^2 n)$ total time.

Fully Dynamic Minimum Spanning Tree

The reduction used to obtain a fully dynamic algorithm is a slight generalization of the construction proposed by Henzinger and King [HeKi97] and works as follows.

FACTS

F22: (HeKi97,HoDeTh01) Suppose we have a deletions-only minimum spanning tree algorithm that, for any k and l, can be initialized on a graph with k vertices and l edges and supports any sequence of $\Omega(l)$ deletions in total time $O(l \cdot t(k, l))$, where t is a non-decreasing function. Then there exists a fully-dynamic minimum spanning tree algorithm for a graph with n nodes starting with no edges, that, for m edges, supports updates in time

$$O\left(\log^3 n + \sum_{i=1}^{3+\log_2 m} \sum_{j=1}^{i} t(min\{n, 2^j\}, 2^j)\right)$$

See [HeKi97] and [HoDeTh01] for description of the construction that proves Fact 22. From Fact 21 we get $t(k, l) = O(\log^2 k)$. Hence, by combining Fact 21 and Fact 22, we get the claimed result.

F23: [HoDeTh01] There exists a fully-dynamic minimum spanning forest algorithm that, for a graph with n vertices, starting with no edges, maintains a minimum spanning forest in $O(\log^4 n)$ amortized time per edge insertion or deletion.

PART 2: DYNAMIC PROBLEMS ON DIRECTED GRAPHS

In this part we survey the newest results for dynamic problems on directed graphs. In particular, we focus on two of the most fundamental problems: transitive closure and shortest paths. These problems play a crucial role in many applications, including network optimization and routing, traffic information systems, databases, compilers, garbage collection, interactive verification systems, industrial robotics, dataflow analysis, and document formatting.

We first present general techniques and tools used in designing dynamic path problems on directed graphs (§ 10.2.4), and then we address the newest results for dynamic transitive closure and dynamic shortest paths (§ 10.2.5 and § 10.2.6, respectively). In the first problem, the goal is to maintain reachability information in a directed graph subject to insertions and deletions of edges. The fastest known algorithms support graph updates in quadratic (or near-quadratic) time and reachability queries in constant time [DeIt00,Ki99]. In the second problem, we wish to maintain information about shortest paths in a directed graph subject to insertion and deletion of edges, or updates of edge weights. Similarly to dynamic transitive closure, this can be done in near-quadratic time per update and optimal time per query [DeIt03].

10.2.4 General Techniques for Directed Graphs

In this subsection we discuss the main techniques used to solve dynamic path problems on directed graphs. We first address combinatorial and algebraic properties, and then we consider some efficient data structures, which are used as building blocks in designing dynamic algorithms for transitive closure and shortest paths.

Path Problems and Kleene Closures

Path problems such as transitive closure and shortest paths are tightly related to matrix sum and matrix multiplication over a closed semiring (see [CoLeRiSt01] for more details).

NOTATION: The usual sum and multiplication operations over Boolean matrices are denoted by $+$ and $\cdot$, respectively.

NOTATION: Given two real-valued matrices A and B, $C = A \odot B$ is the matrix product such that $C[x,y] = \min_{1 \le z \le n}\{A[x,z] + B[z,y]\}$, and $D = A \oplus B$ is the matrix sum such that $D[x,y] = \min\{A[x,y], B[x,y]\}$.

NOTATION: We also denote by AB the product $A \odot B$ and by $AB[x,y]$ entry (x,y) of matrix AB.

FACTS

F24: Let $G = (V,E)$ be a directed graph and let $TC(G)$ be the (reflexive) transitive closure of G. If X is the Boolean adjacency matrix of G, then the Boolean adjacency matrix of $TC(G)$ is the Kleene closure of X on the $\{+, \cdot, 0, 1\}$ Boolean semiring:

$$X^* = \sum_{i=0}^{n-1} X^i$$

F25: Let $G = (V,E)$ be a weighted directed graph with no negative-length cycles. If X is a weight matrix such that $X[x,y]$ is the weight of edge (x,y) in G, then the distance matrix of G is the Kleene closure of X on the $\{\oplus, \odot, \mathcal{R}\}$ semiring:

$$X^* = \bigoplus_{i=0}^{n-1} X^i$$

The next two facts recall two well-known methods for computing the Kleene closure X^* of an $n \times n$ matrix X.

Logarithmic Decomposition. A simple method to compute X^*, based on repeated squaring, requires $O(n^\mu \cdot \log n)$ worst-case time, where $O(n^\mu)$ is the time required for computing the product of two matrices over a closed semiring.

F26: This method performs $\log_2 n$ sums and products of the form $X_{i+1} = X_i + X_i^2$, where $X = X_0$ and $X^* = X_{\log_2 n}$.

Recursive Decomposition. Another method, due to Munro [Mu71], is based on a Divide and Conquer strategy and computes X^* in $O(n^\mu)$ worst-case time.

F27: Munro observed that, if we partition a matrix X into four submatrices A, B, D, C of size $n/2 \times n/2$ (considered in clockwise order), and the closure X^* similarly into four submatrices E, F, H, G of size $n/2 \times n/2$, then X^* is definable recursively according to the following equations:

$$E = (A + BD^*C)^*$$

$$F = EBD^*$$

$$G = D^*CE$$

$$H = D^* + D^*CEBD^*$$

Surprisingly, using this decomposition the cost of computing X^* starting from X is asymptotically the same as the cost of multiplying two matrices over a closed semiring.

Uniform Paths

Some combinatorial properties of shortest paths in directed graphs have been recently discovered by Demetrescu and Italiano [DeIt03]. In particular, we consider shortest paths as a special case of a broader class of paths called *uniform paths*. To characterize how uniform paths change in a fully dynamic graph, we consider the notions of *historical shortest path* and *potentially uniform path*.

DEFINITIONS

D20: A path π in a graph is *uniform* if every proper subpath of π is a shortest path.

D21: A *historical shortest path* is a path that has been a shortest path at some point during the sequence of updates, and none of its edges have been updated since then.

Using this notion we can define a superset of uniform paths that are called *potentially uniform paths.*

D22: A path π in a graph is *potentially uniform* if every proper subpath of π is a historical shortest path.

REMARKS

R30: As an alternative equivalent definition, a path π_{xy} is uniform in a graph if every edge (u, v) in π_{xy} satisfies the relation $d_{xu} + w_{uv} + d_{vy} = w(\pi_{xy})$, where d_{xy} denotes the distance between vertex x and vertex y in the graph, w_{uv} is the weight of edge (u, v), and $w(\pi_{xy})$ is the weight of π_{xy}. This accounts for the terminology used in Definition D20.

R31: It is not difficult to prove that the amortized number of uniform paths that may change due to an edge weight update is $O(n^2)$ if updates are partially dynamic, i.e., increase-only or decrease-only.

FACTS

F28: (Demetrescu and Italiano [DeIt03]) If we denote by SP, UP, HSP, and PUP respectively the sets of shortest paths, uniform paths, historical shortest paths, and potentially uniform paths in a graph, then at any time the following inclusions hold: $SP \subseteq UP \subseteq PUP$ and $SP \subseteq HSP \subseteq PUP$.

F29: (Demetrescu and Italiano [DeIt03]) Let G be a graph subject to a sequence of update operations. If at any time throughout the sequence of updates there are at most $O(z)$ historical shortest paths between each pair of vertices, then the amortized number of paths that become potentially uniform at each update is $O(zn^2)$.

REMARKS

R32: Potentially uniform paths exhibit strong combinatorial properties in graphs subject to (fully) dynamic updates. In particular, it is possible to prove that the number of paths that become potentially uniform in a graph at each edge weight update depends on the number of historical shortest paths in the graph.

R33: To keep changes in potentially uniform paths small, it is then desirable to have as few historical shortest paths as possible. Indeed, it is possible to transform every update sequence into a slightly longer equivalent sequence that generates only a few historical shortest paths. In particular, there exists a simple *smoothing* strategy that, given any update sequence Σ of length k, produces an operationally equivalent sequence $F(\Sigma)$ of length $O(k \log k)$ that yields only $O(\log k)$ historical shortest paths between each pair of vertices in the graph. We refer the interested reader to [DeIt03] for a detailed description of this smoothing strategy. According to Fact 29, this technique implies that only $O(n^2 \log k)$ potentially uniform paths change at each edge weight update in the smoothed sequence $F(\Sigma)$.

R34: As elaborated in [DeIt03], potentially uniform paths can be maintained very efficiently. Since by Fact 28 potentially uniform paths include shortest paths, this yields the fastest known algorithm for fully dynamic all pairs shortest paths.

Long Paths Property

If we pick a subset S of vertices at random from a graph G, then a sufficiently long path will intersect S with high probability. This can be very useful in finding a long path by using short searches. To the best of our knowledge, the long paths property was first given in [GrKn82], and later on it has been used may times in designing efficient algorithms for transitive closure and shortest paths (see e.g., [DeIt01,Ki99,UlYa91,Zw98]).

FACT

F30: (Ullman and Yannakakis [UlYa91]) Let $S \subseteq V$ be a set of vertices chosen uniformly at random. Then the probability that a given simple path has a sequence of more than $(cn \log n)/|S|$ vertices, none of which are from S, for any $c > 0$, is, for sufficiently large n, bounded by $2^{-\alpha c}$ for some positive α.

REMARK

R35: As shown in [Zw98], it is possible to choose set S deterministically by a reduction to a hitting set problem [Ch79,Lo75]. A similar technique has also been used in [Ki99].

Reachability Trees

A special tree data structure has been widely used to solve dynamic path problems on directed graphs. The first appearance of this tool dates back to 1981, when Even and Shiloach showed how to maintain a breadth-first tree of an undirected graph under any sequence of edge deletions [EvSh81]; they used this as a kernel for decremental connectivity on undirected graphs. Later on, Henzinger and King [HeKi99] showed how to adapt this data structure to fully dynamic transitive closure in directed graphs. King [Ki99] designed an extension of this tree data structure to weighted directed graphs for solving fully dynamic all pairs shortest paths.

PROBLEM

In the unweighted directed version, the goal is to maintain information about breadth-first search (BFS) on a directed graph G undergoing deletions of edges. In particular, in the context of dynamic path problems, we are interested in maintaining BFS trees of depth up to d, with $d \leq n$. Given a directed graph $G = (V, E)$ and a vertex $r \in V$, we would like to support any intermixed sequence of the following operations:

`Delete`(x, y): delete edge (x, y) from G.

`Level`(u): return the level of vertex u in the BFS tree of depth d rooted at r (return $+\infty$ if u is not reachable from r within distance d).

FACT

F31: (King [Ki99]) Maintaining BFS levels up to depth d from a given root requires $O(md)$ time in the worst case throughout any sequence of edge deletions in a directed graph with m initial edges.

REMARKS

R36: Fact 31 means that maintaining BFS levels requires d times the time needed for constructing them. Since $d \leq n$, we obtain a total bound of $O(mn)$ if there are no limits on the depth of the BFS levels.

R37: As it was shown in [HeKi99,Ki99], it is possible to extend the BFS data structure presented in this section to deal with weighted directed graphs. In this case, a shortest path tree is maintained in place of BFS levels: after each edge deletion or edge weight increase, the tree is reconnected by essentially mimicking Dijkstra's algorithm rather than BFS. Details can be found in [Ki99].

Matrix Data Structures

We now consider matrix data structures for keeping information about paths in dynamic directed graphs. As we have seen above (Path Problems and Kleene Closures), Kleene closures can be constructed by evaluating polynomials over matrices. It is therefore natural to consider data structures for maintaining polynomials of matrices subject to updates of entries, like the one introduced in [DeIt00].

PROBLEM

In the case of Boolean matrices, the problem can be stated as follows. Let P be a polynomial over $n \times n$ Boolean matrices with constant degree, constant number of terms, and variables $X_1 \ldots X_k$. We wish to maintain a data structure for P subject to any intermixed sequence of update and query operations of the following kind:

`SetRow`$(i, \Delta X, X_b)$: sets to one the entries in the i-th row of variable X_b of polynomial P corresponding to one-valued entries in the i-th row of matrix ΔX.

`SetCol`$(i, \Delta X, X_b)$: sets to one the entries in the i-th column of variable X_b of polynomial P corresponding to one-valued entries in the i-th column of matrix ΔX.

`Reset`$(\Delta X, X_b)$: resets to zero the entries of variable X_b of polynomial P corresponding to one-valued entries in matrix ΔX.

`Lookup`$()$: returns the maintained value of P.

We add to the previous four operations a further update operation especially designed for maintaining path problems:

`LazySet`$(\Delta X, X_b)$: sets to 1 the entries of variable X_b of P corresponding to one-valued entries in matrix ΔX. However, the maintained value of P might not be immediately affected by this operation.

REMARK

R38: Let C_P be the correct value of P that we would have by recomputing it from scratch after each update, and let M_P be the actual value that we maintain. If no `LazySet` operation is ever performed, then always $M_P = C_P$. Otherwise, M_P is not necessarily equal to C_P, and we guarantee the following weaker property on M_P: if $C_P[u, v]$ flips from 0 to 1 due to a `SetRow`/`SetCol` operation on a variable X_b, then $M_P[u, v]$ flips from 0 to 1 as well. This means that `SetRow` and `SetCol` always correctly reveal new 1's in the maintained value of P, possibly taking into account the 1's inserted through previous `LazySet` operations. This property is crucial for dynamic path problems.

FACTS

F32: (Demetrescu and Italiano [DeIt00]) Let P be a polynomial with constant degree of matrices over the Boolean semiring. Any `SetRow`, `SetCol`, `LazySet`, and `Reset` operation on a polynomial P can be supported in $O(n^2)$ amortized time. `Lookup` queries are answered in optimal time.

F33: (Demetrescu and Italiano [DeIt01]) Let P be a polynomial with constant degree of matrices over the $\{\min, +\}$ semiring. Any `SetRow`, `SetCol`, `LazySet`, and `Reset` operation on variables of P can be supported in $O(D \cdot n^2)$ amortized time, where D is the maximum number of different values assumed by entries of variables during the sequence of operations. `Lookup` queries are answered in optimal time.

10.2.5 Dynamic Transitive Closure

In this subsection we survey the best known algorithms for fully dynamic transitive closure. Given a directed graph G with n vertices and m edges, the problem consists of supporting any intermixed sequence of operations of the following kind:

`Insert`(u, v): insert edge (u, v) in G;

`Delete`(u, v): delete edge (u, v) from G;

`Query`(x, y): answer a reachability query by returning "yes" if there is a path from vertex x to vertex y in G, and "no" otherwise;

FACTS

F34: A simple-minded solution to this problem consists of maintaining the graph under insertions and deletions, searching if y is reachable from x at any query operation. This yields $O(1)$ time per update (`Insert` and `Delete`), and $O(m)$ time per query, where m is the current number of edges in the maintained graph.

F35: Another simple-minded solution would be to maintain the Kleene closure of the adjacency matrix of the graph, rebuilding it from scratch after each update operation. Using the recursive decomposition of Munro [Mu71] discussed in §10.2.4 (Path Problems and Kleene Closures) and fast matrix multiplication [CoWi90], this takes constant time per reachability query and $O(n^\omega)$ time per update, where $\omega < 2.38$ is the current best exponent for matrix multiplication.

REMARKS

R39: Despite many years of research in this topic, no better solution to this problem was known until 1995, when Henzinger and King [HeKi99] proposed a randomized Monte Carlo algorithm with one-sided error supporting a query time of $O(n/\log n)$ and an amortized update time of $O(n\hat{m}^{0.58}\log^2 n)$, where $\hat{m}$ is the average number of edges in the graph throughout the whole update sequence. Since $\hat{m}$ can be as high as $O(n^2)$, their update time is $O(n^{2.16}\log^2 n)$.

R40: Khanna, Motwani and Wilson [KhMoWi96] proved that, when a lookahead of $\Theta(n^{0.18})$ in the updates is permitted, a deterministic update bound of $O(n^{2.18})$ can be achieved.

R41: King and Sagert [KiSa99] showed how to support queries in $O(1)$ time and updates in $O(n^{2.26})$ time for general directed graphs and $O(n^2)$ time for directed acyclic graphs; their algorithm is randomized with one-sided error. These bounds were further improved by King [Ki99], who exhibited a deterministic algorithm on general digraphs with $O(1)$ query time and $O(n^2 \log n)$ amortized time per update operations, where updates are insertions of a set of edges incident to the same vertex and deletions of an arbitrary subset of edges.

R42: Using a different framework, Demetrescu and Italiano [DeIt00] obtained a deterministic fully dynamic algorithm that achieves $O(n^2)$ amortized time per update for general directed graphs.

R43: We note that each update might change a portion of the transitive closure as large as $\Omega(n^2)$. Thus, if the transitive closure has to be maintained explicitly after each update so that queries can be answered with one lookup, $O(n^2)$ is the best update bound one could hope for.

R44: If one is willing to pay more for queries, Demetrescu and Italiano [DeIt00] showed how to break the $O(n^2)$ barrier on the single-operation complexity of fully dynamic transitive closure: building on a previous path counting technique introduced by King and Sagert [KiSa99], they devised a randomized algorithm with one-sided error for directed acyclic graphs that achieves $O(n^{1.58})$ worst-case time per update and $O(n^{0.58})$ worst-case time per query.

R45: Other recent results for dynamic transitive closure appear in [RoZw02].

King's $\mathbf{O(n^2 \log n)}$ Update Algorithm

King [Ki99] devised the first deterministic near-quadratic update algorithm for fully dynamic transitive closure. The algorithm is based on the tree data structure considered in §10.2.4 (Reachability Trees) and on the logarithmic decomposition discussed in §10.2.4 (Path Problems and Kleene Closures). It maintains explicitly the transitive closure of a graph G in $O(n^2 \log n)$ amortized time per update, and supports inserting and deleting several edges of the graph with just one operation. Insertion of a bunch of edges incident to a vertex and deletion of any subset of edges in the graph require asymptotically the same time of inserting/deleting just one edge.

APPROACH

The algorithm maintains $\log n + 1$ levels: level i, $0 \leq i \leq \log n$, maintains a graph G_i whose edges represent paths of length up to 2^i in the original graph G. Thus, $G_0 = G$ and $G_{\log n}$ is the transitive closure of G.

FACTS

F36: Each level i is built on top of the previous level $i-1$ by keeping two trees of depth ≤ 2 rooted at each vertex v of G: an out-tree $OUT_i(v)$ maintaining vertices reachable from v by traversing at most two edges in G_{i-1}, and an in-tree $IN_i(v)$ maintaining vertices that reach v by traversing at most two edges in G_{i-1}. An edge (x, y) will be in G_i if and only if $x \in IN_i(v)$ and $y \in OUT_i(v)$ for some v.

F37: The $2 \log n$ trees $IN_i(v)$ and $OUT_i(v)$ are maintained with instances of the BFS tree data structure considered in §10.2.4 (Reachability Trees).

F38: To update the levels after an insertion of edges around a vertex v in G, the algorithm simply rebuilds $IN_i(v)$ and $OUT_i(v)$ for each i, $1 \leq i \leq \log n$, while other trees are not touched. This means that some trees might not be up to date after an insertion operation. Nevertheless, any path in G is represented in at least the in/out trees rooted at the latest updated vertex in the path, so the reachability information is correctly maintained. This idea is the key ingredient of King's algorithm.

F39: When an edge is deleted from G_i, it is also deleted from any data structures $IN_i(v)$ and $OUT_i(v)$ that contain it. For details, see [Ki99].

Demetrescu and Italiano's $O(n^2)$ Update Algorithm

The algorithm by Demetrescu and Italiano [DeIt00] is based on the matrix data structure considered in §10.2.4 (Matrix Data Structures) and on the recursive decomposition discussed in §10.2.4 (Path Problems and Kleene Closures). It maintains explicitly the transitive closure of a graph in $O(n^2)$ amortized time per update, supporting the same generalized update operations of King's algorithm, i.e., insertion of a bunch of edges incident to a vertex and deletion of any subset of edges in the graph with just one operation. This is the best known update bound for fully dynamic transitive closure with constant query time.

APPROACH

The algorithm maintains the Kleene closure X^* of the $n \times n$ adjacency matrix X of the graph as the sum of two matrices X_1 and X_2.

NOTATION

Let V_1 be the subset of vertices of the graph corresponding to the first half of indices of X, and let V_2 contain the remaining vertices.

FACTS

F40: Both matrices X_1 and X_2 are defined according to Munro's equations of §10.2.4 (Path Problems and Kleene Closures), but in such a way that paths appearing due to an insertion of edges around a vertex in V_1 are correctly recorded in X_1, while paths that appear due to an insertion of edges around a vertex in V_2 are correctly recorded in X_2. Thus, neither X_1 nor X_2 encode complete information about X^*, but their sum does.

F41: In more detail, assuming that X is decomposed in sub-matrices A, B, C, D as explained in §10.2.4 (Path Problems and Kleene Closures), and that X_1, and X_2 are similarly decomposed in sub-matrices E_1, F_1, G_1, H_1 and E_2, F_2, G_2, H_2, the algorithm maintains X_1 and X_2 with the following 8 polynomials using the data structure discussed in §10.2.4 (Matrix Data Structures):

$$\begin{array}{ll} Q = A + BP^2C & E_2 = E_1BH_2^2CE_1 \\ F_1 = E_1^2BP & F_2 = E_1BH_2^2 \\ G_1 = PCE_1^2 & G_2 = H_2^2CE_1 \\ H_1 = PCE_1^2BP & R = D + CE_1^2B \end{array}$$

where $P = D^*$, $E_1 = Q^*$, and $H_2 = R^*$ are Kleene closures maintained recursively as smaller instances of the problem of size $n/2 \times n/2$.

F42: To support an insertion of edges around a vertex in V_1, strict updates are performed on polynomials Q, F_1, G_1, and H_1 using `SetRow` and `SetCol`, while E_2, F_2, G_2, and R are updated with `LazySet`.

F43: Insertions around V_2 are performed symmetrically, while deletions are supported via `Reset` operations on each polynomial in the recursive decomposition.

F44: Finally, P, E_1, and H_2 are updated recursively. The low-level details of the method appear in [DeIt00].

10.2.6 Dynamic Shortest Paths

In this subsection we survey the best known algorithms for fully dynamic all pairs shortest paths (in short APSP). Given a weighted directed graph G with n vertices and m edges, the problem consists of supporting any intermixed sequence of operations of the following kind:

`Update`(u, v, w): updates the weight of edge (u, v) in G to the new value w (if $w = +\infty$ this corresponds to edge deletion);

`Query`(x, y): returns the distance from vertex x to vertex y in G, or $+\infty$ if no path between them exists;

REMARKS

R46: The dynamic maintenance of shortest paths has a remarkably long history, as the first papers date back to 35 years ago [Lo67,Mu67,Ro68]. After that, many dynamic shortest paths algorithms have been proposed (see, e.g., [EvGa85,FrMaNa98,FrMaNa00, RaRe96a,RaRe96b,Ro85]), but their running times in the worst case were comparable to recomputing APSP from scratch.

R47: The first dynamic shortest path algorithms which are provably faster than recomputing APSP from scratch, only worked on graphs with small integer weights.

R48: In particular, Ausiello, et al. [AuItMaNa91] proposed a decrease-only shortest path algorithm for directed graphs having positive integer weights less than C: the amortized running time of their algorithm is $O(Cn \log n)$ per edge insertion.

R49: Henzinger, et al. [HeKiRaSu97] designed a fully dynamic algorithm for APSP on planar graphs with integer weights, with a running time of $O(n^{4/3} \log(nC))$ per operation.

R50: This bound has been improved by Fakcharoemphol and Rao in [FaRa01], who designed a fully dynamic algorithm for single-source shortest paths in planar directed graphs that supports both queries and edge weight updates in $O(n^{4/5} \log^{13/5} n)$ amortized time per edge operation.

R51: The first big step on general graphs and integer weights was made by King [Ki99], who presented a fully dynamic algorithm for maintaining all pairs shortest paths in directed graphs with positive integer weights less than C: the running time of her algorithm is $O(n^{2.5}\sqrt{C \log n})$ per update.

R52: Demetrescu and Italiano [DeIt01] gave the first algorithm for fully dynamic APSP on general directed graphs with real weights assuming that each edge weight can attain a limited number S of different *real* values throughout the sequence of updates. In particular, the algorithm supports each update in $O(n^{2.5}\sqrt{S\log^3 n}\,)$ amortized time and each query in $O(1)$ worst-case time.

R53: The same authors discovered the first algorithm that solves the fully dynamic all pairs shortest paths problem in its generality [DeIt03]. The algorithm maintains explicitly information about shortest paths supporting any edge weight update in $O(n^2\log^2 n)$ amortized time per operation in directed graphs with non-negative real edge weights. Distance queries are answered with one lookup and actual shortest paths can be reconstructed in optimal time.

R54: We note that each update might change a portion of the distance matrix as large as $\Omega(n^2)$. Thus, if the distance matrix has to be maintained explicitly after each update so that queries can be answered with one lookup, $O(n^2)$ is the best update bound one could hope for.

R55: Other deletions-only algorithms for APSP, in the simpler case of unweighted graphs, are presented in [BaHaSe02].

King's $O(n^{2.5}\sqrt{C\log n}\,)$ Update Algorithm

The dynamic shortest paths algorithm by King [Ki99] is based on the long paths property discussed in §10.2.4 (Long Paths Property) and on the tree data structure of §10.2.4 (Reachability Trees). Similarly to the transitive closure algorithms described in §10.2.5, generalized update operations are supported within the same bounds, i.e., insertion (or weight decrease) of a bunch of edges incident to a vertex, and deletion (or weight increase) of any subset of edges in the graph with just one operation.

APPROACH

The main idea of the algorithm is to maintain dynamically all pairs shortest paths up to a distance d, and to recompute longer shortest paths from scratch at each update by stitching together shortest paths of length $\leq d$. For the sake of simplicity, we only consider the case of unweighted graphs: an extension to deal with positive integer weights less than C is described in [Ki99].

FACTS

F45: To maintain shortest paths up to distance d, similarly to the transitive closure algorithm by King described in §10.2.5, the algorithm keeps a pair of in/out shortest paths trees $IN(v)$ and $OUT(v)$ of depth $\leq d$ rooted at each vertex v. Trees $IN(v)$ and $OUT(v)$ are maintained with the decremental data structure mentioned in §10.2.4 (Reachability Trees). It is easy to prove that, if the distance d_{xy} between any pair of vertices x and y is at most d, then d_{xy} is equal to the minimum of $d_{xv} + d_{vy}$ over all vertices v such that $x \in IN(v)$ and $y \in OUT(v)$. To support updates, insertions of edges around a vertex v are handled by rebuilding only $IN(v)$ and $OUT(v)$, while edge deletions are performed via operations on any trees that contain them. The amortized cost of such updates is $O(n^2 d)$ per operation.

F46: To maintain shortest paths longer than d, the algorithm exploits the long paths property of Fact 30: in particular, it hinges on the observation that, if H is a random subset of $\Theta((n \log n)/d)$ vertices in the graph, then the probability of finding more than d consecutive vertices in a path, none of which are from H, is very small. Thus, if we look at vertices in H as "hubs", then any shortest path from x to y of length $\geq d$ can be obtained by stitching together shortest subpaths of length $\leq d$ that first go from x to a vertex in H, then jump between vertices in H, and eventually reach y from a vertex in H. This can be done by first computing shortest paths only between vertices in H using any cubic-time static all-pairs shortest paths algorithm, and then by extending them at both endpoints with shortest paths of length $\leq d$ to reach all other vertices. This stitching operation requires $O(n^2|H|) = O((n^3 \log n)/d)$ time.

F47: Choosing $d = \sqrt{n \log n}$ yields an $O(n^{2.5}\sqrt{\log n})$ amortized update time. As mentioned in §10.2.4 (Long Paths Property), since H can be computed deterministically, the algorithm can be derandomized. For further details, see [Ki99].

Demetrescu and Italiano's $O(n^2 \log^2 n)$ Update Algorithm

Demetrescu and Italiano [DeIt03] devised the first deterministic near-quadratic update algorithm for fully dynamic all-pairs shortest paths. This algorithm is also the first solution to the problem in its generality. It is based on the notions of uniform path, potentially uniform path, and historical shortest paths in a graph subject to a sequence of updates, as discussed in §10.2.4 (Uniform Paths).

APPROACH

The main idea is to maintain dynamically the potentially uniform paths of the graph in a data structure. Since by Fact 28 shortest paths are potentially uniform, this guarantees that information about shortest paths is maintained as well.

FACTS

F48: To support an edge weight update operation, the algorithm implements the smoothing strategy mentioned in §10.2.4 (Uniform Paths) and works in two phases. It first removes from the data structure all maintained paths that contain the updated edge: this is correct since historical shortest paths, in view of their definition, are immediately invalidated as soon as they are touched by an update. This means that also potentially uniform paths that contain them are invalidated and have to be removed from the data structure. As a second phase, the algorithm runs an all-pairs modification of Dijkstra's algorithm [Di59], where at each step a shortest path with minimum weight is extracted from a priority queue and it is combined with existing historical shortest paths to form new potentially uniform paths. At the end of this phase, paths that become potentially uniform after the update are correctly inserted in the data structure.

F49: The update algorithm spends constant time for each of the $O(zn^2)$ new potentially uniform path (see Fact 29). Since the smoothing strategy lets $z = O(\log n)$ and increases the length of the sequence of updates by an additional $O(\log n)$ factor, this yields $O(n^2 \log^2 n)$ amortized time per update. For further details, see [DeIt03].

RESEARCH ISSUES

In this work we have surveyed the algorithmic techniques underlying the fastest known dynamic graph algorithms for several problems, both on undirected and on directed graphs. Most of the algorithms that we have presented achieve bounds that are close to optimum. In particular, we have presented fully dynamic algorithms with polylogarithmic amortized time bounds for connectivity and minimum spanning trees [HoDeTh01] on undirected graphs. It remains an interesting open problem to show whether polylogarithmic update bounds can be achieved also in the worst case: we recall that for both problems the current best worst-case bound is $O(\sqrt{n})$ per update, and it is obtained with the sparsification technique [EpGaItNi97] described in §10.2.1.

For directed graphs, we have shown how to achieve constant-time query bounds and nearly-quadratic update bounds for transitive closure and all pairs shortest paths. These bounds are close to optimal in the sense that one update can make as many as $\Omega(n^2)$ changes to the transitive closure and to the all-pairs shortest paths matrices. However, if the problem is just to maintain reachability or shortest paths between two fixed vertices s and t, no solution better that the static is known. Furthermore, if one is willing to pay more for queries, Demetrescu and Italiano [DeIt00] have shown how to break the $O(n^2)$ barrier on the single-operation complexity of fully dynamic transitive closure for directed acyclic graphs. It remains an interesting open problem to show whether effective query/update tradeoffs can be achieved for general graphs and for shortest paths problems. Furthermore, can we solve efficiently fully dynamic single-source shortest paths on general graphs?

Finally, dynamic algorithms for other fundamental problems such as matching and flow problems deserve further investigation.

Further Information

Research on dynamic graph algorithms is published in many computer science journals, including *Algorithmica*, *Journal of ACM*, *Journal of Algorithms*, *Journal of Computer and System Science*, *SIAM Journal on Computing* and *Theoretical Computer Science.* Work on this area is published also in the proceedings of general theoretical computer science conferences, such as the *ACM Symposium on Theory of Computing* (STOC), the *IEEE Symposium on Foundations of Computer Science* (FOCS) and the *International Colloquium on Automata, Languages and Programming* (ICALP). More specialized conferences devoted exclusively to algorithms are the *ACM–SIAM Symposium on Discrete Algorithms* (SODA), and the *European Symposium on Algorithms* (ESA).

Acknowledgments

This work has been supported in part by the IST Programmes of the EU under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-33555 (COSIN), and by the Italian Ministry of University and Scientific Research (Project “ALINWEB: Algorithmics for Internet and the Web”).

References

[AlHoDeTh97] S. Alstrup, J. Holm, K. de Lichtenberg, and M. Thorup, Minimizing diameters of dynamic trees, *Proc. 24th Int. Colloquium on Automata, Languages and Programming (ICALP 97)* (1997), LNCS 1256, 270–280.

[AlHoTh00] S. Alstrup, J. Holm, and M. Thorup, Maintaining center and median in dynamic trees, *Proc. 7th Scandinavian Workshop on Algorithm Theory (SWAT 00)* (2000), 46–56.

[AuItMaNa91] G. Ausiello, G. F. Italiano, A. Marchetti-Spaccamela, and U. Nanni, Incremental algorithms for minimal length paths, *J. of Algorithms* 12(4) (1991), 615–638.

[BaHaSe02] S. Baswana, R. Hariharan, and S. Sen, Improved decremental algorithms for transitive closure and all-pairs shortest paths, *Proc. 34th ACM Symposium on Theory of Computing (STOC'02)* (2002), 117–123.

[Ch79] V. Chvátal, A greedy heuristic for the set-covering problem, *Mathematics of Operations Research* 4(3) (1979), 233–235.

[CoWi90] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *J. of Symbolic Computation* 9 (1990).

[CoLeRiSt01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition, MIT Press, 2001.

[DeIt00] C. Demetrescu and G. F. Italiano, Fully dynamic transitive closure: Breaking through the $O(n^2)$ barrier, *Proc. of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS'00)* (2000), 381–389.

[DeIt01] C. Demetrescu and G. F. Italiano, Fully dynamic all pairs shortest paths with real edge weights, *Proc. of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'01), Las Vegas, Nevada* (2001), 260–267.

[DeIt03] C. Demetrescu and G.F. Italiano, A new approach to dynamic all pairs shortest paths, *Proc. 35th Symp. on Theory of Computing (STOC'03), San Diego, CA* (2003), 159–166.

[Di59] E. W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik* 1 (1959), 269–271.

[EpGaItNi97] D. Eppstein, Z. Galil, G. F. Italiano, and A. Nissenzweig, Sparsification – A technique for speeding up dynamic graph algorithms, *J. Assoc. Comput. Mach.*, 44 (1997), 669–696.

[EvGa85] S. Even and H. Gazit, Updating distances in dynamic graphs, *Methods of Operations Research* 49 (1985), 371–387.

[EvSh81] S. Even and Y. Shiloach, An on-line edge deletion problem, *J. Assoc. Comput. Mach.* 28 (1981), 1–4.

[FaRa01] J. Fakcharoemphol and S. Rao, Planar graphs, negative weight edges, shortest paths, and near linear time, *Proc. of the 42nd IEEE Annual Symposium on Foundations of Computer Science (FOCS'01), Las Vegas, Nevada* (2001), 232–241.

[Fr85] G. N. Frederickson, Data structures for on-line updating of minimum spanning trees, *SIAM J. Comput.* 14 (1985), 781–798.

[Fr97] G. N. Frederickson, Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees, *SIAM J. Comput.* 26(2) (1997), 484–538.

[FrMaNa98] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni, Semi-dynamic algorithms for maintaining single source shortest paths trees, *Algorithmica* 22(3) (1998), 250–274.

[FrMaNa00] D. Frigioni, A. Marchetti-Spaccamela, and U. Nanni, Fully dynamic algorithms for maintaining shortest paths trees, *J. of Algorithms* 34 (2000), 251–281.

[GaIt92] Z. Galil and G. F. Italiano, Fully-dynamic algorithms for 2-edge connectivity, *SIAM J. Comput.* 21 (1992), 1047–1069.

[GrKn82] D. H. Greene and D.E. Knuth, *Mathematics for the analysis of algorithms*, Birkhäuser, 1982.

[Ha69] F. Harary, *Graph Theory*, Addison-Wesley, 1969.

[HeKi97] M. R. Henzinger and V. King, Maintaining minimum spanning trees in dynamic graphs, *Proc. 24th Int. Colloquium on Automata, Languages and Programming (ICALP 97)* (1997) 594–604.

[HeKi99] M. R. Henzinger and V. King, Randomized fully dynamic graph algorithms with polylogarithmic time per operation, *J. Assoc. Comput. Mach.* 46(4) (1999), 502–536.

[HeKiRaSu97] M.R. Henzinger, P. Klein, S. Rao, and S. Subramanian, Faster shortest-path algorithms for planar graphs, *J. of Computer and System Sciences* 55(1) (1997), 3–23.

[HoDeTh01] J. Holm, K. de Lichtenberg and M. Thorup, Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity, *J. Assoc. Comput. Mach.* 48(4) (2001), 723–760.

[KhMoWi96] S. Khanna, R. Motwani, and R. H. Wilson, On certificates and lookahead on dynamic graph problems, *Algorithmica* 21(4) (1998), 377–394.

[Ki99] V. King, Fully dynamic algorithms for maintaining all-pairs shortest paths and transitive closure in digraphs, *Proc. 40-th Symposium on Foundations of Computer Science (FOCS 99)* (1999).

[KiSa99] V. King and G. Sagert, A fully dynamic algorithm for maintaining the transitive closure, *Proc. 31-st ACM Symposium on Theory of Computing (STOC 99)* (1999), 492–498.

[Lo67] P. Loubal, A network evaluation procedure, *Highway Research Record 205* (1967), 96–109.

[Lo75] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Mathematics* 13 (1975), 383–390.

[Mu71] I. Munro, Efficient determination of the transitive closure of a directed graph, *Information Processing Letters* 1(2) (1971), 56–58.

[Mu67] J. Murchland, "The effect of increasing or decreasing the length of a single arc on all shortest distances in a graph", Technical report, LBS-TNT-26, London Business School, Transport Network Theory Unit, London, UK, 1967.

[RaRe96a] G. Ramalingam and T. Reps, An incremental algorithm for a generalization of the shortest path problem, *J. of Algorithms* 21 (1996), 267–305.

[RaRe96b] G. Ramalingam and T. Reps, On the computational complexity of dynamic graph problems, *Theoretical Computer Science* 158 (1996), 233–277.

[Ra95] M. Rauch, Fully dynamic biconnectivity in graphs, *Algorithmica* 13 (1995), 503–538.

[Ro68] V. Rodionov, The parametric problem of shortest distances, *U.S.S.R. Computational Math. and Math. Phys.* 8(5) (1968), 336–343.

[Ro85] H. Rohnert, A dynamization of the all-pairs least cost problem, *Proc. 2nd Annual Symposium on Theoretical Aspects of Computer Science, (STACS 85), LNCS 182* (1985), 279–286.

[SlTa83] D. D. Sleator and R. E. Tarjan, A data structure for dynamic trees, *J. Comp. Syst. Sci.* 24 (1983), 362–381.

[UlYa91] J. D. Ullman and M. Yannakakis, High-probability parallel transitive-closure algorithms, *SIAM J. on Computing* 20(1) (1991), 100–125.

[Zw98] U. Zwick, All pairs shortest paths in weighted directed graphs - exact and almost exact algorithms, *Proc. of the 39th IEEE Annual Symposium on Foundations of Computer Science (FOCS'98)* (1998), 310–319.

10.3 DRAWINGS OF GRAPHS

Giuseppe Liotta, University of Perugia, Italy
Roberto Tamassia, Brown University

10.3.1 Types of Graphs and Drawings
10.3.2 Combinatorics of Geometric Graphs
10.3.3 Properties of Drawings and Bounds
10.3.4 Complexity of Graph Drawing Problems
10.3.5 Example of a Graph Drawing Algorithm
10.3.6 Techniques for Drawing Graphs
10.3.7 Recent Research Trends
10.3.8 Sources and Related Material
References

Introduction

Research on graph drawing has been conducted within several diverse areas, including discrete mathematics (topological graph theory, geometric graph theory, order theory), algorithmics (graph algorithms, data structures, computational geometry, VLSI), and human-computer interaction (visual languages, graphical user interfaces, software visualization). In this section, we overview two different aspects of the current research in graph drawing: the study of the graph theoretic properties of families of geometric representations of graphs and the algorithmic issues involved in computing a drawing of a graph that satisfies a given set of geometric constraints.

10.3.1 Types of Graphs and Drawings

Graph drawing concerns geometric representations of graphs, and it has important applications to key computer technologies such as software engineering, database systems, visual interfaces, and computer-aided-design.

Types of Graphs

First, we define some terminology on graphs pertinent to graph drawing. Throughout this section, let n and m be the number of graph vertices and edges respectively, and let d be the maximum vertex degree (i.e., number of incident edges).

DEFINITIONS

D1: A ***degree-k graph*** is a graph with maximum degree $d \leq k$.

D2: A ***transitive edge*** of a digraph is an edge (u, v) such that there is a directed path from u to v not containing edge (u, v).

D3: A ***reduced digraph*** is a digraph without transitive edges.

D4: A ***source vertex of a digraph*** is a vertex without incoming edges.

D5: A ***sink vertex of a digraph*** (also called a ***target***) is a vertex without outgoing edges.

D6: An **st-*digraph*** (also called a ***bipolar digraph***) is an acyclic digraph with exactly one source and one sink, which are joined by an edge.

D7: A ***biconnected graph*** is a 2-connected graph; that is, any two vertices are joined by two vertex-disjoint paths.

D8: A ***triconnected graph*** is a 3-connected graph; that is, any two vertices are joined by three (pairwise) vertex-disjoint paths.

D9: A ***rooted tree*** is a directed tree with a distinguished vertex, called the ***root*** such that each vertex lies on a directed path to the root. (We observe that this reverses the usual convention.)

D10: A ***binary tree*** is a rooted tree such that each vertex has at most two incoming edges.

D11: A ***layered (di)graph*** is a (di)graph whose vertices are partitioned into sets, called layers. A rooted tree can be viewed as a layered digraph where the layers are sets of vertices at the same distance from the root.

D12: A ***k-layered (di)graph*** layered (di)graph has k layers.

Types of Drawings

In a drawing of a graph, vertices are represented by points (or by geometric figures such as circles or rectangles) and edges are represented by curves such that any two edges intersect at most in a finite number of points. The following definitions are relative to drawings in the plane, that are the main subject of this section.

DEFINITIONS

D13: In a ***polyline drawing***, each edge is a polygonal chain (see Figure 10.3.1(a)).

D14: In a ***straight-line drawing***, each edge is a straight-line segment (see Figure 10.3.1(b)).

D15: In an ***orthogonal drawing***, each edge is a chain of horizontal and vertical segments (see Figure 10.3.1(c)).

D16: A ***bend in a polyline drawing*** is a point where two segments belonging to the same edge meet (see Figure 10.3.1(a)).

D17: An ***orthogonal representation*** of an orthogonal drawing is in terms of the bends along each edge and the angles around each vertex.

D18: A ***crossing*** is a point of a graph drawing where two edges intersect (see Figure 10.3.1(b)).

D19: A ***grid drawing*** is a polyline drawing such that the vertices, crossings, and bends all have integer coordinates.

D20: In a ***planar drawing***, no two edges cross (see Figure 10.3.1(d)).

D21: A ***planar (di)graph*** is a (di)graph that admits a planar drawing.

D22: An ***imbedded (di)graph*** is a planar (di)graph with a prespecified topological imbedding (i.e., set of faces), which must be preserved in the drawing.

D23: In an ***upward drawing*** of a digraph, each edge is monotonically nondecreasing in the vertical direction (see Figure 10.3.1(d)).

D24: An ***upward planar digraph*** admits an upward planar drawing.

D25: In a ***layered drawing*** of a layered graph (also called a ***hierarchical drawing***), the vertices in the same layer all lie on the same horizontal line.

D26: A ***face*** is a region of a planar drawing, and the unbounded region is called the ***external face***.

D27: A ***convex drawing*** is a planar straight-line drawing of a graph such that the boundary of each face is a convex polygon.

D28: A ***visibility drawing*** of a graph is based on a geometrically visible relation; e.g., the vertices might be drawn as horizontal segments, and the edges associated with vertically visible segments.

D29: A ***dominance drawing*** is an upward drawing of an acyclic digraph such that there exists a directed path from vertex u to vertex v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$, where $x(\cdot)$ and $y(\cdot)$ denote the coordinates of a vertex.

D30: An **hv-*drawing*** is an upward orthogonal straight-line drawing of a binary tree such that the drawings of the subtrees of each node are separated by a horizontal or vertical line.

EXAMPLE

E1: In Figure 10.3.1 the first three drawings are of the complete bipartite graph $K_{3,3}$.

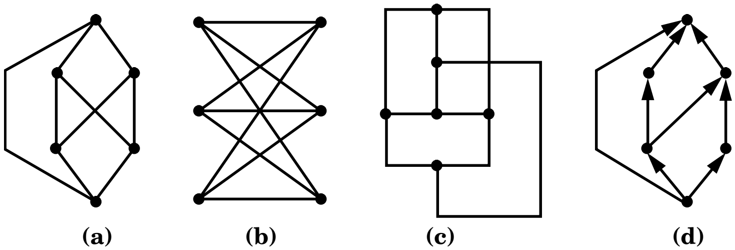


Figure 10.3.1 **Drawings: (a) polyline; (b) straight-line; (c) orthogonal; (d) planar upward.**

REMARKS

R1: Polyline drawings provide great flexibility since they can approximate drawings with curved edges. However, edges with more than two or three bends may be difficult to "follow" for the eye. Also, a system that supports editing of polyline drawings is more complicated than one limited to straight-line drawings. Hence, depending on the application, polyline or straight-line drawings may be preferred.

R2: If vertices are represented by points, then orthogonal drawings exist only for graphs of maximum vertex degree 4.

10.3.2 Combinatorics of Geometric Graphs

Geometric graphs [PS90, OBS92] are straight-line drawings that satisfy some additional geometric constraints. Much attention has been devoted in the literature to the study of combinatorial properties of well-known geometric graphs such as Delaunay triangulations, minimum spanning trees, Gabriel graphs, relative neighborhood graphs, β-skeleton graphs, and rectangle of influence graphs. This interest has been motivated in part by the importance of these structures in numerous application areas, including computer graphics, pattern recognition, computational morphology, communication networks, numerical analysis, computational biology, and GIS.

The problem of analyzing the combinatorial properties of a given type of geometric graph naturally raises the question of the characterization of those graphs which admit the given type of straight-line drawing. This, in turn, leads to the investigation of the design of efficient algorithms for computing such a drawing when one exists. Although these questions are far from being resolved in general, many partial answers have appeared in the literature (see, e.g., [Dil90a, DS94, JT92, LL96, LL02, LLMW98, LM03, LS93] for recent papers dealing with drawability questions related to different types of geometric graphs). In the following, we briefly survey some results concerning well-known families of geometric graphs and outline some open problems.

Delaunay Triangulations

DEFINITIONS

D31: A ***Delaunay triangulation*** is a planar straight-line drawing with all internal faces triangles and such that three vertices form a face if and only if their convex hull does not contain any other vertex of the triangulation.

D32: A planar triangulated graph is ***Delaunay drawable*** if it admits a drawing that is a Delaunay triangulation.

D33: A ***Voronoi diagram*** is the dual graph of a Delaunay triangulation.

FACTS

F1: [Dil90b] All Delaunay drawable triangulations are 1-tough and have perfect matchings.

F2: [Dil90a] All maximal outerplanar graphs are Delaunay drawable.

F3: [DS94] Any triangulation without chords or non-facial triangles is Delaunay drawable.

REMARKS

R3: Di Battista and Vismara [DV96] give a characterization based on a non-linear system of equations involving the angles in the triangulation.

R4: Liotta and Meijer have studied the combinatorial properties of a Voronoi diagram. In particular, a characterization of Voronoi drawable trees can be found in [LM03].

Minimum Weight Triangulations

DEFINITIONS

D34: A ***triangulation*** T of a set P of points on the plane is a straight-line drawing whose vertices are the elements in P and all internal faces are triangles.

D35: T is a ***minimum weight triangulation*** if it is a triangulation of P that minimizes the total edge length.

D36: A ***minimum weight drawing*** of a planar triangulated graph G is a straight-line drawing Γ of G with the additional property that Γ is a minimum weight triangulation of the points representing the vertices.

D37: If a graph admits a minimum weight drawing it is called ***minimum weight drawable***; else, it is called ***minimum weight forbidden***.

FACTS

Little is known about the problem of constructing a minimum weight drawing of a planar triangulation.

F4: It is not known whether computing a minimum weight triangulation of a set of points in the plane is an NP-hard problem (see Garey and Johnson [GJ79]). Several papers have been published on this problem, either providing partial solutions, or giving efficient approximation heuristics. A limited list of references includes the work by Meijer and Rappaport [RM92], Lingas [Lin87], Keil [Kei94], Dickerson et al. [MDM97], Kirkpatrick [Kir80], Aichholzer et al. [AAC$^+$96], Cheng and Xu [CX01], Dickerson and Montague [DM96], and Levcopoulos and Krznaric [LK96, LK98].

F5: [LL96] All maximal outerplanar triangulations are minimum weight drawable and a linear time (real RAM) drawing algorithm for computing a minimum weight drawing of these graphs was given.

This naturally leads to investigate the internal structure of minimum weight drawable triangulations. In [LL02] Lenhart and Liotta examine the ***endoskeleton*** of a triangulation: that is, the subgraph induced by the internal vertices of the triangulation. They construct skeletons that cannot appear in any minimum weight drawable triangulation; skeletons that do appear in minimum weight drawable triangulations; and skeletons that guarantee minimum weight drawability. Wang, Chin, and Yang [WCY00] also focus on the minimum weight drawability of triangulations and show examples of triangulations with acyclic skeletons that do not admit a minimum weight drawing.

F6: [LL02] There exists an infinite class of minimum weight drawable triangulations that cannot be realized as Delaunay triangulations (that is, for any triangulation T of the class it does not exist a set P of points such that the Delaunay triangulation of P is isomorphic to T).

It is worth remarking that the study of the geometric differences between the minimum-weight and Delaunay triangulations of a given set of points in order to compute good approximations of the former has a long tradition (see, e.g., [Kir80, LK96, MZ79]), little is known about the combinatorial difference between Delaunay triangulations and minimum-weight triangulations.

Minimum Spanning Trees

DEFINITIONS

D38: A ***minimum spanning tree of a set P of points*** is a connected, straight-line drawing that has P as vertex set and minimizes the total edge length.

D39: A tree T is ***drawable as a minimum spanning tree*** if there exists a set P of points such that the minimum spanning tree of P is isomorphic to T. (The problem is that whatever plane locations are assigned to vertices of the tree T, perhaps the image of T itself is not the minimum spanning tree for those vertex locations.)

FACTS

The problem of testing whether a tree can be drawn as a Euclidean minimum spanning tree in the plane is essentially solved. The 3-dimensional counterpart of the problem is not yet solved.

F7: [MS92] Each tree with maximum vertex degree at most five can be drawn as a minimum spanning tree of some set of vertices. There is a linear time (real RAM) algorithm.

F8: [MS92] No tree with maximum degree greater than six can be drawn as a minimum spanning tree.

F9: [EW96] It is NP-hard to decide whether trees of maximum degree equal to six can be drawn as minimum spanning trees.

F10: [LD95] No trees with maximum degree greater than twelve can be drawn as a Euclidean minimum spanning tree in 3D-space, while all trees with vertex degree at most nine are drawable.

Proximity Graphs

Different specifications of *proximity region* give rise to different families of proximity graphs.

DEFINITIONS

D40: A ***proximity graph*** of a set P of n distinct points in the plane is a straight-line drawing with P as vertex set, such that two vertices are connected if and only if a suitable region called their ***proximity region*** does not contain any other element of P.

D41: The ***Gabriel graph*** of a set P is a proximity graph such that there exists an edge (u, v) if and only if the disk whose antipodal points are u and v does not contain any points of P.

D42: The ***rectangle of influence graph*** of a set P is a proximity graph such that there exists an edge (u, v) if and only if the axis-aligned rectangle having u and v at opposite corners does not contain any points of P.

REMARK

R5: Proximity graphs have been first studied in the context of pattern recognition and computational morphology, where one is given a set of points on the plane and is asked to display the underlying ***shape*** of the set by constructing a graph whose vertices are the points and whose edges are segments connecting pairs of points. For references on this topics see, e.g., the survey by Toussaint [JT92].

FACTS

More recently, the problem of computing straight-line drawings that have the additional property of being proximity graphs of their vertices has been considered. Besides its theoretical interest, one of the motivations for this study is that proximity drawings capture the natural aesthetic requirement that adjacent vertices are visually clustered together while disjoint vertices are drawn far from each other.

F11: In the work by Bose, Lenhart, and Liotta [BLL96], the problem of characterizing Gabriel-drawable trees has been addressed and an algorithm to compute Gabriel drawings of trees in the plane is given.

F12: Lubiw and Sleumer [LS93] proved that maximal outerplanar graphs admit both relative neighborhood drawings and Gabriel drawings.

F13: Fact 12 has been extended in [LL97] to all biconnected outerplanar graphs. Different families of graphs that admit a rectangle of influence drawing are described by Liotta, Lubiw, Meijer, and Whitesides [LLMW98] and by Biedl, Bretscher, and Meijer [BBM99].

Open Problems

P1. Give a complete combinatorial characterization of Delaunay drawable triangulations.

P2. Let T be a tree with maximum vertex degree at most twelve. Is there a polynomial time algorithm to decide whether T can be drawn as a Euclidean minimum spanning tree in 3D-space? If so, compute such a drawing.

P3. Define new families of minimum weight drawable triangulations. For example, characterize the class of triangulations with acyclic skeleton that admit a minimum weight drawing.

P4. Investigate the combinatorial relationship between minimum weight and Delaunay drawable triangulations. Are there any Delaunay drawable and minimum weight forbidden triangulations?

P5. Further study the combinatorial structure of proximity graphs. For example, characterize the family of Gabriel drawable triangulations, that is the family of those triangulations that admit a straight-line drawing where the angles of each triangular face are less than $\pi/2$.

10.3.3 Properties of Drawings and Bounds

For various classes of graphs and drawing types, many universal/existential upper and lower bounds for specific drawing properties have been discovered. Such bounds typically exhibit tradeoffs between drawing properties. A universal bound applies to all the graphs of a given class. An existential bound applies to infinitely many graphs of the class. Whenever we give bounds on the area or edge length, we assume that the drawing is constrained by some resolution rule that prevents it from being arbitrarily scaled down reduced by an arbitrary scaling (e.g., requiring a grid drawing, or stipulating a minimum unit distance between any two vertices).

Properties of Drawings

In computing graph drawings, we would like to take into account a variety of properties. For example, planarity and the display of symmetries are highly desirable in visualization applications. Or we may want to display trees and acyclic digraphs with upward drawings. In general, to avoid wasting valuable space on a page or a computer screen, it is important to keep the area of the drawing small. Moreover, it is typically desirable to maximize the angular resolution and to minimize the other measures.

DEFINITIONS

D43: The ***crossing number*** χ ***of a drawing*** is its total number of edge-crossings.

D44: The ***area of a drawing*** is the area of its convex hull.

D45: The ***total edge length of a drawing*** is the sum of the lengths of the edges.

D46: The ***number of bends of a polyline drawing*** is the total number of bends on the edges of a drawing.

D47: The ***maximum number of bends of a polyline drawing*** is the maximum number of bends on any edge.

D48: The ***angular resolution*** ρ ***in a polyline drawing*** is the smallest angle formed by any two edges or segments of edges, incident on the same vertex or bend.

D49: The ***aspect ratio of a drawing*** is the ratio of the longest side to the shortest side of the smallest rectangle with horizontal and vertical sides covering the drawing.

EXAMPLES

The need of satisfying different drawing properties at the same time leads to formalizing many graph drawing problems as multi-objective optimization problems (e.g., construct a drawing with minimum area and minimum number of crossings), so that tradeoffs are inherent in solving them.

E2: Figure 10.3.2(a–b) below shows two drawings of K_4, the complete graph on four vertices. The drawing of part (a) is planar, while the drawing of part (b) "maximizes symmetries." It can be shown that no drawing of K_4 is optimal with respect to both criteria, i.e., the maximum number of symmetries cannot be achieved by a planar drawing.

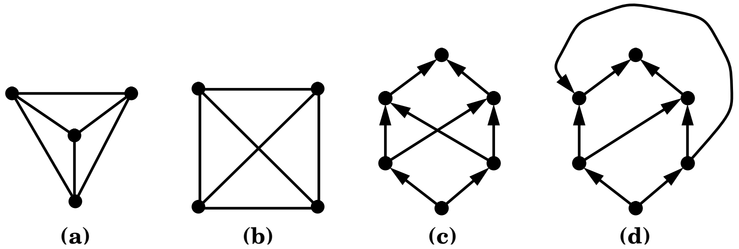


Figure 10.3.2 Tradeoffs: (a–b) planarity symmetry; (c–d) planarity upwardness.

E3: Figure 10.3.2(c–d), shows two drawing of the same acyclic digraph G. The drawing of part (c) is upward, while the drawing of part (d) is planar. It can be shown that there is no drawing of G that is both planar and upward.

Bounds on the Area

Table 10.3.1 below summarizes selected universal upper bounds and existential lower bounds on the area of drawings of graphs. In the table, a is an arbitrary constant $0 \leq a < 1$, and b and c are fixed constants $1 < b < c$, and ϵ for an arbitrary positive constant. The abbreviations "PSL" and "PSLg" are used for "planar straight-line" and "planar straight-line grid", respectively.

In general, the effect of bends on the area requirement is dual. On one hand, bends occupy space and hence negatively affect the area. On the other hand, bends may help in routing edges without using additional space.

FACTS

In these references to Table 10.3.1, specific rows of the table are indicated within parentheses.

F14: Linear or almost-linear bounds on the area can be achieved for trees (1–12).

F15: Planar graphs admit planar drawings with quadratic area (13–18). However, the area requirement of planar straight-line drawings may be exponential if high angular resolution is also desired (14).

F16: Almost linear area instead can be achieved in nonplanar drawings of planar graphs (18), which have applications to VLSI circuits.

F17: Upward planar drawings provide an interesting tradeoff between area and the total number of bends (19–21). Indeed, unless the digraph is reduced (20), the area can become exponential if a straight-line drawing is required (19).

F18: A quadratic area bound is achieved only at the expense of a linear number of bends (21).

See Table 10.3.4 below for tradeoffs between area and aspect ratio in drawings of trees.

Bounds on the Angular Resolution

Table 10.3.2 below summarizes selected universal lower bounds and existential upper bounds on the angular resolution of drawings of graphs. Here c is a fixed constant with $c > 1$.

Table 10.3.1 Universal upper and existential lower bounds on area.

	CLASS OF GRAPHS	DRAWING TYPE	AREA	
1	rooted tree	upward PSLg	$\Omega(n)$	$O(n \log n)$
2	rooted tree	strictly upward PSLg	$\Omega(n \log n)$	$O(n \log n)$
3	degree-$O(n^a)$ rooted tree	upward planar polyline grid	$\Omega(n)$	$O(n)$
4	binary tree	upward planar orthogonal grid	$\Omega(n \log \log n)$	$O(n \log \log n)$
5	binary tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n^{1+\epsilon})$
6	Fibonacci tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
7	AVL tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
8	balanced tree	strictly upward order preserving PSLg	$\Omega(n)$	$O(n)$
9	binary tree	PSLg	$\Omega(n)$	$O(n)$
10	tree	PSLg	$\Omega(n)$	$O(n \log n)$
11	degree-$O(n^a)$ tree	planar polyline grid	$\Omega(n)$	$O(n)$
12	degree-4 tree	planar orthog grid	$\Omega(n)$	$O(n)$
13	planar graph	planar polyline grid	$\Omega(n^2)$	$O(n^2)$
14	planar graph	PSL	$\Omega(c^{\rho n})$	
15	planar graph	PSLg	$\Omega(n^2)$	$O(n^2)$
16	triconnected planar graph	PSL convex grid	$\Omega(n^2)$	$O(n^2)$
17	planar graph	planar orthog grid	$\Omega(n^2)$	$O(n^2)$
18	planar degree-4 graph	orthogonal grid	$\Omega(n \log n)$	$O(n \log^2 n)$
19	up planar digraph	upward PSLg	$\Omega(b^n)$	$O(c^n)$
20	reduced planar *st*-digraph	upward PSLg dominance	$\Omega(n^2)$	$O(n^2)$
21	up planar digraph	up planar grid polyline	$\Omega(n^2)$	$O(n^2)$
22	general graph	polyline grid	$\Omega(n + \chi)$	$O((n + \chi)^2)$

Table 10.3.2 Universal lower bounds and existential upper bounds on angular resolution.

CLASS OF GRAPHS	DRAWING TYPE	ANGULAR RESOLUTION	
general graph	straight-line	$\Omega(\frac{1}{d^2})$	$O(\frac{\log d}{d^2})$
planar graph	straight-line	$\Omega(\frac{1}{d})$	$O(\frac{1}{d})$
planar graph	planar straight-line	$\Omega(\frac{1}{c^d})$	$O(\sqrt{\frac{\log d}{d^3}})$

Bounds on the Number of Bends

Table 10.3.3 summarizes selected universal upper bounds and existential lower bounds on the total and maximum number of bends in orthogonal drawings. Some bounds are stated for $n \geq 5$ or $n \geq 7$ because the maximum number of bends is at least 2 for K_4 and at least 3 for the skeleton graph of an octahedron, in any planar orthogonal drawing.

Table 10.3.3 Orthogonal drawings: universal upper bounds and existential lower bounds on the number of bends. Notes: † $n \geq 7$; ‡ $n \geq 5$.

CLASS OF GRAPHS	DRAWING TYPE	TOTAL # BENDS		MAX # BENDS	
deg-4 †	orthog	$\geq n$	$\leq 2n+2$	≥ 2	≤ 2
planar deg-4 †	orthog planar	$\geq 2n-2$	$\leq 2n+2$	≥ 2	≤ 2
imbedded deg-4	orthog planar	$\geq 2n-2$	$\leq \frac{12}{5}n+2$	≥ 3	≤ 3
biconnected imbedded deg-4	orthog planar	$\geq 2n-2$	$\leq 2n+2$	≥ 3	≤ 3
triconnected imbedded deg-4	orthog planar	$\geq \frac{4}{3}(n-1)+2$	$\leq \frac{3}{2}n+4$	≥ 2	≤ 2
imbedded deg-3 ‡	orthog planar	$\geq \frac{1}{2}n+1$	$\leq \frac{1}{2}n+1$	≥ 1	≤ 1

Tradeoff Between Area and Aspect Ratio

The ability to construct area-efficient drawings is essential in practical visualization applications, where screen space is at a premium. However, achieving small area is not enough: e.g., a drawing with high aspect ratio may not be conveniently placed on a workstation screen, even if it has modest area. Hence, it is important to keep the aspect ratio small. Ideally, one would like to obtain small area for any given aspect ratio in a wide range. This would provide graphical user interfaces with the flexibility of fitting drawings into arbitrarily shaped windows. A variety of tradeoffs for the area and aspect ratio arise even when drawing graphs with a simple structure, such as trees.

FACTS

Table 10.3.4 below summarizes selected universal bounds that can be simultaneously achieved on the area and the aspect ratio of various types of drawings of trees.

Table 10.3.4 Trees: universal upper bounds simultaneously achievable for area and aspect ratio.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ASPECT RATIO
rooted tree	upward PSL layered grid	$O(n^2)$	$O(1)$
rooted tree	upward PSLg	$O(n \log\log n)$	$O(n \log\log n / \log^2 n)$
rooted deg-$O(1)$ tree	upward planar polyline grid	$O(n)$	$O(n^a)$
binary tree	up planar orthog grid	$O(n \log\log n)$	$O(n \log\log n / \log^2 n)$
binary tree	PSLg	$O(n)$	$[O(1), O(n^a)]$
binary tree	PSLog	$O(n \log\log n)$	$\left[O(1), O\left(\frac{n \log\log n}{\log^2 n}\right)\right]$
binary tree	upward PSLog	$O(n \log n)$	$O(1)$
deg-4 tree	orthog grid	$O(n)$	$O(1)$
deg-4 tree	orthog grid, leaves on convex hull	$O(n \log n)$	$O(1)$

In Table 10.3.4, a is an arbitrary constant with $0 \leq a < 1$ and the abbreviation "PSLog" is used for "planar straight-line orthgonal grid," that is a straight-line grid drawing where the edges are either horizontal or vertical straight-line segments. Only for a few cases there exist algorithms that guarantee efficient area performance and that can accept any user-specified aspect ratio in a given range. For such cases the aspect ratio in Table 10.3.4 is given as an interval.

REMARKS

R6: While upward planar straight-line drawings are the most natural way of visualizing rooted trees, the existing drawing techniques are unsatisfactory with respect to either the area requirement or the aspect ratio. The situation is similar for orthogonal drawings.

R7: Regarding polyline drawings, linear area can be achieved with a prescribed aspect ratio. However, experiments show that this is done at the expense of a somehow aesthetically unappealing drawing.

R8: For nonupward drawings of trees, linear area and optimal aspect ratio are possible for planar orthogonal drawings, and a small (logarithmic) amount of extra area is needed if the leaves are constrained to be on the convex hull of the drawing (e.g., pins on the boundary of a VLSI circuit). However, the nonupward drawing methods do not seem to yield aesthetically pleasing drawings, and are suited more for VLSI layout than for visualization applications.

Tradeoff between Area and Angular Resolution

Table 10.3.5 summarizes selected universal bounds that can be simultaneously achieved on the area and the angular resolution of drawings of graphs. Here b and c are fixed constants, $b > 1$ and $c > 1$. Universal lower bounds on the angular resolution exist that depend only on the degree of the graph. Also, substantially better bounds can be achieved by drawing a planar graph with bends or in a nonplanar way.

Table 10.3.5 Universal upper bounds for area and lower bounds for angular resolution, simultaneously achievable.

CLASS OF GRAPHS	DRAWING TYPE	AREA	ANGULAR RESOLUTION
planar graph	straight-line	$O(d^6 n)$	$\Omega(\frac{1}{d^2})$
planar graph	straight-line	$O(d^3 n)$	$\Omega(\frac{1}{d})$
planar graph	planar straight-line grid	$O(n^2)$	$\Omega(\frac{1}{n^2})$
planar graph	planar straight-line	$O(b^n)$	$\Omega(\frac{1}{c^d})$
planar graph	planar polyline grid	$O(n^2)$	$\Omega(\frac{1}{d})$

Open Problems

P6. Determine the area requirement of (upward) planar straight-line drawings of trees. There is currently an $O(\log n)$ gap between the known upper and lower bounds (Table 10.3.1).

P7. Determine the area requirement of strictly upward planar order preserving straight-line drawings of binary trees (Table 10.3.1).

P8. Determine the area requirement of orthogonal (or, more generally, polyline) nonplanar drawings of planar graphs. There is currently an $O(\log n)$ gap between the known upper and lower bounds (Table 10.3.1).

P9. Close the wide gap between the $\Omega(\frac{1}{d^2})$ universal lower bound and the $O(\frac{\log d}{d^2})$ existential upper bound on the angular resolution of straight-line drawings of general graphs (Table 10.3.2).

P10. Close the gap between the $\Omega(\frac{1}{c^d})$ universal lower bound and the $O(\sqrt{\frac{\log d}{d^3}})$ existential upper bound on the angular resolution of planar straight-line drawings of planar graphs (Table 10.3.2).

P11. Determine the best possible aspect ratio and area simultaneously achievable for (upward) planar straight-line and orthogonal drawings of trees (Table 10.3.4).

10.3.4 Complexity of Graph Drawing Problems

Tables 10.3.6–10.3.8 below summarize selected results on the time complexity of some fundamental graph drawing problems.

It is interesting that apparently similar problems exhibit very different time complexities. For example, while planarity testing can be done in linear time, upward planarity testing is NP-hard. Note that, as illustrated in Figure 10.3.1(c–d), planarity and acyclicity are necessary but not sufficient conditions for upward planarity. While many efficient algorithms exist for constructing drawings of trees and planar graphs with good univer-

Table 10.3.6 Time complexity of some fundamental graph drawing problems: general graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
general graph	minimize crossings	NP-hard	
2-layered graph	minimize crossings in layered drawing with preassigned order on one layer	NP-hard	
general graph	maximum planar subgraph	NP-hard	
general graph	planarity testing and computing a planar imbedding	$\Omega(n)$	$O(n)$
general graph	maximal planar subgraph	$\Omega(n+m)$	$O(n+m)$
general digraph	upward planarity testing	NP-hard	
imbedded digraph	upward planarity testing	$\Omega(n)$	$O(n^2)$
single-source digraph	upward planarity testing	$\Omega(n)$	$O(n)$
general graph	draw as the intersection graph of a set of unit diameter disks in the plane	NP-hard	

sal area bounds, exact area minimization for most types of drawings is NP-hard, even for trees.

Open Problems

P12. Reduce the time complexity of upward planarity testing for imbedded digraphs (currently $O(n^2)$), or prove a superlinear lower bound (Table 10.3.1).

P13. Reduce the time complexity of bend minimization for planar orthogonal drawings of imbedded graphs (currently $O(n^{7/4}\log n)$), or prove a superlinear lower bound (Table 10.3.7).

P14. Reduce the time complexity of bend minimization for planar orthogonal drawings of deg-3 graphs (Table 10.3.7).

10.3.5 Example of a Graph Drawing Algorithm

In this subsection we outline the algorithm by the author [Tam97] for computing, for an imbedded degree-4 graph G, a planar orthogonal grid drawing with minimum number of bends and using $O(n^2)$ area (see Table 10.3.7). This algorithm is the core of a practical drawing algorithm for general graphs (see §10.3.6 and Figure 10.3.3(d)).

Graph Drawing Algorithm The algorithm consists of two main phases:

1. Computation of an orthogonal representation for G, where only the bends and the angles of the orthogonal drawing are defined.
2. Assignment of integer lengths to the segments of the orthogonal representation.

Table 10.3.7 Time complexity of some fundamental graph drawing problems: planar graphs and digraphs.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
planar graph	planar straight-line drawing with prescribed edge lengths	NP-hard	
planar graph	planar straight-line drawing with max angular resolution	NP-hard	
imbedded graph	test the existence of a planar st-line drawing with prescribed angles betw pairs of consecutive edges incident on a vertex	NP-hard	
maximal planar graph	test the existence of a planar st-line drawing with prescribed angles betw pairs of consecutive edges incident on a vertex	$\Omega(n)$	$O(n)$
planar graph	planar st-line grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
planar graph	planar polyline drawing with $O(n^2)$ area, $O(n)$ bends, and $O(1/d)$ angular resolutions	$\Omega(n)$	$O(n)$
triconn planar graph	planar straight-line convex grid drawing with $O(n^2)$ area and $O(1/n^2)$ angular resolution	$\Omega(n)$	$O(n)$
triconn planar graph	planar st-line strictly convex drawing	$\Omega(n)$	$O(n)$
reduced planar *st*-digraph	up planar grid st-line dominance drawing with min area	$\Omega(n)$	$O(n)$
upward planar digraph	up planar polyline grid drawing with $O(n^2)$ area & $O(n)$ bends	$\Omega(n)$	$O(n)$
planar deg-4 graph	planar orthogonal grid drawing with min number of bends	NP-hard	
planar deg-3 graph	planar orthog grid drawing with min # bends and $O(n^2)$ area	$\Omega(n)$	$O(n^5 \log n)$
imbedded deg-4 graph	planar orthog grid drawing with min # bends and $O(n^2)$ area	$\Omega(n)$	$O(n^{\frac{7}{4}} \log n)$
planar deg-4 graph	planar orthog grid drawing with $O(n^2)$ area and $O(n)$ bends	$\Omega(n)$	$O(n)$
planar orthogonal representation	planar orthog grid drawing with minimum area	NP-hard	

Table 10.3.8 Time complexity of some fundamental graph drawing problems: trees.

CLASS OF GRAPHS	PROBLEM	TIME COMPLEXITY	
tree	draw as the Euclidean min spanning tree of a set of points in the plane	NP-hard	
degree-4 tree	minimize area in planar orthogonal grid drawing	NP-hard	
degree-4 tree	minimize total/maximum edge length in planar orthogonal grid drawing	NP-hard	
rooted tree	minimize area in a planar st-line up layered grid drawing that displays symmetries and isom's of subtrees	NP-hard	
rooted tree	minimize area in a planar straight-line up layered drawing that displays symmetries and isom's of subtrees	$\Omega(n)$	$O(n^k)$, $k \geq 1$
binary tree	minimize area in hv-drawing	$\Omega(n)$	$O(n\sqrt{n \log n})$
rooted tree	planar straight-line up layered grid drawing with $O(n^2)$ area	$\Omega(n)$	$O(n)$
rooted tree	planar polyline up grid drawing with $O(n)$ area	$\Omega(n)$	$O(n)$

REMARKS

R9: Phase 1 uses a transformation into a network flow problem (Figure 10.3.3(a–c) below), where each unit of flow is associated with a right angle in the orthogonal drawing. Hence, angles are viewed as a commodity that is produced by the vertices, transported across faces by the edges through their bends, and eventually consumed by the faces.

R10: From the imbedded graph G we construct a flow network N as follows. The nodes of network N are the vertices and faces of G. Let $\deg(f)$ denote the number of edges of the circuit bounding face f. Each vertex v supplies $\sigma(v) = 4$ units of flow, and each face f consumes $\tau(f)$ units of flow, where

$$\tau(f) = \begin{cases} 2\deg(f) - 4 & \text{if } f \text{ is an internal face} \\ 2\deg(f) + 4 & \text{if } f \text{ is the external face} \end{cases}$$

By Euler's formula, $\sum_v \sigma(v) = \sum_f \tau(f)$, i.e., the total supply is equal to the total consumption.

R11: Network N has two types of arcs:

- arcs of the type (v, f), where f is a face incident on vertex v; the flow in (v, f) represents the angle at vertex v in face f, and has lower bound 1, upper bound 4, and cost 0;
- arcs of the type (f, g), where face f shares an edge e with face g; the flow in (f, g) represents the number of bends along edge e with the right angle inside face f, and has lower bound 0, upper bound $+\infty$, and cost 1.

R12: The conservation of flow at the vertices expresses the fact that the sum of the angles around a vertex is equal to 2π. The conservation of flow at the faces expresses the fact that the sum of the angles at the vertices and bends of an internal face is equal to $\pi(p-2)$, where p is the number of such angles. For the external face, the above sum is equal to $\pi(p+2)$.

R13: It can be shown that every feasible flow ϕ in network N corresponds to an admissible orthogonal representation for graph G, whose number of bends is equal to the cost of flow ϕ. Hence, an orthogonal representation for G with the minimum number of bends can be computed from a minimum cost flow in G. This flow can be constructed in $O(n^2 \log n)$ time with standard flow-augmentation methods.

R14: Phase 2 uses a simple compaction strategy derived from VLSI layout, where the lengths of the horizontal and vertical segments are computed independently after a preliminary refinement of the orthogonal representation that decomposes each face into rectangles. The resulting drawing is shown in Figure 10.3.3(d).

10.3.6 Techniques for Drawing Graphs

In this subsection we outline some of the most successful techniques that have been devised for drawing general graphs.

Planarization

The planarization approach is motivated by the availability of many efficient and well-analyzed drawing algorithms for planar graphs (see Table 10.3.7). If the graph is non-planar, it is transformed into a planar graph by means of a preliminary planarization step that replaces each crossing with a fictitious vertex. Finding the minimum number of crossings or a maximum planar subgraph are NP-hard problems. Hence, existing planarization algorithms use heuristics. The best available heuristic for the maximum planar subgraph problem is described in [JM96].

This method has a solid theoretical foundation in polyhedral combinatorics, and achieves good results in practice. A successful drawing algorithm based on the planarization approach and a bend-minimization method [Tam87] is described in [TDB88] (Figure 10.3.3(d) was generated by this algorithm). It has been widely used in software visualization systems.

Layering

The layering approach for constructing polyline drawings of directed graphs transforms the digraph into a layered digraph and then constructs a layered drawing. A typical algorithm based on the layering approach consists of the following main steps:

1. Assign each vertex to a layer, with the goal of maximizing the number of edges oriented upward.
2. Insert fictitious vertices along the edges that cross layers, so that each edge in the resulting digraph connects vertices in consecutive layers. (The fictitious vertices will be displayed as bends in the final drawing.)
3. Permute the vertices on each layer with the goal of minimizing crossings.
4. Adjust the positions of the vertices in each layer with the goal of distributing the vertices uniformly and minimizing the number of bends.

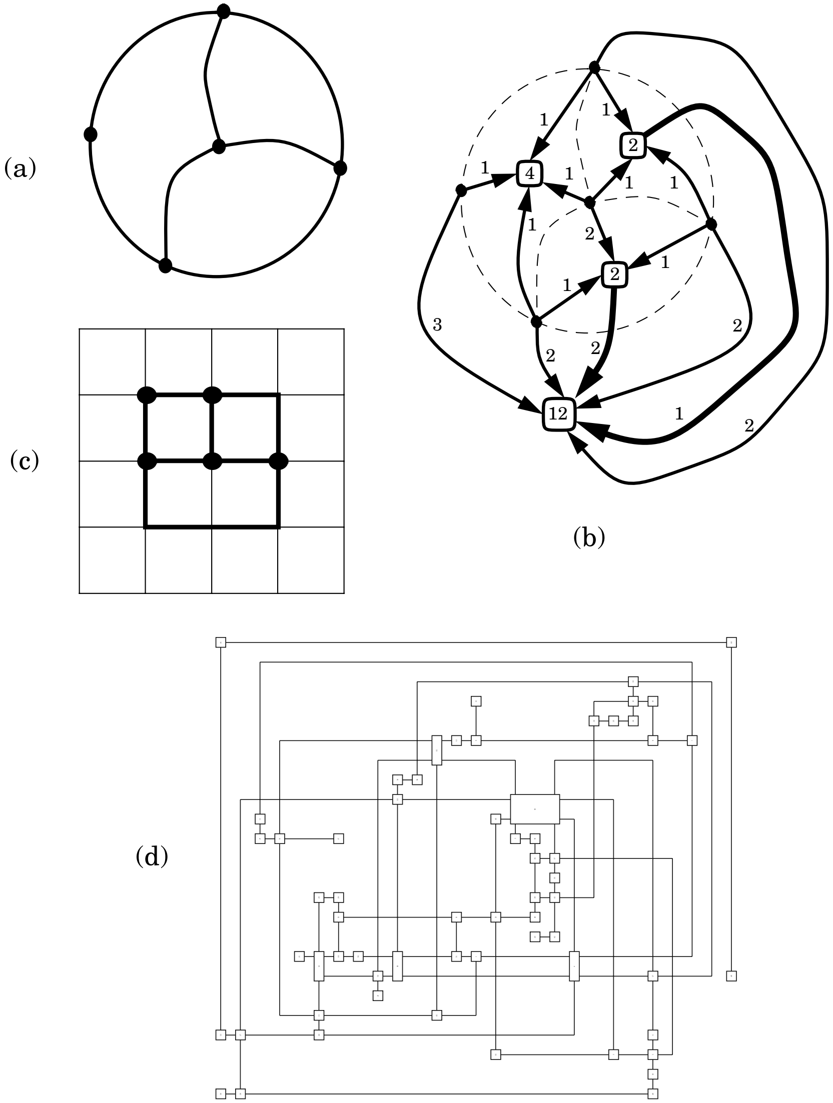


Figure 10.3.3 **(a) Imbedded graph G.**
(b) Minimum cost flow in network N: the flow is shown next to each arc; arcs with zero flow are omitted; arcs with unit cost are drawn with thick lines; a face f is represented by a box labeled with $\tau(f)$.
(c) Planar orthogonal grid drawing of G with minimum number of bends.
(d) Orthogonal grid drawing of a nonplanar graph produced by a drawing method for general graphs based on the algorithm of this subsection.

Most of the subproblems involved in the various steps are NP-hard, hence heuristics must be used. The layering approach was pioneered by Sugiyama et al. [STT81]. The most notable developments of this technique are due to Gansner et al. [GNV88, GKNV93]. For a survey on heuristics for the layering approach see also the paper by Jünger and Mutzel [JM97].

Physical Simulation

This approach uses a physical model where the vertices and edges of the graph are viewed as objects subject to various forces. Starting from an initial random configuration, the physical system evolves into a final configuration of minimum energy, which yields the drawing. Rather than solving a system of differential equations, the evolution of the system is usually simulated using numerical methods (e.g., at each step, the forces are computed and corresponding incremental displacements of the vertices are performed).

Drawing algorithms based on the physical simulation approach are often able to detect and display symmetries in the graph. However, their running time is typically high. The physical simulation approach was pioneered in [Ead84, KS80]. Sophisticated developments and applications include [DH96, EH00, FR91, GGK01, LMP01]. Related topics include declarative methods for graph drawing and approaches to graph drawing based on graph grammars; see, e.g. [CG95, LE95, Bra95].

10.3.7 Recent Research Trends

This subsection presents an overview of selected areas of graph drawing that have recently attracted increasing attention.

Compact 3D Drawings

The increasing demand of visualization algorithms to draw and browse very large networks makes it natural to investigate how much benefit can be obtained from the third dimension to represent the overall structure of a huge graph in a small portion of a virtual 3D environment. While the problem of computing small-sized crossing-free straight-line drawings in the plane has a long tradition, its 3D counterpart has become the subject of much attention only in recent years.

FACTS

F19: Chrobak, Goodrich, and Tamassia [CGT96] gave an algorithm for constructing 3D convex drawings of triconnected planar graphs with $O(n)$ volume and non-integer coordinates.

F20: Cohen, Eades, Lin and Ruskey [CELR97] showed that every graph admits a straight-line crossing-free 3D drawing on an integer grid of $O(n^3)$ volume, and proved that this is asymptotically optimum.

F21: Calamoneri and Sterbini [CS97] showed that all 2-, 3-, and 4-colorable graphs can be drawn in a 3D grid of $O(n^2)$ volume with $O(n)$ aspect ratio and proved a lower bound of $\Omega(n^{1.5})$ on the volume of such graphs.

F22: For r-colorable graphs, Pach, Thiele and Tóth [PTT97] showed a bound of $\theta(n^2)$ on the volume.

F23: Garg, Tamassia, and Vocca [GTV96] showed that all 4-colorable graphs (and hence all planar graphs) can be drawn in $O(n^{1.5})$ volume and with $O(1)$ aspect ratio by using a grid model where the coordinates of the vertices may not be integer.

F24: Felsner, Liotta, and Wismath [FGW01] showed that all outerplanar graphs can be drawn in a restricted integer 3D grid of linear volume consisting of three parallel lines at distance one from each other.

F25: Dujmović, Morin, and Wood [DMW02] present $O(n \log^2 n)$ volume drawings of graphs with bounded tree-width and $O(n)$ volume for graphs with bounded path-width.

F26: Wood [Woo02] shows that also graphs with bounded queue number have 3D straight-line grid drawings of $O(n)$ volume.

F27: A result by Dujmović and Wood [DWo03a] shows that linear volume can also be achieved for graphs with bounded tree-width; they show 3D straight-line grid drawings of volume $c \times n$ for these graphs, where c is a constant whose value exponentially depends on the tree-width.

F28: Di Giacomo, Liotta, and Wismath [DLW02a, DLW02b] show $4 \times n$ and $32 \times n$ volume for two subclasses of series-parallel graphs. The problem of computing straight-line 3D drawings of planar graphs on an integer grid of $O(n)$ volume is still open.

Graph Drawing Checkers

The intrinsic structural complexity of the implementation of geometric algorithms makes the problem of formally proving the correctness of the code unfeasible in most of the cases. This has been motivating the research on *checkers.*

DEFINITIONS

D50: A ***drawing checker*** is an algorithm that receives as input a geometric structure and a predicate stating a property that should hold for the structure. The task of the checker is to verify whether the structure satisfies or not the given property.

The expectation is that it is often easier to evaluate the quality of the output than the correctness of the software that produces it. Different papers (see, e.g., [DLPT98, MSNS$^+$99]) have agreed on the basic features that a “good” checker should have:

D51: ***Correctness:*** The checker should be correct beyond any reasonable doubt. Otherwise, one would incur in the problem of checking the checker.

D52: ***Simplicity:*** The implementation should be straightforward.

D53: ***Efficiency:*** The expectation is to have a checker that is not less efficient than the algorithm that produces the geometric structure.

D54: ***Robustness:*** The checker should be able to handle degenerate configurations of the input and should not be affected by errors in the flow of control due to round-off approximations.

REMARKS

R15: Checking is especially relevant in the graph drawing context. Indeed, graph drawing algorithms are among the most sophisticated of the entire computational geometry field, and their goal is to construct complex geometric structures with specific properties. Also, because of their immediate impact on application areas, graph drawing algorithms are usually implemented right after they have been devised. Further, such implementations are often available on the Web without any certification of their correctness.

R16: Of course, the checking problem becomes crucial when the drawing algorithm deals with very large data sets, when a simple complete visual inspection of the drawing is difficult or unfeasible. Devising graph drawing checkers involves answering only apparently innocent questions like: “is this drawing planar?” or “is this drawing upward?” or “are the faces convex polygons?”.

R17: The problem of checking the planarity of a subdivision has been independently studied by Mehlhorn et al. [MSNS$^+$99] and by Devillers et al. [DLPT98]. In these papers linear time algorithms are given to check the planarity of a subdivision composed by convex faces. Di Battista and Liotta [DL98] check the upward planarity of straight-line oriented drawings that may also have non-convex faces.

Incremental Graph Drawing

In several applications, such as software engineering and database design, users interact extensively with a displayed graph, continuously adding or deleting vertices and edges. Under such a scenario, a graph drawing system should update the drawing each time the displayed graph is modified by the user. Unfortunately, traditional drawing algorithms may not be suitable in these situations. Since they typically construct a drawing from scratch, they may fail to update the drawing quickly after the user modifies the displayed graph.

REMARKS

R18: The new drawing constructed after a modification may be significantly different from the previous one, even if only a small change has been made in the displayed graph. In this case, the user's *mental map* [ELMS95], that is, the mental image the user has of the graph, is not preserved, and a considerable cognitive effort is required to correlate the new drawing and the previous one. Bridgeman and Tamassia [BT00a] formulate and validate several difference metrics that can be used to measure how much a drawing algorithm changes the user's mental map in an interactive environment. Tradeoffs between running time, optimization of the drawing properties, and preservation of the mental map are typical issues to be addressed in incremental graph drawing.

R19: Several papers dealing with these issues have recently appeared in journal and conference proceedings. A limited list includes the work by Cohen, Di Battista, Tamassia, and Tollis [CDTT95] on data structures for dynamic graph drawing; the Bayesian framework of Brandes and Wagner [BW97]; the definition and experimental investigation of four scenarios for interactive orthogonal graph drawing by Papakostas and Tollis [PT98] and Papakostas, Six, and Tollis [PST97]; the papers on interactive orthogonal graph drawing by Biedl and Kaufmann [BK97], and by Brandes and Wagner [BW98]; the fully dynamic algorithms for orthogonal drawings in 3D-space by Closson, Gartshore, Johansen, and Wismath [CGJW00]; and the work by North and Woodhull [NW02] on online hierarchical graph drawing.

Experimentation

Many graph drawing algorithms have been implemented and used in practical applications. Most papers show sample outputs, and some also provide limited experimental results on small test suites. However, in order to evaluate the practical performance of a graph drawing algorithm in visualization applications, it is essential to perform extensive experimentations with input graphs derived from the application domain and over a large set of aesthetic requirements that are desirable for the user to have in the drawing.

FACTS

F29: Among papers that test the human perception of the aesthetic properties of graph drawing we mention the work by Purchase, Allder, and Carrington [PAC02] and by Bridgeman and Tamassia in an incremental setting [BT00b].

F30: The first broad-view experimental study on graph drawing algorithms, due to Himsolt [Him95], presents a comparative study of twelve graph drawings algorithms based on various approaches.

F31: Di Battista et al. [DGL+97] report on an extensive experimental study comparing four orthogonal drawing algorithms based on the planarization approach. The test data are 11,582 graphs, ranging from 10 to 100 vertices, which are generated from a core set of 112 graphs used in "real-life" software engineering and database applications. A similar experimental setting is then used to analyze the performance of four graph drawing algorithms for directed acyclic graphs in [DGL+00].

F32: Heuristics for computing orthogonal drawings with good area performance are experimentally validated in the works by Klau, Klein, and Mutzel [KKM01] and by Di Battista et al. [BDD+00].

F33: Experimentation of graph drawing techniques for computing graphical representations of database schemas are conducted by Di Battista, Didimo, Patrignani, and Pizzonia [DDPP02].

F34: An extensive experimental comparison of five algorithms based on force-directed and randomized methods is described in the work by Brandenburg, Himsolt, and Roher [BHR96].

F35: Jünger and Mutzel [JM97] experimentally compare the performance of eight heuristics for straight-line drawings of 2-layer graphs.

F36: An extensive survey on experimental studies on graph drawing can be found in [VBL+00].

Fixed Parameter Tractability

Recently, the theory of parametrized complexity [DF97] has been applied with success to some computationally difficult graph drawing problems.

DEFINITION

D55: A problem Π specified in terms of one or more parameters is ***fixed parameter tractable***, or in the ***FPT class***, if there is an algorithm that solves Π in $O(f(k) \cdot n^c)$ time, where n is the input size, k is the parameter size, c is a constant, and f is an arbitrary function dependent only on parameter k.

FACTS

F37: It is NP-complete to decide, given a graph G and a positive integer k, whether G can be drawn on the plane with at most k edge crossings (edges are drawn as simple curves). However, it has been shown by Grohe [Gro01] that this problem is fixed-parameter tractable since there exists a quadratic time algorithm that solves it for any fixed value of k.

F38: Other relevant NP-hard graph drawing problems have been proved to be in the FPT class [DFH$^+$01a, DFH$^+$01b, DW]; some of these results are summarized in Table 10.3.9. In the table, h and k denote integer positive constants.

It has to be remarked, however, that the hidden constants in the time complexities shown in Table 10.3.9 below may heavily depend on the values of the parameter. For example, the crossing minimization problem for general graph has time complexity $O(f(k) \cdot n^2)$ where $f(k)$ is a doubly exponential function [Gro01]. Thus, not only proving that computationally hard graph drawing problems are fixed parameter tractable is an emerging research topic, but also finding time complexity bounds that can be of practical use for FPT problems.

Table 10.3.9 Some NP-hard graph drawing problems that are fixed-parameter-tractable.

CLASS OF GRAPHS	NP-HARD PROBLEM	TIME COMPLEXITY
2-layered graph	2-*layers planarization*: remove at most k edges so that the resulting graph is biplanar	$O(f(k) + \|G\|)$
2-layered graph	2-*layers crossing minimization*: compute a straight-line drawing of the graph on two layers with at most k crossings	$O(f(k) \cdot n^2)$
general graph	h-*layers planarization*: remove at most k edges so that the resulting graph is h-level planar	$O(f(h,k) \cdot n)$
general graph	h-*layers crossing minimization*: compute a straight-line drawing of the graph on h layers with at most k crossings	$O(f(h,k) \cdot n^2)$
general graph	*crossing minimization*: compute a straight-line drawing of the graph with at most k crossings	$O(f(k) \cdot n^2)$

10.3.8 Sources and Related Material

Three books devoted to graph drawing are published [DETT99, KE01, Sug02]. The proceedings of the annual Symposium on *Graph Drawing* are published by Springer-Verlag in the *Lecture Notes in Computer Science* series (volumes 2265, 1984, 1731, 1547, 1353, 1190, 1027, 894).

REMARKS

R20: Surveys on various aspects of graph drawing appear in [DLL95, DPS02, GT95, HMM00, JM97, Riv93, San99, SSV95, Tam90a, Tam90b, Tam99, VBL$^+$00]. Special issues devoted to graph drawing have appeared in *Algorithmica* (vol. 16, no. 1, 1996), *Computational Geometry: Theory and Applications* (vol. 9, no. 1-2, 1998), the *Journal of Visual Languages and Computing* (vol. 6, 1995), and the *Journal of Graph Algorithms and Applications* (vol. 3, no. 4, 1999; vol. 4, no. 3, 2000; vol. 6, no. 1, 2002; vol. 6, no.3, 2002).

R21: Sites with pointers to graph drawing resources and tools include the WWW page maintained by Tamassia (`http://www.cs.brown.edu/people/rt/gd.html`) and the WWW page maintained by Brandes (`http://graphdrawing.org/`).

References

[AAC$^+$96] O. Aichholzer, F. Aurenhammer, S-W Chen, N. Katoh, M. Taschwer, G. Rote, and Y-F Xu, Triangulations intersect nicely, *Discrete Comput. Geom.* 16 (1996), 339–359.

[BBM99] T. Biedl, A. Bretscher, and H. Meijer, Rectangle of influence drawings of graphs without filled 3-cycles, pp 359–368 in Graph Drawing (Proc. GD '99) (ed. J. Kratochvil), *Lecture Notes Comput. Sci.* vol. 1731, Springer-Verlag, 1999.

[BDD$^+$00] S. S. Bridgeman, G. Di Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara, Turn-regularity and optimal area drawings of orthogonal representations, *Computational Geometry: Theory and Applications* 16 (2000), 53–93.

[BHR96] F. Brandenburg, M. Himsolt, and C. Roher, An experimental comparison of force-directed and randomized graph drawing algorithms, pp 76–87 in Graph Drawing (Proc. GD '95) (ed. by F. J. Brandenburg), *Lecture Notes Comput. Sci.* vol. 1027, Springer-Verlag, 1996.

[BK97] T. Biedl and M. Kaufmann, Area-efficient static and incremental graph drawings, pp 37–52 in European Sympos. on Algorithms (Proc. ESA '97), *Lecture Notes Comput. Sci.*, vol. 1284, Springer-Verlag, 1997.

[BLL96] P. Bose, W. Lenhart, and G. Liotta, Characterizing proximity trees, *Algorithmica* 16 (1996), 83–110. (special issue on Graph Drawing, ed. by G. Di Battista and R. Tamassia).

[Bra95] F. J. Brandenburg, Designing graph drawings by layout graph grammars, pp 416–427 in Graph Drawing (Proc. GD '94), *Lecture Notes Comput. Sci.* vol. 894, Springer-Verlag, 1995.

[BT00a] S. Bridgeman and R. Tamassia, Difference metrics for interactive orthogonal graph drawing algorithms, *J. of Graph Algorithms and Applications* 4 (2000), 47–74.

[BT00b] S. Bridgeman and R. Tamassia, A user study in similarity measures for graph drawing, pp 19–30 in Graph Drawing (Proc. GD 2000), *Lecture Notes Comput. Sci.* vol. 1984, Springer-Verlag, 2000.

[BW97] U. Brandes and D. Wagner, A Bayesian paradigm for dynamic graph layout, pp 236–247 in Graph Drawing (Proc. GD '97) (ed. by G. Di Battista), *Lecture Notes Comput. Sci.* vol. 1353, Springer-Verlag, 1997.

[BW98] U. Brandes and D. Wagner, Dynamic grid embedding with few bends and changes, pp 89–98 in Ann. Intl. Symp. Algorithms and Computation (Proc. ISAAC '98), *Lecture Notes Comput. Sci.* vol. 1533, Springer-Verlag, 1998.

[CDTT95] R. F. Cohen, G. Di Battista, R. Tamassia, and I. G. Tollis, Dynamic graph drawings: Trees, series-parallel digraphs, and planar st-digraphs, *SIAM J. on Computing* 24 (1995), 970–1001.

[CELR97] R. F. Cohen, P. Eades, T. Lin, and F. Ruskey, Three-dimensional graph drawing, *Algorithmica* 17 (1997), 199–208.

[CG95] I. F. Cruz and A. Garg, Drawing graphs by example efficiently: Trees and planar acyclic digraphs, pp 404–415 in *Graph Drawing (Proc. GD '94)* (ed. by R. Tamassia and I. G. Tollis), *Lecture Notes Comput. Sci.* vol. 894, Springer-Verlag, 1995.

[CGJW00] M. Closson, S. Gartshoer, J. Johansen, and S. K. Wismath. Fully dynamic 3-dimensional orthogonal graph drawing, *J. of Graph Algorithms and Appl.* 5 (2000), 1–34.

[CGT96] M. Chrobak, M. T. Goodrich, and R. Tamassia, Convex drawings of graphs in two and three dimensions, pp 319–328 in Proc. 12th Annu. ACM Sympos. Comput. Geom., 1996.

[CS97] T. Calamoneri and A. Sterbini, Drawing 2-, 3-, and 4-colorable graphs in $o(n^2)$ volume, pp 53–62 in Graph Drawing (Proc. GD '96) (ed. by S. North), *Lecture Notes Comput. Sci.* vol. 1190, Springer-Verlag, 1997.

[CX01] Siu-Wing Cheng and Yin-Feng Xu, On -skeleton as a subgraph of the minimum weight triangulation, *Theoretical Computer Science* 262 (2001), 459–471.

[DDPP02] G. Di Battista, W. Didimo, M. Patrignani, and M. Pizzonia, Drawing database schemas, *Software–Practice and Experience* 32 (2002), 1065–1098.

[DETT99] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing*, Prentice-Hall, 1999.

[DF97] R. G. Downey and M. R. Fellows, *Parameterized Compexity*, Springer, 1997.

[DFH$^+$01a] V. Dujmović, M. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Radge, F. Rosamond, m. Suderman, S. Whitesides, and D. R. Wood, On the parameterized complexity of layered graph drawing, pp 488–499 in European Sympos, on Algorithms (Proc, ESA 2001), *Lecture Notes Comput. Sci.* vol. 2161, Springer-Verlag, 2001.

[DFH$^+$01b] V. Dujmović, M. Fellows, M. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Radge, F. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood, A fixed parameter approach to two-layer crossing minimization, pp 1–15 in

Graph Drawing (Proc. GD 2001) (ed. by P. Mutzel, M. Jünger, and S. Leipert), *Lecture Notes Comput. Sci.* vol. 2265, Springer-Verlag, 2001.

[DGL$^+$97] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara, An experimental comparison of four graph drawing algorithms, *Computational Geometry: Theory and Applications* 7 (1997), 303–325.

[DGL$^+$00] G. Di Battista, A. Garg, G. Liotta, A. Parise, R. Tamassia, E. Tassinari, F. Vargiu, and L. Vismara, Drawing directed acyclic graphs: An experimental study, *International Journal on Computational Geometry and Applications* 10 (2000), 623–648.

[DH96] R. Davidson and D. Harel, Drawing graphics nicely using simulated annealing, *ACM Transactions on Graphics* 15 (1996), 301–331.

[Dil90a] M. B. Dillencourt, Realizability of Delaunay triangulations, *Inform. Process. Lett.* 33 (1990), 283–287.

[Dil90b] M. B. Dillencourt, Toughness and Delaunay triangulations, *Discrete Comput. Geom.* 5 (1990), 575–601.

[DL98] G. Di Battista and G. Liotta, Upward planarity checking "faces are more than polygons", in Graph Drawing (Proc. GD '98) (ed. by S.H. Whitesides), *Lecture Notes Comput. Sci.*, Springer-Verlag, 1998.

[DLL95] G. Di Battista, W. Lenhart, and G. Liotta, Proximity drawability: a survey, pp 328–339 in Graph Drawing (Proc. GD '94) (ed. by R. Tamassia and I. G. Tollis), *Lecture Notes Comput. Sci.* vol. 894, Springer-Verlag, 1995.

[DLPT98] O. Devillers, G. Liotta, F. P. Preparata, and R. Tamassia, Checking the convexity of polytopes and the planarity of subdivisions, *Comput. Geom. Theory Appl.* 11 (1998), 187–208.

[DLW02a] E. Di Giacomo, G. Liotta, and S.K. Wismath, Drawing series-parallel graphs on a box, in 14th Canadian Conference On Computational Geometry (CCCG '02), 2002.

[DLW02b] E. Di Giacomo, G. Liotta, and S.K. Wismath, The k-lines drawability problem for series-parallel graphs, Technical report, Dept, Computer Science, University of Lethbridge, 2002.

[DM96] M. T. Dickerson and M. H. Montague, A (usually?) connected subgraph of the minimum weight triangulation, pp 204–213 in *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, 1996.

[DMW02] V. Dujmović, P. Morin, and D. R. Wood, Pathwidth and three-dimensional straight line grid drawings of graphs, pp 42–53 in Graph Drawing (Proc, GD'02) (ed. by M. Goodrich and S. G. Kobourov), *Lecture Notes Comput. Sci.* vol. 2528, Springer-Verlag, 2002.

[DPS02] J. Diaz, J. Petit, and M. Serna, A survey of graph layout problems, *ACM Computing Surveys* 34 (2002), 313–356.

[DS94] M. B. Dillencourt and W. D. Smith, Graph-theoretical conditions for inscribability and Delaunay realizability, pp 287–292 in *Proc. 6th Canad. Conf. Comput. Geom.*, 1994.

[DV96] G. Di Battista and L. Vismara, Angles of planar triangular graphs, *SIAM J. Discrete Math.* 9 (1996), 349–359.

[DW] V. Dujmović and S. Whitesides, An efficient fixed parameter tractable algorithm for 1-sided crossing minimization, in Graph Drawing (Proc. GD 2002), *Lecture Notes Comput. Sci* 2528 (2002), Springer-Verlag, 118–119.

[DW03a] V. Dujmović and D. R. Wood, Tree-partitions of k-trees with application in graph layout, in Workshop on Graph Theoretic Concepts in Computer Science (Proc. WG '03), *Lecture Notes Comput. Sci.*, Springer-Verlag, to appear.

[DW03b] V. Dujmović and D. R. Wood, New resuls in graph layout, Technical Report TR-2003-04, School of Computer Science, Carleton University, Ottawa, Canada.

[Ead84] P. Eades, A heuristic for graph drawing, *Congr. Numer.* 42 (1984), 149–160.

[EH00] P. Eades and M. L. Huang, Navigating clustered graphs using force-directed methods, *Journal of Graph Algorithms and Applications* 4 (2000), 157–181.

[ELMS95] P. Eades, W. Lai, K. Misue, and K. Sugyiama, Layout adjustment and the mental map, *Journal of Visual Languages and Computing* 6 (1995), 183–210.

[EW96] P. Eades and S. Whitesides, The realization problem for Euclidean minimum spanning trees is NP-hard, *Algorithmica* 16 (1996), 60–82. (Special issue on Graph Drawing, ed. by G. Di Battista and R. Tamassia.)

[FGW01] S. Felsner, G.Liotta, and S. Wismath, Straight line drawings on restricted integer grids in two and three dimensions, pp 328–342 in *Graph Drawing (Proc. GD 2001)* (ed. by P. Mutzel, M. Jünger, and S. Leipert), *Lecture Notes Comput. Sci.* vol. 2265, Springer-Verlag, 2001.

[FR91] T. Fruchterman and E. Reingold, Graph drawing by force-directed placement, *Software–Practice and Experience* 21 (1991), 1129–1164.

[GGK01] P. Gajer, M. T. Goodrich, and S. G. Kobourov, A multi-dimensional approach to force-directed layout of large graphs, pp 211–221 in Graph Drawing (Proc. GD 2000) (ed. by Joe Marks), *Lecture Notes Comput. Sci.* vol. 1984, Springer-Verlag, 2001.

[GJ79] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.

[GKNV93] E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo, A technique for drawing directed graphs, *IEEE Transactions on Software Engineering* 19 (1993), 214–230.

[GNV88] E. R. Gansner, S. C. North, and K. P. Vo. DAG – A program that draws directed graphs, *Software–Practice and Experience* 18 (1988), 1047–1062.

[Gro01] M. Grohe, Computing crossing numbers in quadratic time, pp 231–236 in *Symposium on the Theory of Computing (Proc. STOC 2001)*, 2001.

[GT95] A. Garg and R. Tamassia, Upward planarity testing, *Order* 12 (1995), 109–133.

[GTV96] A. Garg, R. Tamassia, and P. Vocca, Drawing with colors, pp 12–26 in Proc. 4th Annu. European Sympos. Algorithms, *Lecture Notes Comput. Sci.* vol. 1136, Springer-Verlag, 1996.

[Him95] M. Himsolt, Comparing and evaluating layout algorithms within GraphEd, *Journal of Visual Languages and Computing* 6 (1995), 255–273. (Special issue on Graph Visualization, ed. by I. F. Cruz and P. Eades.)

[HMM00] I. Herman, G. Melancon, and M. S. Marshall, Graph visualization and navigation in information visualizaion: A survey, *IEEE Transactions on Visualization and Computer Graphics* 6 (2000), 24–43.

[JM96] M. Jünger and P. Mutzel, Maximum planar subgraphs and nice embeddings: Practical layout tools, *Algorithmica* 16 (1996). (special issue on Graph Drawing, ed. by G. Di Battista and R. Tamassia).

[JM97] M. Jünger and P. Mutzel, 2-layer straightline crossing minimization: performance of exact and heuristics algorithms, *J. of Graph Algorithms and Applications* 1 (1997), 1–25.

[JT92] J. W. Jaromczyk and G. T. Toussaint, Relative neighborhood graphs and their relatives, *Proc. IEEE* 80 (1992), 1502–1517.

[KE01] M. Kaufmann and D. Wagner (Eds.), *Drawing Graphs Methods and Models*, Springer-Verlag, 2001.

[Kei94] M. Keil, Computing a subgraph of the minimum weight triangulation, *Comput. Geom. Theory Appl.* 4 (1994), 13–26.

[Kir80] D. G. Kirkpatrick, A note on Delaunay and optimal triangulations, *Inform. Process. Lett.* 10 (1980), 127–128.

[KKM01] G. Klau, K. Klein, and P. Mutzel, An experimental comparison of orthogonal compaction algorithms, pp 37–51 in Graph Drawing (GD 2000) (ed. by Joe Marks), *Lecture Notes Comput. Sci.* vol. 1984, 2001.

[KS80] J. B. Kruskal and J. B. Seery, Designing network diagrams, pp 22–50 in *Proc. First General Conference on Social Graphics*, U. S. Dept. of the Census, 1980.

[LD95] Giuseppe Liotta and Giuseppe Di Battista, Computing proximity drawings of trees in the 3-dimensional space, pp 239–250 in Proc. 4th Workshop Algorithms Data Struct., *Lecture Notes Comput. Sci.* vol. 955, Springer-Verlag, 1995.

[LE95] T. Lin and P. Eades, Integration of declarative and algorithmic approaches for layout creation, pp 376–387 in Graph Drawing (Proc. GD '94) (ed. by R. Tamassia and I. G. Tollis), *Lecture Notes Comput. Sci.* vol. 894, Springer-Verlag, 1995.

[Lin87] A. Lingas, A new heuristic for minimum weight triangulation, *SIAM J. Algebraic Discrete Methods* 8 (1987), 646–658.

[LK96] C. Levcopoulos and D. Krznaric, Tight lower bounds for minimum weight triangulation heuristic, *Inform. Process. Lett.* 57 (1996), 129–135.

[LK98] C. Levcopoulos and D. Krznaric, A linear-time approximation schema for minimum weight triangulation of convex polygons, *Algorithmica* 21 (1998), 285–311.

[LL96] W. Lenhart and G. Liotta, Drawing outerplanar minimum weight triangulations, *Inform. Process. Lett.* 6 (1996), 253–260.

[LL97] W. Lenhart and G. Liotta, Proximity drawings of outerplanar graphs, pp. 286–302 in Graph Drawing (Proc. GD '96) (ed. by S. North), *Lecture Notes Comput. Sci.* vol. 1190, Springer-Verlag, 1997.

[LL02] W. Lenhart and G. Liotta, The drawability problem for minimum weight triangulations, *Theoret. Comp. Sci.* 270 (2002), 261–286.

[LLMW98] G. Liotta, A. Lubiw, H. Meijer, and S. H. Whitesides, The rectangle of influence drawability problem, *Comput. Geom. Theory Appl.* 10 (1998), 1–22.

[LM03] G. Liotta and H. Meijer, Voronoi drawings of trees, *Comput. Geom. Theory Appl.* 24 (2003), 147–178.

[LMP01] N. Leash, J. Marks, and M. Patrignani, Interactive partinioning, pp 31–36 in Graph Drawing (Proc. GD 2000), *Lecture Notes Comput. Sci.* vol. 1984, Springer-Verlag, 2001.

[LS93] A. Lubiw and N. Sleumer, Maximal outerplanar graphs are relative neighborhood graphs, pp 198–203 in *Canadian Conference on Computational Geometry (Proc. CCCG '93)*, 1993.

[MDM97] J. Keil M. Dickerson and M. Montague, A large subgraph of the minimum weight triangulation, *Discrete and Computational Geometry* 18 (1997), 289–304.

[MS92] C. Monma and S. Suri, Transitions in geometric minimum spanning trees, *Discrete Comput. Geom.* 8 (1992), 265–293.

[MSNS$^+$99] K. Mehlhorn, T. Schilz, S. Näher, S. Schirra, M. Seel, R. Seidel, and C. Uhrig, Checking geometric programs or verification of geometric structures, *Comput. Geom. Theory Appl.* 12 (1999), 85–113.

[MZ79] G. K. Manacher and A. L. Zobrist, Neither the greedy nor the Delaunay triangulation of a planar point set approximates the optimal triangulation, *Inform. Process. Lett.* 9 (1979), 31–34.

[NW02] S. C. North and G. Woodhull, Online hierarchical graph drawing, pp 232–246 in Graph Drawing (Proc. GD 2001) (ed. by P. Mutzel, M. Jünger, and S. Leipert), *Lecture Notes Comput. Sci.* vol. 2265, Springer-Verlag, 2002.

[OBS92] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, 1992.

[PAC02] H. C. Purchase, J. A. Allder, and D. Carrington, Aesthetics in UML diagrams: User preferences, *J. of Graph Algorithms and Applications* 6 (2002), 255–279.

[PS90] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, 3rd edition, 1990.

[PST97] A. Papakostas, J. Six, and I. G. Tollis, Experimental and theoretical results in interactive orthogonal graph drawing, pp 371–386 in Graph Drawing (Proc. GD '96) (ed. by S. North), *Lecture Notes Comput. Sci.* vol. 1190, Springer-Verlag, 1997.

[PT98] A. Papakostas and I. G. Tollis, Interactive orthogonal graph drawing, *IEEE Trans. on Computers* C-47(11) (1998), 83–110.

[PTT97] J. Pach, T. Thiele, and G. Tóth, Three-dimensional grid drawings of graphs, pp 47–51 in Graph Drawing (Proc. GD '97) (ed. by G. Di Battista), *Lecture Notes Comput. Sci.* vol. 1353, Springer-Verlag, 1997.

[Riv93] I. Rival, Reading, drawing, and order, pp 359–404 in *Algebras and Orders* (I. G. Rosenberg and G. Sabidussi, editors), Kluwer Academic Publishers, 1993.

[RM92] D. Rappaport and H. Meijer, Computing the minimum weight triangulation of a set of linearly ordered points, *Information Processing Letters* (1992), 35–38.

[San99] G. Sander, Graph layout for applications in compiler construction, *Theoretical Computer Science* 217 (1999), 175–214.

[SSV95] F. Shahrokhi, L. A. Székely, and I. Vrt'o, Crossing numbers of graphs, lower bound techniques and algorithms: a survey, pp 131–142 in Graph Drawing (Proc. GD '94) (ed. by R. Tamassia and I. G. Tollis), *Lecture Notes Comput. Sci.* vol. 894, Springer-Verlag, 1995.

[STT81] K. Sugiyama, S. Tagawa, and M. Toda, Methods for visual understanding of hierarchical systems, *IEEE Trans. Syst. Man Cybern.* SMC-11(2) (1981), 109–125.

[Sug02] K. Sugyiama, *Graph Drawing and Applications for Software and Knowledge Engineers*, World Scientific, 2002.

[Tam87] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, *SIAM Journal on Computing* 16 (1987), 421–444.

[Tam90a] R. Tamassia, Drawing algorithms for planar st-graphs, *Australasian J. of Combinatorics* 2 (1990), 217–235.

[Tam90b] R. Tamassia, Planar orthogonal drawings of graphs, in *Proc. IEEE Internat. Sympos. on Circuits and Systems*, 1990.

[Tam99] R. Tamassia, Advances in the theory and practice of graph drawing, *Theoretical Computer Science* 17 (1999), 235–254.

[TDB88] R. Tamassia, G. Di Battista, and C. Batini, Automatic graph drawing and readability of diagrams, *IEEE Trans. Syst. Man Cybern.* SMC-18(1) (1988), 61–79.

[VBL$^+$00] L. Vismara, G. Di Battista, G. Liotta, R. Tamassia, and F. Vargiu, Experimental studies on graph drawing algorithms, *Software–Practice and Experience* 30 (2000), 1235–1284.

[WCY00] Cao An Wang, F. Y. Chin, and B. Yang, Triangulations without minimum weight drawing, pp 163–173 in Algorithms and Complexity (Proc. CIAC 2000), *Lecture Notes Comput. Sci.* vol. 1767, Springer-Verlag, 2000.

[Woo02] D. R. Wood, Queue layouts, tree-width, and three-dimensional graph drawing, pp 348–359 in 22nd Foundations of Software Technology and Theoretical Comput. Sci. (FSTTCS '02), *Lecture Notes Comput. Sci.* vol. 2556, Springer-Verlag, 2002.

10.4 ALGORITHMS ON RECURSIVELY CONSTRUCTED GRAPHS

R. B. Borie, University of Alabama
R. Gary Parker, Georgia Institute of Technology
C. A. Tovey, Georgia Institute of Technology

10.4.1 Algorithms on Trees
10.4.2 Algorithms on Series-Parallel Graphs
10.4.3 Algorithms on Treewidth-k Graphs
10.4.4 Algorithms on Cographs
10.4.5 Algorithms on Cliquewidth-k Graphs
10.4.6 Algorithms on k-HB Graphs
References

Introduction

In this section we demonstrate algorithms for the recursively defined classes of trees, series-parallel graphs, treewidth-k graphs, cographs, cliquewidth-k graphs, and k-HB graphs. For convenience, definitions of these graph classes, which appear in §2.4 (among others given there), are repeated here. Our emphasis in this section is on the solution *technique* for each graph class and less so on the variety of problems solvable *within* each class per se. Accordingly, for each graph class, we will consider the problem of finding an *independent set* in a given graph (§5.3); where it is natural to do so, some additional problems are solved as well. For a much broader accounting using this same format and one that includes a host of other problems such as *clique*, *dominating set*, *vertex coloring*, *matching*, *hamiltonian cycle*, *hamiltonian path*, and others, the reader is directed to [BoPaTo02].

DEFINITIONS

D1: A ***recursively constructed graph class*** is defined by a set (usually finite) of primitive or ***base graphs***, in addition to one or more operations (called ***composition rules***) that compose larger graphs from smaller subgraphs. Each operation involves fusing specific vertices from each subgraph or adding new edges between specific vertices from each subgraph.

D2: Each graph in a recursive class has a corresponding ***decomposition tree*** that shows how to build it from base graphs.

D3: For a graph G, a set S of vertices is an ***independent set*** if no two vertices in S are adjacent.

Algorithm Design Strategy

Efficient algorithms for problems restricted to recursively constructed graph classes typically employ a *dynamic programming* approach as follows: first solve the problem on the base graphs defined for the given class; then combine the solutions for subgraphs into a solution for a larger graph that is formed by the specific composition rules that govern construction of members in the class. A linear-time algorithm is achieved by determining a finite number of equivalence classes that correspond to each node in a member graph's decomposition tree. The number of such equivalence classes is constant with respect to the size of the input graph, but may depend upon a parameter (k) associated with the class. A polynomial algorithm can often be created if the number of equivalence classes required for the problem grows only polynomially with input graph size. Also key is that a graph's decomposition tree be given or be computable efficiently.

NOTATION: In the descriptions of the algorithms in this section, we use $G.x$ to denote an attribute (or class) of a given graph G. When we write $G.x =$ *maximum-cardinality independent set*, then $G.x$ carries with it two pieces of information: the *size* of a maximum-cardinality independent set, and *one particular instance* of such a set. Moreover, these two pieces of information are carried forward in computations and assignments involving $G.x$.

REMARK

R1: Note that when the simplest version of a problem (cardinality or existence) can be solved using a dynamic programming approach, then other more complicated versions (involving vertex or edge weights, counting, bottleneck, min-max, etc.) can generally be routinely solved. Following, we begin with the primitive recursive class of trees.

10.4.1 Algorithms on Trees

DEFINITION

D4: The graph with a single vertex r (and no edges) is a ***tree*** with root r (the sole base graph). Let (G, r) denote a tree with root r. Then $(G_1, r_1) \oplus (G_2, r_2)$ is a tree formed by taking the disjoint union of G_1 and G_2 and adding an edge (r_1, r_2). The root of this new tree is $r = r_1$.

TERMINOLOGY NOTE: Technically, the pairs (G, r) in Definition 4 denote *rooted trees*. However, the specification of distinguished vertices r_1 and r_2 (and hence r) is relevant here only as a vehicle in the recursive construction.

NOTATION: Given any tree (or subtree) G, the designated root is denoted by $root[G]$.

Maximum-Cardinality Independent Set in a Tree

FACTS

F1: Any independent set of vertices either includes $root[G]$ or does not. Whether or not independent sets of two trees G_1, G_2 can be combined into an independent set of the tree $G_1 \oplus G_2$ depends only on these inclusions.

NOTATION: The following notation is used in the description of Algorithm 10.4.1 below.

- $G.a =$ max-cardinality independent set that includes $root[G]$.
- $G.b =$ max-cardinality independent set that excludes $root[G]$.
- $G.c =$ max-cardinality independent set.

F2: The following multiplication table suffices to describe all possible types of outcomes from the composition $G_1 \oplus G_2$. In the table, rather than displaying $G.a$ and $G.b$ we simply specify a and b, respectively. The row by column product assumes the convention where subgraph (i.e., subtree) G_1 is on the left and subtree G_2 is on the right.

$\oplus$	**a**	**b**
a		a
b	b	b

REMARK

R2: The values for $G.a$, $G.b$, and $G.c$ are known trivially for the base graph – a single-vertex tree (which is its own root). For composed graphs, the values may be computed via $O(1)$ additions and comparisons across the outcomes in the table. There is only one product producing a possible member of $G.a$ while $G.b$ may be produced from a pair of possible products; the maximum of these yields the desired $G.b$. The final step in Algorithm 10.4.1 computes $G.c$, which, at the root of the decomposition tree, is the solution.

Algorithm 10.4.1: Maximum-Cardinality Independent Set in a Tree

Input: a tree $G = (V, E)$.
Output: $G.c$ (the size and an instance of an independent set in G).

```
If $|V| = 1$
    $G.a \leftarrow 1$
    $G.b \leftarrow 0$
Else
    If $G = G_1 \oplus G_2$
        $G.a \leftarrow G_1.a + G_2.b$
        $G.b \leftarrow \max\{G_1.b + G_2.a, G_1.b + G_2.b\}$
$G.c \leftarrow \max\{G.a, G.b\}$
```

COMPUTATIONAL NOTE: The decomposition tree for trees is easy to determine and accordingly, can be assumed to be part of the instance.

Maximum-Weight Independent Set in a Tree

Here we assume that each vertex in the graph is assigned a weight. Switching to a weighted version of the independent set problem is straightforward.

NOTATION: The notation used in the Algorithm 10.4.2 is as follows:

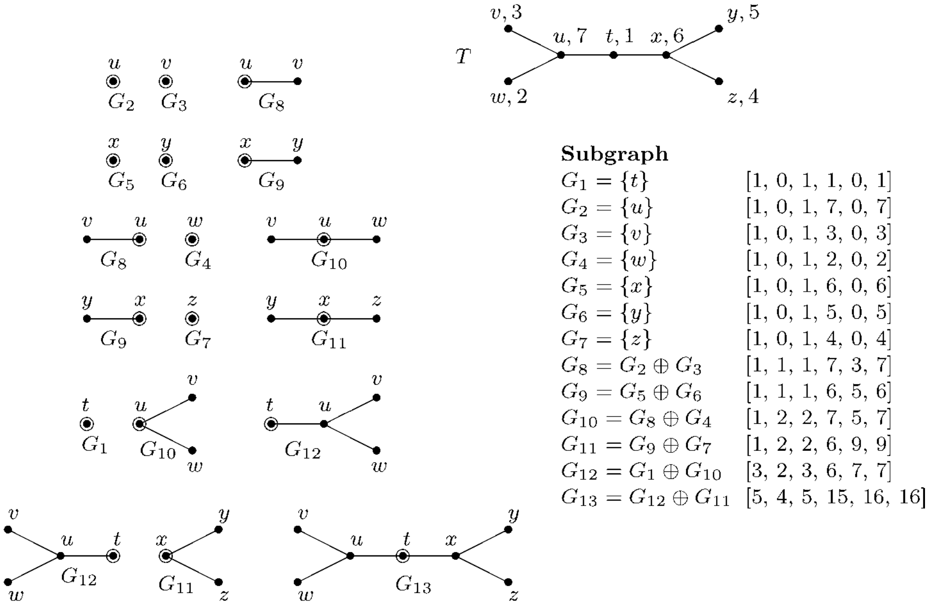
- $G.d$ = max-weight independent set containing $root[G]$.
- $G.e$ = max-weight independent set without $root[G]$.
- $G.f$ = max-weight independent set.

```
Algorithm 10.4.2: Maximum-Weight Independent Set in a Tree

Input: a tree $G = (V, E)$.
Output: $G.f$ (weight and instance of a maximum-weight independent set in $G$).
    If $|V| = 1$
        $G.d \leftarrow$ weight($root[G]$)
        $G.e \leftarrow 0$
    Else
        If $G = G_1 \oplus G_2$
            $G.d \leftarrow G_1.d + G_2.e$
            $G.e \leftarrow$ max $\{G_1.e + G_2.d, G_1.e + G_2.e\}$
    $G.f \leftarrow$ max $\{G.d, G.e\}$
```

EXAMPLE

E1: Consider the tree shown by T in Figure 10.4.1. Vertices are labeled $t, u, \ldots, z$ and next to each label is a vertex weight. Algorithms 10.4.1 and 10.4.2 are applied and the computations are summarized by the listing on the right. The 6-tuples aligned with each composed subgraph, G_k, correspond to values $[G.a, G.b, G.c, G.d, G.e, G.f]$. The maximum cardinality and maximum weight of any independent set is $G.c = 5$ and $G.f = 16$, respectively; these are read from the computation for G_{13}. Standard backtracking can be applied to determine that the explicit solutions are sets $\{t, v, w, y, z\}$ and $\{u, y, z\}$, respectively.



Subgraph	
$G_1 = \{t\}$	[1, 0, 1, 1, 0, 1]
$G_2 = \{u\}$	[1, 0, 1, 7, 0, 7]
$G_3 = \{v\}$	[1, 0, 1, 3, 0, 3]
$G_4 = \{w\}$	[1, 0, 1, 2, 0, 2]
$G_5 = \{x\}$	[1, 0, 1, 6, 0, 6]
$G_6 = \{y\}$	[1, 0, 1, 5, 0, 5]
$G_7 = \{z\}$	[1, 0, 1, 4, 0, 4]
$G_8 = G_2 \oplus G_3$	[1, 1, 1, 7, 3, 7]
$G_9 = G_5 \oplus G_6$	[1, 1, 1, 6, 5, 6]
$G_{10} = G_8 \oplus G_4$	[1, 2, 2, 7, 5, 7]
$G_{11} = G_9 \oplus G_7$	[1, 2, 2, 6, 9, 9]
$G_{12} = G_1 \oplus G_{10}$	[3, 2, 3, 6, 7, 7]
$G_{13} = G_{12} \oplus G_{11}$	[5, 4, 5, 15, 16, 16]

Figure 10.4.1 Max-cardinality and max-weight independent sets in a tree.

REMARKS

R3: Many other problems such as variations of the vertex cover, dominating set, matching, or longest path problems could have been selected to represent the basic computation on trees (cf. [BoPaTo92], [BoPaTo02]). Note that some problems such as minimum bandwidth [GaGrJoKn78] are NP-complete on trees.

R4: A number of problems are trivial when restricted to trees. For example, any tree with at least 2 vertices has maximum clique size of 2 and chromatic number 2, otherwise these are both 1. Also, no tree can contain a hamiltonian cycle, and a tree has a hamiltonian path if and only if it is a path.

R5: The definition and number of equivalence classes that are required to solve a given problem depend on both the graph class and the problem to be solved. The effect of graph class is demonstrated throughout in subsequent subsections of this section. To provide a sense of dependence on the problem, let the distance in connected graph $G = (V, E)$ between $U \subset V$ and $W \subset V$ be the shortest (edge) length of a path from any $u \in U$ to any $w \in W$, and let a *κ-independent set* be a subset of V containing no two distinct vertices at distance κ or less. Finding a maximum-cardinality κ-independent set requires $\kappa + 2$ equivalence classes $G.c$ and $G.i : \ i = 0 \ldots \kappa$, where: $G.c$ is the maximum-cardinality κ-independent sets; $G.\kappa$ denotes maximum-cardinality κ-independent sets at distance at least κ to the root; $G.i : i = 0 \ldots \kappa - 1$ denotes maximum-cardinality κ-independent sets at distance i to the root; multiplication table entry $G_1.i, G_2.j = \min$ $\{i, j + 1\}$ if $i + j \geq \kappa$ and null otherwise. Note that the resulting algorithm may be superlinear if κ is not $O(1)$.

10.4.2 Algorithms on Series-Parallel Graphs

Every series-parallel graph can be composed from single edges using only the three composition rules given in Definition 5 below. For illustrations of these rules, see §2.4.1.

DEFINITION

D5: A ***series-parallel graph*** with distinguished ***terminals*** l and r is denoted (G, l, r) and is defined recursively as follows:

- (base graph) The graph consisting of a single edge (v_1, v_2) is a series-parallel graph (G, l, r) with $l = v_1$ and $r = v_2$.
- A ***series operation*** $(G_1, l_1, r_1) \odot_s (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2. The terminals of the new graph are l_1 and r_2.
- A ***parallel operation*** $(G_1, l_1, r_1) \odot_p (G_2, l_2, r_2)$ forms a series-parallel graph by identifying l_1 with l_2 and r_1 with r_2. The terminals of the new graph are l_1 and r_1.
- A ***jackknife operation*** $(G_1, l_1, r_1) \odot_j (G_2, l_2, r_2)$ forms a series-parallel graph by identifying r_1 with l_2; the new terminals are l_1 and r_1.

REMARK

R6: Series-parallel graphs are recognizable, and their decomposition trees can be constructed in linear time (see §2.4.3). This leads to fast, often linear-time, dynamic programming algorithms for many problems when instances are confined to series-parallel graphs.

Maximum-Cardinality Independent Set in a Series-Parallel Graph

In the case of trees, only a single point of composition involving constituent subtrees, the root vertex, was relevant; a series-parallel graph G has two such points, its terminal vertices. These are denoted $left[G]$ and $right[G]$ in the following description.

NOTATION: Algorithm 10.4.3 uses the following notation:

- $G.a$ = max cardinality independent set containing both $left[G]$ and $right[G]$.
- $G.b$ = max cardinality independent set with $left[G]$ but not $right[G]$.
- $G.c$ = max cardinality independent set with $right[G]$ but not $left[G]$.
- $G.d$ = max cardinality independent set with neither $left[G]$ nor $right[G]$.
- $G.e$ = max cardinality independent set.

Multiplication Tables for Series, Parallel, and Jacknife Operations

$\odot_s$	**a**	**b**	**c**	**d**
a	a	b		
b			a	b
c	c	d		
d			c	d

$\odot_p$	**a**	**b**	**c**	**d**
a	a			
b		b		
c			c	
d				d

$\odot_j$	**a**	**b**	**c**	**d**
a	a	a		
b			b	b
c	c	c		
d			d	d

Algorithm 10.4.3: Max-Card. Independent Set in a Series-Parallel Graph

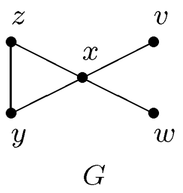
Input: a series-parallel graph $G = (V, E)$.
Output: $G.e$ (size and instance of a maximum-cardinality independent set in G).

```
If $|E| = 1$
    $[G.a, G.b, G.c, G.d] \leftarrow [-\infty, 1, 1, 0]$
Else
    If $G = G_1 \odot_s G_2$
        $G.a \leftarrow \max\{G_1.a + G_2.a - 1, G_1.b + G_2.c\}$
        $G.b \leftarrow \max\{G_1.a + G_2.b - 1, G_1.b + G_2.d\}$
        $G.c \leftarrow \max\{G_1.c + G_2.a - 1, G_1.d + G_2.c\}$
        $G.d \leftarrow \max\{G_1.c + G_2.b - 1, G_1.d + G_2.d\}$
    Else
        If $G = G_1 \odot_p G_2$
            $G.a \leftarrow G_1.a + G_2.a - 2$
            $G.b \leftarrow G_1.b + G_2.b - 1$
            $G.c \leftarrow G_1.c + G_2.c - 1$
            $G.d \leftarrow G_1.d + G_2.d$
        Else
            If $G = G_1 \odot_j G_2$
                $G.a \leftarrow G_1.a + \max\{G_2.a, G_2.b\} - 1$
                $G.b \leftarrow G_1.b + \max\{G_2.c, G_2.d\}$
                $G.c \leftarrow G_1.c + \max\{G_2.a, G_2.b\} - 1$
                $G.d \leftarrow G_1.d + \max\{G_2.c, G_2.d\}$
$G.e \leftarrow \max\{G.a, G.b, G.c, G.d\}$
```

COMPUTATIONAL NOTE: Subtraction of values 2 and 1 in the respective computational expressions above, avoids multiple counting when terminal vertices are fused.

EXAMPLE

E2: Algorithm 10.4.3 is demonstrated on the series-parallel graph G given to the left in Figure 10.4.2. Vertices are labeled in G as shown and to the right, the explicit computation is summarized. The 4-tuples exhibit values for $G.a, G.b, G.c$, and $G.d$. From the last computation, it follows that either $G.b$ or $G.d$ produces an optimum. In the first case, the set is $\{z, v, w\}$, while in the second, we have $\{y, v, w\}$.



Subgraph	
$G_1 = (z, y)$	$[-\infty, 1, 1, 0]$
$G_2 = (y, x)$	$[-\infty, 1, 1, 0]$
$G_3 = (z, x)$	$[-\infty, 1, 1, 0]$
$G_4 = (x, v)$	$[-\infty, 1, 1, 0]$
$G_5 = (x, w)$	$[-\infty, 1, 1, 0]$
$G_6 = G_1 \odot_s G_2$	$[2, 1, 1, 1]$
$G_7 = G_6 \odot_p G_3$	$[-\infty, 1, 1, 1]$
$G_8 = G_7 \odot_j G_4$	$[-\infty, 2, 1, 2]$
$G_9 = G_8 \odot_j G_5$	$[-\infty, 3, 1, 3]$

Figure 10.4.2 **Max-cardinality of an independent set in a series-parallel graph.**

FACTS

F3: Other problems solvable in linear time on series-parallel graphs include variations of clique, dominating set, matching, hamiltonian path, and hamiltonian cycle. Indeed, a series-parallel graph can have at most one hamiltonian cycle (cf. [Sy83]).

F4: A series-parallel graph has chromatic number 3 if it is not bipartite; otherwise it has chromatic number 2 (because it has at least one edge).

REMARKS

R7: Following Fact 3, solving the *traveling salesman problem* (cf. §4.6) on series-parallel graphs reduces to deciding hamiltonicity.

R8: Following Fact 4, the chromatic number for a series-parallel graph can be determined in linear time by using depth-first search to simply test for the existence of an odd cycle.

R9: In some problem settings, the jackknife operation can simply be neglected. For example, a hamiltonian graph must be *2-connected* (i.e., no *cut-vertices*) but the jackknife operation destroys this property, hence, the jackknife operation is not relevant in this case. On the other hand, if the aim is deciding the existence of a hamiltonian path, the jackknife operation is relevant.

R10: Solutions to numerous other problems on series-parallel graphs follow the machinery demonstrated by Algorithm 10.4.3. In addition to the references already cited in Remark 2, a good basic source dealing with problems such as *vertex cover*, *maximum eulerian subgraph*, *Steiner subgraph*, *edge-covering*, etc. is [Ri85].

10.4.3 Algorithms on Treewidth-k Graphs

DEFINITIONS

D6: A ***tree-decomposition*** of a graph $G = (V, E)$ is given by a pair $(\{X_i | i \in I\}, T)$ where $\{X_i | i \in I\}$ is a family of subsets of V and T is a tree with vertex set I such that

- $\bigcup_{i \in I} X_i = V$
- for all edges $(x, y) \in E$ there is an element $i \in I$ with $x, y \in X_i$
- for all triples $i, j, k \in I$ if j is on the path from i to k in T, then we have that $X_i \bigcap X_k \subseteq X_j$.

D7: The ***width*** of a given tree-decomposition is measured as $\max_{i \in I}\{|X_i| - 1\}$.

D8: The ***treewidth*** of a graph G is the minimum width taken over all tree-decompositions of G.

D9: A graph G is a ***treewidth-k graph*** if it has treewidth no greater than k.

FACT

F5: Every treewidth-k graph has a tree-decomposition T such that T is a rooted binary tree [Sc89]. We write $(G, X) = (G_1, X_1) \otimes (G_2, X_2)$ where $X \subseteq V$ is the set of vertices of G associated with $root[T]$, and graphs G_1 and G_2 have tree-decompositions given by the left and right subtrees of T. This is enough to produce linear-time dynamic programming algorithms for many problems on treewidth-k graphs, because each $|X| \leq k + 1$.

NOTATION: For a graph G, let binary tree T be a tree-decomposition of G and let $X \subseteq V$ be the set of vertices of G associated with $root[T]$. Also,

- $G[S]$ = max-cardinality independent set that contains $S \subseteq X$ but not $X - S$.
- $G.max$ = max-cardinality independent set.

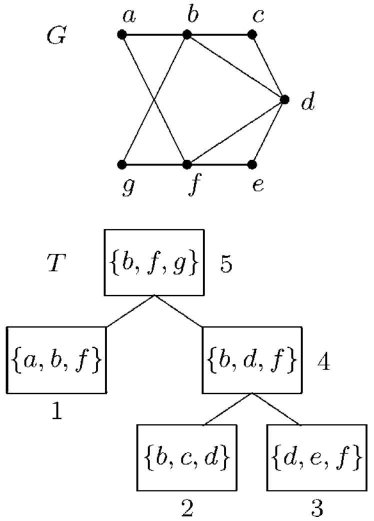
Algorithm 10.4.4: Max-Card. Indep. Set in a Treewidth-k Graph

Input: a treewidth-k graph $G = (V, E)$.
Output: $G.max$ (size and instance of a max-cardinality independent set in G).

```
If $X = V$
    For all $S \subseteq X$
        If $S$ contains two adjacent vertices
            $G[S] \leftarrow -\infty$
        Else
            $G[S] \leftarrow |S|$
Else
    If $(G, X) = (G_1, X_1) \otimes (G_2, X_2)$
        For all $S \subseteq X$
            If $S$ contains two adjacent vertices
                $G[S] \leftarrow -\infty$
            Else
                $G[S] \leftarrow \max\{G_1[S_1] + G_2[S_2] - |S_1 \cap S| - |S_2 \cap S| + |S| :$
                        $S_1 \subseteq X_1, S_2 \subseteq X_2, S_1 \cap X = S \cap X_1, S_2 \cap X = S \cap X_2\}$
$G.max \leftarrow \max\{G[S] : S \subseteq X\}$
```

EXAMPLE

E3: Algorithm 10.4.4 is demonstrated on the treewidth-2 graph G shown in Figure 10.4.3. T is a binary rooted tree-decomposition of G, and each 8-tuple exhibits values for $G[S]$ for each $S \subseteq X$. However, only values larger than $-\infty$ within each 8-tuple are shown as the computation progresses. The maximum independent set has size 4, and the explicit solution is $\{a, c, e, g\}$.



Subgraph

$(G_1, X_1 = \{a, b, f\})$
$[\emptyset \rightarrow 0, \{a\} \rightarrow 1, \{b\} \rightarrow 1, \{f\} \rightarrow 1, \{b, f\} \rightarrow 2]$

$(G_2, X_2 = \{b, c, d\})$
$[\emptyset \rightarrow 0, \{b\} \rightarrow 1, \{c\} \rightarrow 1, \{d\} \rightarrow 1]$

$(G_3, X_3 = \{d, e, f\})$
$[\emptyset \rightarrow 0, \{d\} \rightarrow 1, \{e\} \rightarrow 1, \{f\} \rightarrow 1]$

$(G_4, X_4 = \{b, d, f\}) = G_2 \otimes G_3$
$[\emptyset \rightarrow 2, \{b\} \rightarrow 2, \{d\} \rightarrow 1, \{f\} \rightarrow 2, \{b, f\} \rightarrow 2]$

$(G_5, X_5 = \{b, f, g\}) = G_1 \otimes G_4$
$[\emptyset \rightarrow 3, \{b\} \rightarrow 2, \{f\} \rightarrow 2, \{g\} \rightarrow 4, \{b, f\} \rightarrow 2]$

Figure 10.4.3 Maximum-cardinality independent set in a treewidth-2 graph.

REMARK

R11: Many other problems including variations of independent set, dominating set, clique, m-vertex coloring (for arbitrary, fixed m), matching, and hamiltonian cycle/path problems can be solved in linear time for treewidth-k graphs (cf. [BoPaTo92]).

Monadic Second-Order Logic Expressions for a Graph

DEFINITION

NOTATION: Let variables v_i denote a vertex with domain V, e_i denote an edge with domain E, V_i denote a vertex set with domain 2^V (subsets of V), and E_i denote an edge set with domain 2^E.

D10: ***Monadic second-order logic (MSOL)*** for a graph $G = (V, E)$ is a *predicate calculus language* in which predicates are constructed recursively as follows.

- MSOL contains *primitive* predicates such as $v_i = v_j$, $v_i \in V_j$, $e_i \in E_j$ and Incident(v_i, e_j).
- If P and Q are MSOL predicates then each of $(\neg P)$, $(P \wedge Q)$, and $(P \vee Q)$ is also a MSOL predicate.
- If P is a MSOL predicate and x is any variable, then $(\exists x)(P)$ and $(\forall x)(P)$ are also MSOL predicates.

EXAMPLE

E4: Some simple MSOL predicates are listed below.

$\mathrm{P} \rightarrow \mathrm{Q} \Leftrightarrow \neg \mathrm{P} \vee \mathrm{Q}$

$\mathrm{P} \leftrightarrow \mathrm{Q} \Leftrightarrow (\mathrm{P} \rightarrow \mathrm{Q}) \wedge (\mathrm{Q} \rightarrow \mathrm{P})$

$\mathrm{e}_i = \mathrm{e}_j \Leftrightarrow (\forall \mathrm{v}_1)\ (\mathrm{Incident}(\mathrm{v}_1, \mathrm{e}_i) \leftrightarrow \mathrm{Incident}(\mathrm{v}_1, \mathrm{e}_j))$

$\mathrm{Adjacent}(\mathrm{v}_i, \mathrm{v}_j) \Leftrightarrow \neg(\mathrm{v}_i = \mathrm{v}_j) \wedge (\exists \mathrm{e}_1)\ (\mathrm{Incident}(\mathrm{v}_i, \mathrm{e}_1) \wedge \mathrm{Incident}(\mathrm{v}_j, \mathrm{e}_1))$

MSOL-Expressible Graph Problems

Many important graph problems can be expressed in MSOL (cf. [ArLaSe91], [BoPaTo91], [BoPaTo92]). Here is a sampling of several such problems.

$$\mathrm{IndependentSet}(\mathrm{V}_1) \Leftrightarrow (\forall \mathrm{v}_2)\ (\forall \mathrm{v}_3)\ ((\mathrm{v}_2 \in \mathrm{V}_1 \wedge \mathrm{v}_3 \in \mathrm{V}_1) \rightarrow \neg\ \mathrm{Adjacent}(\mathrm{v}_2, \mathrm{v}_3))$$

$$\mathrm{Clique}(\mathrm{V}_1) \Leftrightarrow (\forall \mathrm{v}_2)\ (\forall \mathrm{v}_3)\ ((\mathrm{v}_2 \in \mathrm{V}_1 \wedge \mathrm{v}_3 \in \mathrm{V}_1) \rightarrow \mathrm{Adjacent}(\mathrm{v}_2, \mathrm{v}_3))$$

$$\mathrm{DominatingSet}(\mathrm{V}_1) \Leftrightarrow (\forall \mathrm{v}_2)\ (\mathrm{v}_2 \in \mathrm{V}_1 \vee (\exists \mathrm{v}_3)\ (\mathrm{v}_3 \in \mathrm{V}_1 \wedge \mathrm{Adjacent}(\mathrm{v}_2, \mathrm{v}_3))$$

$$\begin{aligned}\mathrm{VertexColorable}_m(\mathrm{V}_1, \ldots, \mathrm{V}_m) \Leftrightarrow\ & (\forall \mathrm{v}_0)\ (\mathrm{v}_0 \in \mathrm{V}_1 \vee \ldots \vee \mathrm{v}_0 \in \mathrm{V}_m) \\ & \wedge \mathrm{IndependentSet}(\mathrm{V}_1) \wedge \ldots \wedge \mathrm{IndependentSet}(\mathrm{V}_m)\end{aligned}$$

$$\begin{aligned}\mathrm{Matching}(\mathrm{E}_1) \Leftrightarrow\ & (\forall \mathrm{e}_2)\ (\forall \mathrm{e}_3)\ ((\mathrm{e}_2 \in \mathrm{E}_1 \wedge \mathrm{e}_3 \in \mathrm{E}_1 \wedge \neg\ (\mathrm{e}_2 = \mathrm{e}_3)) \rightarrow \\ & \neg(\exists \mathrm{v}_4)\ (\mathrm{Incident}(\mathrm{v}_4, \mathrm{e}_2) \wedge \mathrm{Incident}(\mathrm{v}_4, \mathrm{e}_3)))\end{aligned}$$

$$\begin{aligned}\mathrm{Connected}(\mathrm{E}_1) \Leftrightarrow\ & (\forall \mathrm{V}_2)\ (\forall \mathrm{V}_3)\ (\neg\ (\exists \mathrm{v}_4)\ (\mathrm{v}_4 \in \mathrm{V}_2) \vee \neg\ (\exists \mathrm{v}_5)\ (\mathrm{v}_5 \in \mathrm{V}_3) \vee \\ & (\exists \mathrm{v}_6)\ (\neg\ (\mathrm{v}_6 \in \mathrm{V}_2) \wedge \neg\ (\mathrm{v}_6 \in \mathrm{V}_3)) \vee \\ & (\exists \mathrm{e}_7)\ (\exists \mathrm{v}_8)\ (\exists \mathrm{v}_9)\ (\mathrm{e}_7 \in \mathrm{E}_1 \wedge \mathrm{v}_8 \in \mathrm{V}_2 \wedge \mathrm{v}_9 \in \mathrm{V}_3 \wedge \\ & \mathrm{Incident}(\mathrm{v}_8, \mathrm{e}_7) \wedge \mathrm{Incident}(\mathrm{v}_9, \mathrm{e}_7)))\end{aligned}$$

$$\begin{aligned}\mathrm{HamCycle}(\mathrm{E}_1) \Leftrightarrow\ & \mathrm{Connected}(\mathrm{E}_1) \wedge (\forall \mathrm{v}_2)\ (\exists \mathrm{e}_3)\ (\exists \mathrm{e}_4)\ (\mathrm{e}_3 \in \mathrm{E}_1 \wedge \mathrm{e}_4 \in \mathrm{E}_1 \wedge \\ & \neg(\mathrm{e}_3 = \mathrm{e}_4) \wedge \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_3) \wedge \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_4) \wedge \\ & (\forall \mathrm{e}_5)\ ((\mathrm{e}_5 \in \mathrm{E}_1 \wedge \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_5)) \rightarrow (\mathrm{e}_5 = \mathrm{e}_3 \vee \mathrm{e}_5 = \mathrm{e}_4)))\end{aligned}$$

$$\begin{aligned}\mathrm{HamPath}(\mathrm{E}_1) \Leftrightarrow\ & \mathrm{Connected}(\mathrm{E}_1) \wedge (\forall \mathrm{v}_2)\ (\exists \mathrm{e}_3)\ (\exists \mathrm{e}_4)\ (\mathrm{e}_3 \in \mathrm{E}_1 \wedge \mathrm{e}_4 \in \mathrm{E}_1 \wedge \\ & \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_3) \wedge \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_4) \wedge \\ & (\forall \mathrm{e}_5)\ ((\mathrm{e}_5 \in \mathrm{E}_1 \wedge \mathrm{Incident}(\mathrm{v}_2, \mathrm{e}_5)) \rightarrow (\mathrm{e}_5 = \mathrm{e}_3 \vee \mathrm{e}_5 = \mathrm{e}_4)) \wedge \\ & (\exists \mathrm{v}_6)\ (\exists \mathrm{e}_7)\ (\forall \mathrm{e}_8)\ ((\mathrm{e}_8 \in \mathrm{E}_1 \wedge \mathrm{Incident}(\mathrm{v}_6, \mathrm{e}_8)) \rightarrow \mathrm{e}_8 = \mathrm{e}_7))\end{aligned}$$

FACTS

F6: Every MSOL-expressible problem can be solved in linear time for treewidth-k graphs [ArLaSe91], [BoPaTo92], [Co90]. Moreover, this is the case for many variations of each MSOL problem, including existence, minimum or maximum cardinality, minimum or maximum total weight, minimum-maximal or maximum-minimal sets, bottleneck weight, and counting.

F7: Once a problem is expressed in MSOL, a linear-time dynamic programming algorithm can be created mechanically [BoPaTo92].

F8: The *chromatic number* problem (§5.1) for treewidth-k graphs is solvable in linear time, because every treewidth-k graph possesses a vertex coloring with at most $k + 1$ colors.

F9: For some problems, a MSOL expression cannot be written and a linear-time algorithm cannot be found. In these cases it may still be possible to develop a linear-time algorithm via an extension to MSOL [BoPaTo92], or to develop a polynomial-time algorithm. Polynomial time is achieved by constructing a polynomial-size data structure that corresponds to each node in the tree decomposition (see Remark 1; also see [Bo95] and [BoPaTo02]).

REMARK

R12: The literature contains hundreds of linear-time algorithms for problems on trees, series-parallel graphs, treewidth-k graphs, and related classes. Many of these algorithms are predicted by the results cited in Fact 6. For example, all of the linear-time algorithms given in [BoPaTo02] for problems on trees, series-parallel graphs, and treewidth-k graphs are predicted by these results.

Two Open Problems

The hidden constant in the running time of a mechanically-created algorithm can grow superexponentially with the number of quantifiers $\exists$ and $\forall$ present in the formula, so Fact 7 is impractical for complex formulas. *Ad hoc* methods often suffice to design an equivalent linear-time dynamic programming algorithm with a small hidden constant, but a computationally practical algorithm remains elusive [Ka01].

Open Problem 1. Determine an optimally efficient algorithm to create a linear time dynamic programming algorithm given an MSOL expression.

Open Problem 2. Determine a procedure that given an MSOL expression produces a linear-time dynamic programming algorithm with minimum hidden constant.

REMARKS

R13: Chromatic index can be solved on treewidth-k graphs in polynomial time, but it is not known to be solvable in linear time. Polynomial time is achieved by constructing a polynomial-size data structure that corresponds to each node in the tree decomposition (cf. [BoPaTo02]).

R14: Algorithms on treewidth-k graphs can be adapted to solve the same problems on related classes such as Halin graphs, partial k-trees, bandwidth-k graphs, pathwidth-k graphs, branchwidth-k graphs, and k-terminal graphs (cf. [Wi87], [WiHe88]).

10.4.4 Algorithms on Cographs

One can produce linear-time dynamic programming algorithms for problems related to cographs. Here, in addition to the maximum-cardinality independent-set problem, we present algorithms for several other problems.

DEFINITIONS

D11: A ***clique*** in a graph is a maximal set of pairwise adjacent vertices. (§5.3)

D12: The ***chromatic number*** of a graph G is the minimum number of colors that can be used to color the vertices of G so that no two adjacent vertices get the same color. (§5.1)

D13: A ***dominating set*** in a graph G is a subset S of vertices such that every vertex in G is either in S or adjacent to a vertex in S. (§9.2)

D14: A ***matching*** in a graph is a set of edges no two of which have an endpoint in common.

D15: A ***cograph*** is defined recursively as follows:

- (base graph) A graph with a single vertex is a cograph.
- If G_1 and G_2 are cographs, then the disjoint union $G_1 \cup G_2$ is a cograph.
- If G_1 and G_2 are cographs, then the cross-product $G_1 \times G_2$ is a cograph, which is formed by taking the union of G_1 and G_2 and adding all edges (v_1, v_2), where v_1 is in G_1 and v_2 is in G_2.

Algorithm 10.4.5: Maximum-Cardinality Independent Set in a Cograph

Input: a cograph $G = (V, E)$.
Output: $G.i$ (the size and an instance of an independent set in G).

```
If $|V| = 1$
    $G.i \leftarrow 1$
Else
    If $G = G_1 \cup G_2$
        $G.i \leftarrow G_1.i + G_2.i$
    Else
        If $G = G_1 \times G_2$
            $G.i \leftarrow \max\{G_1.i, G_2.i\}$
```

Three More Algorithms for Cographs

Algorithm 10.4.6 works for both the *maximum-clique* and *chromatic number* problems in a cograph.

Algorithm 10.4.6: Maximum-Clique (Chromatic Number) in a Cograph

Input: a cograph $G = (V, E)$.
Output: $G.c$ (maximum clique in [chromatic number of] G).

```
If $|V| = 1$
    $G.c \leftarrow 1$
Else
    If $G = G_1 \cup G_2$
        $G.c \leftarrow \max\{G_1.c, G_2.c\}$
    Else
        If $G = G_1 \times G_2$
            $G.c \leftarrow G_1.c + G_2.c$
```

Algorithm 10.4.7: Minimum-Cardinality Dominating Set in a Cograph

Input: a cograph $G = (V, E)$.
Output: $G.d$ (the size and an instance of a minimum dominating set in G).

```
If $|V| = 1$
    $G.d \leftarrow 1$
Else
    If $G = G_1 \cup G_2$
        $G.d \leftarrow G_1.d + G_2.d$
    Else
        If $G = G_1 \times G_2$
            $G.d \leftarrow \max\{G_1.d, G_2.d, 2\}$
```

Algorithm 10.4.8: Maximum-Cardinality Matching in a Cograph

Input: a cograph $G = (V, E)$.
Output: $G.m$ (the size and an instance of a maximum matching in G).

```
If $|V| = 1$
    $G.m \leftarrow 0$
Else
    If $G = G_1 \cup G_2$
        $G.m \leftarrow G_1.m + G_2.m$
    Else
        If $G = G_1 \times G_2$
            $G.m \leftarrow \min\{G_1.m + |V_2|,\ G_2.m + |V_1|,\ \lfloor(|V_1| + |V_2|)/2\rfloor\}$
```

REMARKS

R15: The right-hand side of the final assignment in Algorithm 10.4.8 above is obtained by simplifying the more straightforward but less efficient formula below, wherein k denotes the number of matching edges with one endpoint in each of the subgraphs G_1 and G_2.

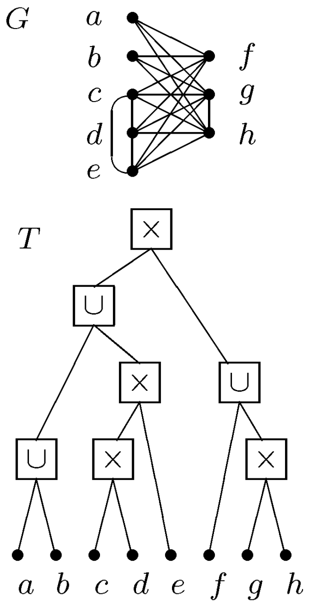
$$\min_{0 \le k \le \min\{|V_1|, |V_2|\}} \left\{ k + \min\{G_1.m, \lfloor(|V_1| - k)/2\rfloor\} + \min\{G_2.m, \lfloor(|V_2| - k)/2\rfloor\} \right\}$$

R16: The hamiltonian cycle and hamiltonian path problems can be solved in linear time on cographs. Weighted versions of the independent set, clique, and dominating set problems are also solvable in linear time on cographs by extending Algorithms 10.4.5 through 10.4.7. However, weighted versions of the matching and hamiltonian problems do not appear to be solvable in linear time on cographs (although they are solvable in polynomial time). Intuitively, the reason is that the cross product operation adds too many edges, where each edge potentially has a different weight.

EXAMPLE

E5: Algorithms 10.4.5 through 10.4.8 are demonstrated on the cograph G shown in Figure 10.4.4. T denotes the tree decomposition of G, and each 4-tuple exhibits values for $G.i, G.c, G.d$, and $G.m$. The maximum independent set has size 3, for example

$\{a,b,c\}$. The maximum clique has size 5, given by $\{c,d,e,g,h\}$. The minimum dominating set has size 2, for example $\{a,f\}$. The maximum matching has size 4, for example $\{(a,f),(b,g),(c,h),(d,e)\}$.



Subgraph	
$G_1 = \{a\}$	$[1,1,1,0]$
$G_2 = \{b\}$	$[1,1,1,0]$
$G_3 = \{c\}$	$[1,1,1,0]$
$G_4 = \{d\}$	$[1,1,1,0]$
$G_5 = \{e\}$	$[1,1,1,0]$
$G_6 = \{f\}$	$[1,1,1,0]$
$G_7 = \{g\}$	$[1,1,1,0]$
$G_8 = \{h\}$	$[1,1,1,0]$
$G_9 = G_1 \cup G_2$	$[2,1,2,0]$
$G_{10} = G_3 \times G_4$	$[1,2,1,1]$
$G_{11} = G_7 \times G_8$	$[1,2,1,1]$
$G_{12} = G_{10} \times G_5$	$[1,3,1,1]$
$G_{13} = G_6 \cup G_{11}$	$[2,2,2,1]$
$G_{14} = G_9 \cup G_{12}$	$[3,3,3,1]$
$G_{15} = G_{13} \times G_{14}$	$[3,5,2,4]$

Figure 10.4.4 **Illustrating Algorithms 10.4.5 through 10.4.8.**

10.4.5 Algorithms on Cliquewidth-k Graphs

DEFINITION

D16: Let [k] denote the set of integers $\{1,2,\ldots,k\}$. A ***cliquewidth-k graph*** is defined recursively as follows:

- (base graph) Any graph G with $V(G)=\{v\}$ and $l(v)\in[k]$ is a cliquewidth-k graph.
- If G_1 and G_2 are cliquewidth-k graphs and $i,j\in[k]$, then
  (1) the disjoint union $G_1 \cup G_2$ is a cliquewidth-k graph.
  (2) the graph $(G_1)_{i\times j}$ is a cliquewidth-k graph, where $(G_1)_{i\times j}$ is formed from G_1 by adding all edges (v_1,v_2) such that $l(v_1)=i$ and $l(v_2)=j$.
  (3) the graph $(G_1)_{i\to j}$ is a cliquewidth-k graph, where $(G_1)_{i\to j}$ is formed from G_1 by switching all vertices with label i to label j.

Maximum-Cardinality Independent Set of a Cliquewidth-k Graph

NOTATION: Algorithm 10.4.9 below uses the following notation:

- $G[S]$ = max-cardinality independent set that contains exactly the labels in $S\subseteq[k]$.
- $G.max$ = max-cardinality independent set.

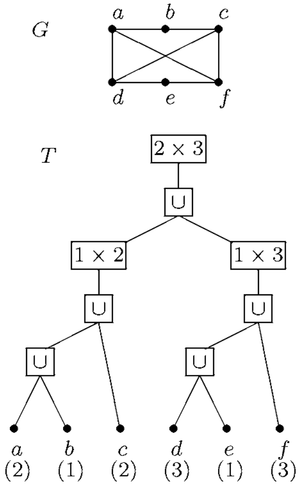
Algorithm 10.4.9: Max-Card. Indep. Set in a Cliquewidth-k Graph

Input: a cliquewidth-k graph $G = (V, E)$.
Output: $G.max$ (size and instance of a max-cardinality independent set in G).

```
    If $V = \{v\}$
        For all $S \subseteq [k]$
            If $S = \{l(v)\}$
                $G[S] \leftarrow 1$
            Else
                $G[S] \leftarrow 0$
    Else
        If $G = G_1 \cup G_2$
            For all $S \subseteq [k]$
                $G[S] \leftarrow G_1[S] + G_2[S]$
        Else
            If $G = (G_1)_{i \times j}$
                For all $S \subseteq [k]$
                    $G[S] \leftarrow \max\{G_1[S - \{i\}],\ G_1[S - \{j\}]\}$
            Else
                If $G = (G_1)_{i \to j}$
                    For all $S \subseteq [k]$
                        If $j \in S$
                            $G[S] \leftarrow G_1[S \cup \{i\}]$
                        Else
                            $G[S] \leftarrow G_1[S - \{i\}]$
    $G.max \leftarrow \max\{G[S] : S \subseteq [k]\}$
```

EXAMPLE

E6: Algorithm 10.4.9 is demonstrated on the cliquewidth-3 graph G shown in Figure 10.4.5 below. T denotes the decomposition tree of G. Each 8-tuple consists of $G[\emptyset]$, $G[\{1\}]$, $G[\{2\}]$, $G[\{3\}]$, $G[\{1,2\}]$, $G[\{1,3\}]$, $G[\{2,3\}]$, and $G[\{1,2,3\}]$. The maximum independent set has size 3, given by either $\{a, c, e\}$ or $\{b, d, f\}$.



Subgraph	
$G_1 = \{a\}$	$[0,0,1,0,1,0,1,1]$
$G_2 = \{b\}$	$[0,1,0,0,1,1,0,1]$
$G_3 = \{c\}$	$[0,0,1,0,1,0,1,1]$
$G_4 = \{d\}$	$[0,0,0,1,0,1,1,1]$
$G_5 = \{e\}$	$[0,1,0,0,1,1,0,1]$
$G_6 = \{f\}$	$[0,0,0,1,0,1,1,1]$
$G_7 = G_1 \cup G_2$	$[0,1,1,0,2,1,1,2]$
$G_8 = G_4 \cup G_5$	$[0,1,0,1,1,2,1,2]$
$G_9 = G_3 \cup G_7$	$[0,1,2,0,3,1,2,3]$
$G_{10} = G_6 \cup G_8$	$[0,1,0,2,1,3,2,3]$
$G_{11} = (G_9)_{1\times 2}$	$[0,1,2,0,2,1,2,2]$
$G_{12} = (G_{10})_{1\times 3}$	$[0,1,0,2,1,2,2,2]$
$G_{13} = G_{11} \cup G_{12}$	$[0,2,2,2,3,3,4,4]$
$G_{14} = (G_{13})_{2\times 3}$	$[0,2,2,2,3,3,2,3]$

Figure 10.4.5 **Maximum-cardinality independent set in a cliquewidth-3 graph.**

A Subset of the MSOL Expressions for a Graph

Many problems including variations of independent set, dominating set, clique, and m-vertex colorability (for any fixed m) can be solved in linear time on cliquewidth-k graphs, provided that a decomposition tree is known. These problems are all expressible using a certain subset of the MSOL expressions.

DEFINITION

D17: The ***MSOL***$'$ set of expressions for a graph $G = (V, E)$ is the subset of MSOL expressions restricted to variables v_i with domain V, e_i with domain E, and V_i with domain 2^V. The MSOL$'$ set contains primitive predicates such as $v_i = v_j$, Incident(v_i, e_j), and $v_i \in V_j$. MSOL$'$ permits the logical operators ($\neg$, $\wedge$, $\vee$) and quantifiers ($\exists$, $\forall$). Thus, MSOL$'$ is the same as MSOL without the edge-set variables E_i and without primitive predicates such as $e_i \in E_j$ that refer to edge-set variables.

FACT

F10: Every MSOL$'$-expressible problem can be solved in linear time on any class of cliquewidth-k graphs [CoMaRo00], provided that either there exists a linear time decomposition algorithm for the class (as for cographs) or a decomposition tree is provided as part of the input. This statement holds for variations of each MSOL$'$ problem that involve existence, optimum cardinality or total weight, counting the number of solutions, etc. Once a problem is expressed in MSOL$'$, a linear-time dynamic programming algorithm can be created mechanically.

REMARKS

R17: Observe that the MSOL expressions given in §10.4.3 for IndependentSet(V_1), Clique(V_1), DominatingSet(V_1), and VertexColorable$_m$(V_1, …, V$_m$), are also MSOL$'$ expressions. However, the MSOL expressions given for Matching(E_1), Connected(E_1), HamCycle(E_1), and HamPath(E_1) are not in MSOL$'$.

R18: Some problems such as variations of matching and hamiltonicity do not appear to be expressible in MSOL$'$, and it is not known whether these problems can be solved in linear time on cliquewidth-k graphs. However, such problems can often be solved in polynomial time, given the decomposition tree. Polynomial time is achieved by constructing a polynomial-size data structure that corresponds to each node in the tree decomposition (cf. [BoPaTo02]).

R19: Algorithms on cliquewidth-k graphs can be adapted to solve the same problems on related classes such as k-NLC graphs [Wa94].

10.4.6 Algorithms on k-HB Graphs

DEFINITION

D18: ***k-HB (homogeneous balanced) graphs*** are graphs for which there is a particular $O(n^{k+2})$-time top-down decomposition algorithm that constructs a pseudo-cliquewidth-$(k + 2^k)$ balanced decomposition. (Also see §2.4.1 and [BoJoRaSp02].)

FACTS

F11: Every k-HB graph can be composed from single vertices using only the operation $G = G_1 \times_{B,h} G_2$. Here G_1 and G_2 denote child subgraphs, each $|V_i| \leq 2/3\ |V|$, $B = (V_B, E_B)$ is a bipartite graph with $V_B = Z_1 \cup Z_2$ and $E_B \subseteq Z_1 \times Z_2$, $|Z_1| \leq k$, $|Z_2| \leq 2^k$, h: $V \rightarrow V_B$ is a mapping with each $h(V_i) \subseteq Z_i$, and $(x, y) \in E$ iff $(h(x), h(y)) \in E_B$ for all $x \in V_1$ and $y \in V_2$.

F12: This k-HB decomposition leads to polynomial-time dynamic programming algorithms for many problems on k-HB graphs, using recursion (top-down) rather than dynamic programming (bottom-up). Each algorithm's running time is polynomial because at each node of the decomposition it evaluates $O(1)$ parameters, each of which produces $O(1)$ recursive calls on smaller subproblems. Also, the decomposition has $O(\log |V|)$ height, hence $|V|^{O(1)}$ nodes. [BoJoRaSp02]

Maximum-Cardinality Independent Set in a k-HB Graph

NOTATION: Algorithm 10.4.10 below uses the following notation:

- $G[S]$ = max-cardinality independent set that contains only vertices in $S \subseteq V$.
- $G.indep$ = max-cardinality independent set.

Algorithm 10.4.10: Max-Cardinality Independent Set in a k-HB Graph

Input: a k-HB graph $G = (V, E)$.
Output: $G.indep$ (size and instance of a maximum-cardinality independent set in G).

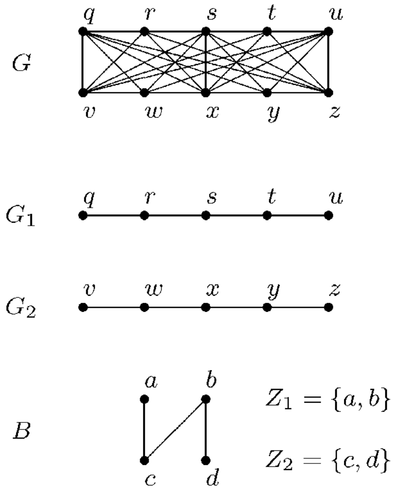
```
If $|V| = 1$
    $G[S] \leftarrow |S|$
Else
    If $G = G_1 \times_{B,h} G_2$
        $G[S] \leftarrow \max\{G_1[T] + G_2[U] : X \subseteq h(V_1),\ Y \subseteq h(V_2),$
                $(X \times Y) \cap E_B = \emptyset,\ T = S \cap h^{-1}(X),\ U = S \cap h^{-1}(Y)\}$
$G.indep = G[V]$
```

EXAMPLE

E7: We demonstrate Algorithm 10.4.10 on the 2-HB graph G shown in Figure 10.4.6 below. Note that $G = G_1 \times_{B,h} G_2$, where G_1, G_2, B, and h are as shown. The top-level computations are summarized on the right. The maximum independent set has size 4, and the explicit solution is $\{r, t, w, y\}$.

REMARK

R20: The maximum-cardinality clique and m-vertex colorability problems can also be solved in polynomial time on k-HB graphs. However, the chromatic number, dominating set, and hamiltonian problems are not known to be solvable in polynomial time on k-HB graphs. Maximum matching is of course solvable in polynomial time on k-HB graphs, but it is not known whether this can be done more efficiently than for arbitrary graphs.



h

$h(r) = h(t) = a$
$h(q) = h(s) = h(u) = b$
$h(v) = h(x) = h(z) = c$
$h(w) = h(y) = d$

Recursive calls

$X = \emptyset \Rightarrow T = \emptyset \Rightarrow G_1[T] = 0$
$X = \{a\} \Rightarrow T = \{r, t\} \Rightarrow G_1[T] = 2$
$X = \{b\} \Rightarrow T = \{q, s, u\} \Rightarrow G_1[T] = 3$
$X = \{a, b\} \Rightarrow T = \{q, r, s, t, u\} \Rightarrow G_1[T] = 3$

$Y = \emptyset \Rightarrow U = \emptyset \Rightarrow G_2[U] = 0$
$Y = \{c\} \Rightarrow U = \{v, x, z\} \Rightarrow G_2[U] = 3$
$Y = \{d\} \Rightarrow U = \{w, y\} \Rightarrow G_2[U] = 2$
$Y = \{c, d\} \Rightarrow U = \{v, w, x, y, z\} \Rightarrow G_2[U] = 3$

Compatible subsets

$X = \emptyset, Y = \emptyset \Rightarrow G_1[T] + G_2[U] = 0 + 0 = 0$
$X = \emptyset, Y = \{c\} \Rightarrow G_1[T] + G_2[U] = 0 + 3 = 3$
$X = \emptyset, Y = \{d\} \Rightarrow G_1[T] + G_2[U] = 0 + 2 = 2$
$X = \emptyset, Y = \{c, d\} \Rightarrow G_1[T] + G_2[U] = 0 + 3 = 3$
$X = \{a\}, Y = \emptyset \Rightarrow G_1[T] + G_2[U] = 2 + 0 = 2$
$X = \{a\}, Y = \{d\} \Rightarrow G_1[T] + G_2[U] = 2 + 2 = 4$
$X = \{b\}, Y = \emptyset \Rightarrow G_1[T] + G_2[U] = 3 + 0 = 3$
$X = \{a, b\}, Y = \emptyset \Rightarrow G_1[T] + G_2[U] = 3 + 0 = 3$

Figure 10.4.6 **Maximum-cardinality independent set in a 2-HB graph.**

A Subset of the MSOL′ Expressions

Most problems that are known to be solvable in polynomial time for k-HB graphs are expressible in a particular predicate language whose expressions form a subset of the MSOL′ expressions.

DEFINITION

D19: MSOL″ for a graph $G = (V, E)$ denotes a subset of MSOL′ restricted to variables v_i with domain V, and variables V_i with domain 2^V. MSOL″ contains primitive predicates such as Adjacent(v_i, v_j) and $v_i \in V_j$. MSOL″ permits the logical operators ($\neg$, $\wedge$, $\vee$) and quantifiers ($\exists$, $\forall$). However, these primitives and connectors cannot be combined in any arbitrary way; rather every MSOL″ expression must possess the following format.

$$\begin{aligned}&(\exists V_1)\ldots(\exists V_m)\ ((\forall v_1)F_0(v_1 \in V_1, \ldots, v_1 \in V_m)\\ &\qquad \wedge(\forall v_2)(\forall v_3)\ (\text{Adjacent}(v_2, v_3) \rightarrow \wedge_{1\le i\le j\le m}F_{ij}(v_2 \in V_i, v_3 \in V_j))\\ &\qquad \wedge(\forall v_4)(\forall v_5)(\neg\ \text{Adjacent}(v_4, v_5) \rightarrow \wedge_{1\le i\le j\le m}F'_{ij}(v_4 \in V_i, v_5 \in V_j)))\end{aligned}$$

REMARK

R21: In Definition 19 above, each F_0, each F_{ij}, and each F'_{ij} is an arbitrary formula that combines the indicated primitive predicates using operators $\neg$, $\wedge$, and $\vee$. If any of these formulas is identically true, it may be omitted.

EXAMPLE

E8: The MSOL expressions for IndependentSet, Clique, and VertexColorable$_m$ can be rewritten as equivalent MSOL$''$ expressions as shown below. However, other MSOL expressions such as DominatingSet do not appear to be expressible in MSOL$''$.

$$\text{IndependentSet} \Leftrightarrow (\exists V_1)(\forall v_2)(\forall v_3)\ (\text{Adjacent}(v_2, v_3) \rightarrow \neg(v_2 \in V_1 \wedge v_3 \in V_1))$$

$$\text{Clique} \Leftrightarrow (\exists V_1)(\forall v_4)(\forall v_5)(\neg\ \text{Adjacent}(v_4, v_5) \rightarrow \neg(v_4 \in V_1 \wedge v_5 \in V_1))$$

$$\text{VertexColorable}_m \Leftrightarrow (\exists V_1)\ldots(\exists V_m)((\forall v_1)(v_1 \in V_1 \vee \ldots \vee v_1 \in V_m)$$
$$\wedge(\forall v_2)(\forall v_3)\ (\text{Adjacent}(v_2, v_3) \rightarrow \wedge_{1 \le i \le m} \neg(v_2 \in V_i \wedge v_3 \in V_i)))$$

FACT

F13: Every MSOL$''$-expressible problem can be solved in polynomial time when the input graph is restricted to any class of k-HB graphs [BoJoRaSp02]. This includes every cliquewidth-k graph, even if its decomposition tree is not provided as part of the input. Once a problem is expressed in MSOL$''$, the polynomial-time recursive algorithm can be created mechanically.

References

[Ar85] S. Arnborg, Efficient algorithms for combinatorial problems on graphs with bounded decomposibility — a survey, *BIT* 25 (1985), 2–23.

[ArCoPrSe93] S. Arnborg, B. Courcelle, A. Proskurowski, D. Seese, An algebraic theory of graph reduction, *Journal of the ACM* 40 (1993), 1134–1164.

[ArLaSe91] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, *Journal of Algorithms* 12 (1991), 308–340.

[ArPr89] S. Arnborg, A. Proskurowski, Linear time algorithms for NP-hard problems restricted to partial k-trees, *Discrete Applied Mathematics* 23 (1989), 11–24.

[BeLaWo87] M. W. Bern, E. L. Lawler, A. L. Wong, Linear time computation of optimal subgraphs of decomposable graphs, *Journal of Algorithms* 8 (1987), 216–235.

[Bo87] H. L. Bodlaender, Dynamic programming on graphs with bounded tree-width, Ph.D. Dissertation, Massachusetts Institute of Technology, 1987; extended abstract in *Proceedings of ICALP* (1988).

[Bo95] R. B. Borie, Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs, *Algorithmica* 14 (1995), 123–137.

[BoJoRaSp02] R. B. Borie, J. L. Johnson, V. Raghavan, J. P. Spinrad, Robust polynomial time algorithms on clique-width k graphs, manuscript, 2002.

[BoPaTo91] R. Borie, R. G. Parker, C. A. Tovey, Algorithms for recognition of regular properties and decomposition of recursive graph families, *Annals of Operations Research* 33 (1991), 127–149.

[BoPaTo92] R. B. Borie, R. G. Parker, C. A. Tovey, Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families, *Algorithmica* 7 (1992), 555–581.

[BoPaTo02] R. B. Borie, R. G. Parker, C. A. Tovey, Solving problems on recursively constructed graphs, Technical Report, Department of Computer Science, University of Alabama, 2002.

[BrLeSp99] A. Brandstadt, V. B. Lee, J. P. Spinrad, *Graph Classes: A Survey*, SIAM monograph, 1999.

[CoLeBu81] D. G. Corneil, H. Lerchs, L. S. Burlingham, Complement reducible graphs, *Discrete Applied Mathematics* 3 (1981), 163–174.

[CoPeSt85] D. G. Corneil, Y. Perl, L. K. Stewart, A linear recognition algorithm for cographs, *SIAM Journal on Computing* 14 (1985), 926–934.

[Co90] B. Courcelle, The monadic second-order logic of graphs I: Recognizable sets of finite graphs, *Information and Computation* 85 (1990), 12–75.

[CoMaRo00] B. Courcelle, J. A. Makowsky, U. Rotics, Linear time solvable optimization problems on graphs of bounded clique width, *Theory of Computing Systems* 33 (2000), 125–150.

[CoMo93] B. Courcelle, M. Mosbah, Monadic second-order evaluations on tree-decomposable graphs, *Theoretical Computer Science* 109 (1993), 49–82.

[CoOl00] B. Courcelle, S. Olariu, Upper bounds to the clique-width of graphs, *Discrete Applied Mathematics* 101 (2000), 77–114.

[De97] B. De Fluiter, Algorithms for graphs of small treewidth, Ph.D. Dissertation, University of Utrecht, 1997.

[Ed65-a] J. Edmonds, Paths, trees, and flowers, *Canadian Journal of Mathematics* 17 (1965), 449–467.

[Ed65-b] J. Edmonds, Maximum matching and polyhedron of 0,1 vertices, *J. Research National Bureau of Standards* 69B (1965), 125–130.

[ElCo88] E. El-Mallah, C. Colbourn, Partial k-tree algorithms, *Congressus Numerantium* 64 (1988), 105–119.

[EsGuWa01] W. Espelage, F. Gurski, E. Wanke, How to solve np-hard graph problems on clique-width bounded graphs in polynomial time, pp. 117–128 in *WG 2001, Lecture Notes in Computer Science* 2204 (2001).

[GaGrJoKn78] M. Garey, R. Graham, D. Johnson, D. Knuth, Complexity results for bandwidth minimization, *SIAM Journal on Applied Mathematics* 34 (1978), 477–495.

[HaHeLaPeWi87] E. Hare, S. Hedetniemi, R. Laskar, K. Peters, T. Wimer, Linear-time computability of combinatorial problems on generalized series-parallel graphs, *Discrete Algorithms and Complexity* 14 (1987), 437–457.

[JaOl95] B. Jamison, S. Olariu, Linear time optimization algorithms for P4-sparse graphs, *Discrete Applied Mathematics* 61 (1995), 155–175.

[Ka01] I. T. Kassios, Translating Borie-Parker-Tovey calculus into mutumorphisms, manuscript, 2001.

[Ri85] M. B. Richey, Combinatorial optimization on series-parallel graphs: algorithms and complexity, Ph.D. Dissertation, Georgia Institute of Technology, 1985.

[Sc87] P. Scheffler, Linear-time algorithms for NP-complete problems restricted to partial k-trees, Technical report R-MATH-03/87, *IMATH*, Berlin, 1987.

[Sc89] P. Scheffler, Die Baumweite von Graphen als ein Maβ für die Kompliziertheit algorithmischer Probleme, Ph.D. Dissertation, Akademie der Wissenschaften der DDR, 1989.

[ScSe86] P. Scheffler, D. Seese, Graphs of Bounded tree-width and linear-time algorithms for NP-complete problems, *Proceedings of the Bilateral Seminar* Samarkand, 1986.

[ScSe88] P. Scheffler, D. Seese, A combinatorial and logical approach to linear-time computability, *EUROCAL* 487; pp. 379–380 *Lecture Notes in Computer Science* 378 (1988).

[Sp97] J. P. Spinrad, *Representations of Graphs*, book manuscript (1997).

[Sy83] M. Syslo, NP-complete Problems on Some Tree-structured graphs: a review, *Proceedings of 9th Workshop on Graph-theoretic Concepts in Computer Science* (1983), 342–353.

[TaNiSa82] K. Takamizawa, T. Nishizeki, N. Saito, Linear-time computability of combinatorial problems on series-parallel graphs, *Journal of the ACM* 29 (1982), 623–641.

[Wa94] E. Wanke, k-NLC graphs and polynomial algorithms, *Discrete Applied Mathematics* 54 251–266 (1994); later revised with new co-author F. Gurski.

[Wi87] T. V. Wimer, Linear algorithms on k-terminal recursive graphs, Ph.D. Dissertation, Clemson University, 1987.

[WiHe88] T. V. Wimer, S. T. Hedetniemi, k-terminal recursive families of graphs, *Congressus Numerantium* 63 (1988), 161–176.

[WiHeLa85] T. V. Wimer, S. T. Hedetniemi, R. Laskar, A methodology for constructing linear graph algorithms, *Congressus Numerantium* 50 (1985), 43–60.

GLOSSARY FOR CHAPTER 10

ancestor – of a vertex y in a rooted tree: a vertex x that lies on the unique path from the root to y.

___, **proper** – of a vertex y in a rooted tree: an ancestor other than vertex y itself.

angular resolution ρ – in a polyline drawing: the smallest angle formed by two edges, or segments of edges, incident on the same vertex or bend.

α**-approximation algorithm** – for minimization problems: a polynomial-time algorithm that is guaranteed to find a solution of size at most α times the minimum.

area of a drawing: the area of the convex hull of the drawing.

articulation point: see *cutpoint.*

aspect ratio – of a drawing: the ratio of the longest to the shortest side of the smallest rectangle with horizontal and vertical sides covering the drawing.

back edge$_1$ – for a dfs-tree in a directed graph: an edge directed from descendant to ancestor.

back edge$_2$ – for a dfs-tree in a undirected graph: any non-tree edge.

bend in a polyline drawing: a point where two segments belonging to the same edge meet.

bfs: abbreviation for *breadth-first search.*

biconnected component – of an undirected graph: a maximal set of edges that has no cutpoint.

biconnected graph: an undirected graph with no cutpoint.

breadth-first search (bfs): a graph search method that finds shortest paths.

breadth-first tree: an ordered tree in which the children of a vertex x are the vertices discovered from x in a breadth-first search.

bridge$_1$ – in an undirected graph: a cutedge, i.e., an edge whose removal disconnects the graph.

bridge$_2$ – in a mixed graph: an edge whose removal disconnects the underlying undirected graph.

bridge$_3$ – in a flow graph with root r: an edge (v, w) in a flow graph that belongs to every rw-path.

bridgeless graph: an undirected graph with no bridges.

bridge component (BC) – of a graph: a connected component of the graph that results when all the bridges are deleted.

bridge tree: the tree formed by contracting every bridge component of a connected graph.

bridgeless spanning subgraph, smallest: a bridgeless spanning subgraph of a connected bridgeless undirected graph that has the minimum possible number of edges.

certificate – for a graph property P and a graph G: a graph G' such that G has property P if and only if G' has the property.

chromatic index – of a graph: the minimum number of colors needed to color all edges such that edges with a common endpoint receive different colors.

chromatic number – of a graph: the minimum number of colors needed to color all vertices such that adjacent vertices receive different colors.

clique: a maximal set of vertices that are pairwise adjacent; sometimes maximality is not required.

cliquewidth-k graph – defined recursively (with $[k]$ denoting the set $\{1, 2, \ldots, k\}$):

- Any graph G with $V(G) = \{v\}$ and $l(v) \in [k]$ is a cliquewidth-k graph.
- If G_1 and G_2 are cliquewidth-k graphs and $i, j \in [k]$, then
  1. the disjoint union $G_1 \cup G_2$ is a cliquewidth-k graph.
  2. the graph $(G_1)_{i \times j}$ is a cliquewidth-k graph, where $(G_1)_{i \times j}$ is formed from G_1 by adding all edges (v_1, v_2) such that $l(v_1) = i$ and $l(v_2) = j$.
  3. the graph $(G_1)_{i \to j}$ is a cliquewidth-k graph, where $(G_1)_{i \to j}$ is formed from G_1 by switching all vertices with label i to label j.

cluster – in dynamic graph algorithms: a connected subgraph, subject to various additional problem-specific conditions.

clustering technique – used in the design of dynamic algorithms: a technique based on partitioning the graph into a suitable collection of connected subgraphs called clusters, such that each update involves only a small number of such clusters.

cograph – defined recursively:

- A graph with a single vertex is a cograph.
- If G_1 and G_2 are cographs, then the the disjoint union $G_1 \cup G_2$ is a cograph.
- If G_1 and G_2 are cographs, then the cross-product $G_1 \times G_2$ is a cograph, which is formed by taking the union of G_1 and G_2 and adding all edges (v_1, v_2) where v_1 is in G_1 and v_2 is in G_2.

convex drawing: a planar straight-line drawing such that the boundary of each face is a convex polygon.

cross edge – for a *dfs-tree* in a directed graph: a nontree edge joining two unrelated vertices.

crossing: a point where two edges intersect.

cutedge: an edge whose removal disconnects a graph.

cutpoint: a vertex whose removal disconnects a graph.

dag: acronym for "directed acyclic graph", i.e., a directed graph with no (directed) cycle.

Delaunay drawable graph: a planar triangulated graph that admits a drawing that is a Delaunay triangulation.

Delaunay triangulation: a planar straight-line drawing with all internal faces triangles, and such that three vertices form a face if and only if their convex hull does not contain any other vertex of the triangulation.

depth-first search (dfs): a graph search method that iteratively scans an edge incident to the most recently discovered vertex that still has unscanned edges.

depth-first spanning forest$_1$ – in an undirected graph: a collection of depth-first trees, one for each connected component of the graph.

depth-first spanning forest$_2$ – in a directed graph: a collection of depth-first trees containing every vertex once, with all cross edges joining two different trees directed from right to left.

depth-first tree (dfs-tree) – in a graph: an ordered tree in which the children of a vertex x are the vertices discovered from x in a depth-first search.

descendant – of a vertex y in a rooted tree: a vertex x such that y lies on the unique path from the root to x.

___, **proper** – of a vertex y in a rooted tree: a descendant other than vertex y itself.

dfs: abbreviation for *depth-first search.*

diameter – of a graph: the maximum distance between two vertices.

discovery (of a vertex): when a search reaches that vertex for the first time.

discovery order – induced by a graph search algorithm: a numbering of the vertices from 1 to $|V|$, in the order the vertices are discovered in the search, i.e., the preorder of the search tree.

distance – from vertex u to vertex v: length of a shortest uv-path.

dominance drawing: an upward drawing of an acyclic digraph, such that there exists a directed path from vertex u to vertex v if and only if $x(u) \leq x(v)$ and $y(u) \leq y(v)$, where $x(\cdot)$ and $y(\cdot)$ denote the coordinates of a vertex.

dominating set – of a graph: a set of vertices such that every vertex is either in this set or has a neighbor in this set.

domination of a vertex – in a flow graph: vertex v dominates vertex $w \neq v$ if every rw-path contains v.

dominator tree – for a flow graph: a tree that represents all the dominance relations.

dynamic graph problem: a problem concerned with efficiently answering queries regarding whether a dynamic graph has the specified property.

___, **connectivity**: answering a query whether the graph is connected, or whether two vertices are in the same connected component.

___, **decremental**: a partially dynamic problem in which only deletions are allowed.

___, **fully**: a problem in which the update operations include unrestricted insertions and deletions of edges or vertices.

___, **incremental**: a partially dynamic problem in which only insertions are allowed.

___, **minimum spanning tree**: the problem of maintaining a minimum spanning forest of a graph during insertions of edges, deletions of edges, and edge-cost changes.

___, **partially**: a problem in which only one type of update, either insertions or deletions, is allowed.

dynamic graph: a graph that is undergoing a sequence of updates; formally, a graph-valued variable.

dynamic programming: evaluation of a recursive formula in such a way as to avoid repeated computations.

k-ECSS: a k-edge-connected spanning subgraph of a k-edge-connected graph.

___, **smallest**: a k-ECSS that has the minimum possible number of edges.

ET tree – for a tree T: a dynamic balanced binary tree whose leaves are the sequence of vertex occurrences in an Euler tour of T.

Euler tour of a tree T: in dynamic graph algorithms, a closed walk over the digraph obtained by replacing each edge of T by two directed edges with opposite direction, such that the walk traverses each edge exactly once.

face of a drawing: a region of the plane defined by a planar drawing.

finish-time at a vertex: when a search leaves that vertex for the last time.

finish-time order – induced by a graph search: a numbering of the vertices from 1 to $|V|$, in the order they are finished in a search, i.e., the postorder of the search tree.

flow graph: a directed graph with a distinguished root vertex r that can reach every vertex.

forward edge – for a dfs-tree in a directed graph: a nontree edge from ancestor to descendant.

grid drawing: a polyline drawing such that vertices, crossings, and bends have integer coordinates.

hamiltonian cycle – in a graph: a simple cycle that includes every vertex.

hamiltonian path – in a graph: a simple path that includes every vertex.

historical shortest path – in a dynamic graph: a path that has been a shortest path at some point during a sequence of updates of the graph, and such that none of its edges have been updated since then.

hv-drawing: an upward orthogonal straight-line drawing of a binary tree such that the drawings of the subtrees of each node are separated by a horizontal or vertical line.

imbedded (di)graph drawing problem: a planar (di)graph with a prespecified topological imbedding (i.e., set of faces) that must be preserved in the drawing.

immediate dominator $idom(w)$ – of a vertex w in a flow graph: the unique dominating vertex v such that every other dominator of w dominates v.

independent set – in a graph: a set of vertices that are pairwise non-adjacent.

interlacing segments – of a cycle C: two segments S, T such that either $|V(S) \cap V(T) \cap V(C)| \geq 3$, or there are four distinct vertices u, v, w, x, which occur in that cyclic order along C, such that $u, w \in S$ and $v, x \in T$.

internal vertex – of path: a vertex that is not an endpoint of the path.

in-tree: a tree representing paths of a graph that lead to a given vertex.

irreducible Markov chain: a Markov chain whose transition graph remains strongly connected after all null-probable edges are deleted.

layer of vertices – in a rooted graph or digraph: a vertex subset comprising all the vertices at a given distance from the root.

layered drawing: a drawing of a rooted graph such that vertices at the same distance from the root lie on the same horizontal line.

left of a vertex y – in an ordered tree: a vertex x such that some common ancestor of x and y has children c and d, with c preceding d as siblings.

length function – on a graph: an assignment of numerical lengths to the edges, usually nonnegative numbers.

matching, perfect matching – in a graph: a spanning subgraph such that every vertex has degree exactly 1.

matching – in a graph: a set of edges that share no common endpoints.

mixed graph: a graph in which directed and undirected edges both occur.

monadic second-order logic (MSOL): a type of logic in which each variable may represent an individual element (vertex or edge) or a set of elements (vertex set or edge set); see Definition 10 of §10.4.

___, **MSOL′** – for a graph $G = (V, E)$: the subset of MSOL expressions restricted to variables v_i with domain V, e_i with domain E, and V_i with domain 2^V.

open ear decomposition – of an undirected graph: a partition of the edges into a simple cycle P_0 and simple paths $P_1, \ldots, P_k$ such that for each $i > 0$, P_i is joined to P_0 and previous paths only at its (2 distinct) ends.

ordered tree: a rooted tree in which the children of each vertex are linearly ordered.

orientation – of a graph: assignment of a unique direction to each undirected edge.

orthogonal drawing: a drawing in which each edge is a chain of horizontal and vertical segments.

orthogonal representation: a representation of orthogonal drawing in terms of bends along each edge and angles around each vertex.

out-tree: a tree representing paths of a graph that originate from a given vertex.

uv-**path**: a path starting at vertex u and ending at vertex v.

planar drawing: a drawing in which no two edges cross.

polyline drawing: a drawing in which each edge is a polygonal chain.

postorder – of an ordered tree: the finish-time order of a depth first search of the tree itself.

potentially uniform path – of a graph G: a path such that every proper subpath is a historical shortest path.

predicate calculus: a type of logic in which predicates have arguments and expressions are built using various operators ($\neg$, $\wedge$, $\vee$), and in which the quantifiers ($\forall$, $\exists$).

preorder – of an ordered tree: the discovery order of a depth first search of the tree itself.

proximity drawing: a drawing of a graph based on a geometric proximity relation.

recursively constructed graph class: defined by a set (usually finite) of primitive or *base graphs*, in addition to one or more operations that compose larger graphs from smaller subgraphs; each operation involves either fusing specific vertices from each subgraph or adding new edges between specific vertices from each subgraph.

reducible flow graph: a flow graph that can be transformed into its root vertex r by a sequence of reduction operations; that is, if e is the only edge entering a vertex $w \neq r$, then contract edge e (and its other endpoint) into the vertex v.

related vertices – in a rooted tree: two vertices such that one is an ancestor of the other.

scanning an edge: the work done by a graph searching algorithm when it traverses the edge.

searching a graph: a methodical (linear-time) exploration of all the vertices and edges of a graph.

segment – of a cycle C in a biconnected graph G: either (i) a chord of C (i.e., an edge not in C that joins two vertices of C); or (ii) a connected component of the graph $G - V(C)$, plus all the edges of G joining that component to C.

semidominator: a useful intermediate concept in computing dominators, defined in terms of a depth-first search tree.

separation pair: two vertices in a biconnected graph whose removal disconnects the graph.

series-parallel graph – a recursively defined graph: see Definition 5 of §10.4.

set-merging problem: the problem of maintaining a partition of a given universe subject to a sequence of *union* and *find* operations.

shortest-path tree – in a rooted graph: a tree in which the path from the root r to any vertex v is a shortest rv-path.

sink – in a directed graph: a vertex with outdegree 0.

source – in a directed graph: a vertex with indegree 0.

sparse graph: a graph with at most $O(|V|)$ edges.

sparsification technique: a technique for speeding up dynamic graph algorithms, which when applicable, transforms a time bound of $T(|V|, |E|)$ into $O(T(|V|, |V|))$.

start vertex: the distinguished vertex of a flow graph.

straight-line drawing: a drawing in which each edge is a straight-line segment.

strong component (SC) – of a directed graph: a maximal subgraph in which any two vertices are reachable from each other.

strong component graph (SC graph) – for a directed graph: the result of contracting every strong component to a vertex; also called the *condensation graph*.

strongly connected digraph: a digraph in which every vertex can reach every other vertex by a directed path.

top tree – in dynamic graph algorithms: a tree that describes a hierarchical partition of the edges of another tree, well suited to maintaining path information.

topological numbering (topological order, topological sort) – of an acyclic directed graph: assignment of an integer to each vertex so that each edge is directed from a lower number to a higher number.

topology tree: in dynamic graph algorithms, a hierarchical representation of a tree T into clusters.

tournament: a directed simple graph such that each pair of vertices is joined by exactly one edge.

traversable mixed graph: a mixed graph in which every vertex can reach every other by a path with all its directed edges pointing in the forward direction.

tree: a connected graph with no cycles and sometimes with a designated *root* used to describe recursive constructions.

___, **drawable as a minimum spanning tree**: a tree T such that there exists a set P of points (especially in $\mathbb{R}^2$ or $\mathbb{R}^3$) such that the minimum spanning tree of P (using Euclidean metric distances) is isomorphic to T.

___, **rooted, recursively defined**: the graph whose only vertex is the root r (and no edges); or the result of joining the roots of two disjoint trees with a new edge.

tree edge: edge of a spanning tree in a graph.

treewidth – of a graph G: the minimum width taken over all tree-decompositions of G; measures how closely the graph resembles a tree.

treewidth-k graph: a graph whose treewidth is no greater than k.

triconnected graph: an undirected connected graph that remains connected whenever any two or fewer vertices are deleted.

uniform path – of a graph G: a path such that every proper subpath is a shortest path.

update on a graph: an operation that inserts or deletes edges or vertices of the graph or changes attributes associated with edges or vertices, such as cost or color.

upward drawing: a drawing of a digraph such that each edge is monotonically nondecreasing in the vertical direction.

upward planar digraph: a digraph that admits an upward planar drawing.

vertex cover – of a graph: a set of vertices such that every edge is incident on at least one vertex in the set.

visibility drawing: a drawing of a graph based on a geometric visibility relation; e.g., the vertices might be drawn as horizontal segments, and the edges associated with vertically visible segments.

Voronoi diagram: the dual graph of a Delaunay triangulation.

Chapter 11

NETWORKS and FLOWS

11.1 MAXIMUM FLOWS

Clifford Stein, Columbia University

11.2 MINIMUM COST FLOWS

Lisa Fleischer, IBM/Carnegie Mellon University

11.3 MATCHINGS and ASSIGNMENTS

Douglas R. Shier, Clemson University

11.4 COMMUNICATION NETWORK DESIGN MODELS

Prakash Mirchandani, University of Pittsburgh
David Simchi-Levi, Massachusetts Institute of Technology

GLOSSARY

11.1 MAXIMUM FLOWS

Clifford Stein, Columbia University

11.1.1 The Basic Maximum Flow Problem
11.1.2 Minimum Cuts and Duality
11.1.3 Max-Flow Min-Cut Theorem
11.1.4 Algorithms for Maximum Flow
11.1.5 Variants and Extensions of Maximum Flow
References

Introduction

The *Maximum Flow Problem* is one of the basic problems in combinatorial optimization. It models a large variety of problems in a diverse set of application areas including data flowing through a communications network, power flowing through an electrical network, liquids flowing through pipes and parts flowing through a factory. It has also served as a prototypical problem in algorithm design, and many useful and powerful ideas were first introduced in the context of maximum flows. This section will describe maximum flows and some generalizations. The book by Ahuja, Magnanti and Orlin [AhMaOr93] is an excellent reference and includes a large number of applications. Other texts and surveys with significant coverage of maximum flows include [CoCuPuSc98], [Ev79], [La76], [PaSt82], [Ta83], and [GoTaTa90].

11.1.1 The Basic Maximum Flow Problem

Informally, in a maximum flow problem, we wish to send as much stuff as possible from one place in a network to another, while limiting the amount of stuff sent through an arc by the capacity of that arc.

DEFINITIONS

D1: An ***s-t* (flow) network** $G = (V, E, s, t, cap)$ is a directed graph with vertex-set V and arc-set E, two distinguished vertices, a ***source*** s and a ***sink*** t, and a nonnegative ***capacity function*** $cap : E \to N$. We adopt the convention, without loss of generality, that if arc $(v, w) \in E$ then the reverse arc $(w, v) \notin E$. (See subsection 11.1.4.)

D2: A ***(feasible) flow*** is a function $f : E \to R$ which obeys three types of constraints:

capacity constraints: $f(v, w) \le cap(v, w)$, for each arc $(v, w) \in E$.

conservation constraints:

$$\sum_{(w,v)\in E} f(w, v) = \sum_{(v,w)\in E} f(v, w) \text{ for each vertex } v \in V - \{s, t\}.$$

nonnegativity constraints: $f(v, w) \ge 0$, for each arc $(v, w) \in E$.

D3: The ***value of a flow*** f, denoted $val(f)$ (or $|f|$), is the total flow into the sink, i.e.,

$$val(f) = \sum_{(v,t)\in E} f(v,t)$$

D4: In the ***maximum flow problem***, we are given a flow network $G = (V, E, s, t, cap)$ and wish to find a flow f of maximum value.

EXAMPLE

E1: An example of a flow network appears in Figure 11.1.1. A maximum flow for the network appears in Figure 11.1.2. The numbers on the arcs are capacities, the flows appear in parentheses. It is straightforward to verify that the three properties of a flow are satisfied.

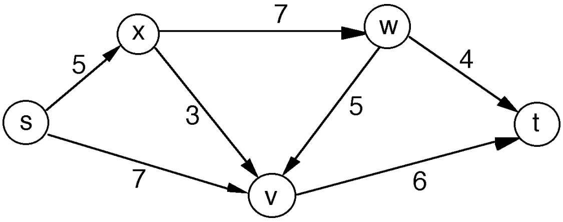


Figure 11.1.1 **An s-t network.**

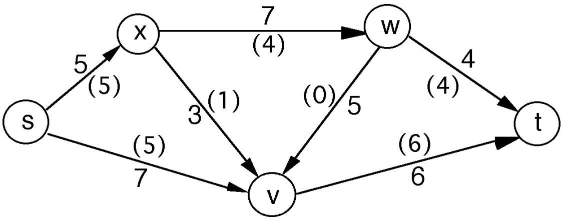


Figure 11.1.2 **A maximum flow of value 10.**

FACTS

F1: The flow conservation properties imply that the value of a flow is also equal to total flow out of the source, i.e.,

$$val(f) = \sum_{(v,t)\in E} f(v,t) = \sum_{(s,v)\in E} f(s,v)$$

F2: Although a flow is defined as a real-valued function, an integer-valued maximum flow always exists (since the capacity function is integer-valued).

11.1.2 Minimum Cuts and Duality

An important and *dual concept* related to maximum flows is that of *minimum cuts.*

Cuts in a Network

An *s-t cut* combines the concepts of *partition-cut* (§6.4) and an $(s|t)$-*disconnecting set* of edges. (§4.1).

DEFINITIONS

D5: Let $G = (V, E, s, t, cap)$ be an *s-t* network, and let S and T form a partition of V such that source $s \in S$ and sink $t \in T$. Then the set of all arcs that have one endpoint in set S and the other endpoint in set T is called an ***s-t* cut** of network G and is denoted $\langle S, T\rangle$. An arc (v, w) is a ***forward arc*** of the cut if $v \in S$ and $w \in T$, and (v, w) is a ***backward arc*** if $v \in T$ and $w \in S$.

D6: The ***capacity of an s-t cut*** $\langle S, T\rangle$, denoted $cap\langle S, T\rangle$, is the sum of the capacities of the forward arcs of the cut in the forward direction, i.e.,

$$cap\langle S, T\rangle = \sum_{(v,w)\in E: v\in S, w\in T} cap(v, w)$$

D7: A ***minimum s-t cut*** in a flow network G is a cut of minimum value, that is $\min\{cap\langle S, T\rangle : \langle S, T\rangle \text{ is an } s\text{-}t \text{ cut}\}$.

EXAMPLES

E2: The *s-t* cut $\langle\{s, x, v\}, \{w, t\}\rangle = \{(x, w), (v, t)\}$ appears in Figure 11.1.3. The capacity of the cut $cap\langle S, T\rangle = cap(x, w) + cap(v, t) = 13$. Notice that (w, v) is a backward arc and hence its capacity is not included in the capacity of the cut.

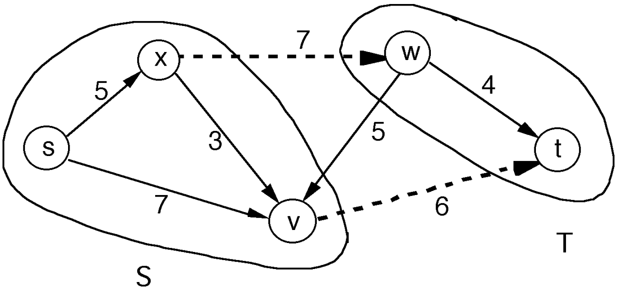


Figure 11.1.3 **The *s-t* cut $\langle\{s, x, v\}, \{w, t\}\rangle = \{(x, w), (v, t)\}$ has capacity 13.**

E3: A minimum cut of capacity 10 appears in Figure 11.1.4.

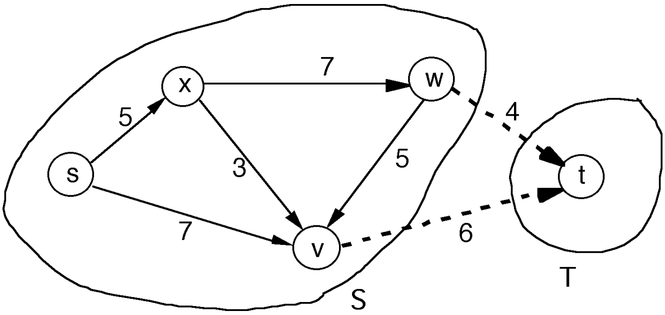


Figure 11.1.4 **A minimum cut of capacity 10.**

Weak Duality

DEFINITION

D8: Given a flow f and an s-t cut $\langle S,T\rangle$, the ***flow across cut*** $\langle S,T\rangle$, denoted $f\langle S,T\rangle$, is the sum of the flows on the forward arcs minus the sum of the flows on the backward arcs, i.e.,

$$f\langle S,T\rangle = \sum_{(v,w)\in E: v\in S, w\in T} f(v,w) - \sum_{(v,w)\in E: v\in T, w\in S} f(v,w).$$

FACTS

F3: Flow conservation implies that for a given flow f and *any* cut $\langle S,T\rangle$, $val(f) = f\langle S,T\rangle$.

F4: (*Weak Duality*) Let f be any flow in an s-t network G, and let $\langle S,T\rangle$ be any s-t cut. Then

$$val(f) \leq cap\langle S,T\rangle$$

F5: Let f be a flow in an s-t network G and $\langle S,T\rangle$ an s-t cut, and suppose that $val(f) = cap\langle S,T\rangle$. Then flow f is a maximum flow in network G, and $\langle S,T\rangle$ a minimum s-t cut. (This is an immediate consequence of weak duality [Fact 4].)

EXAMPLE

E4: The flow f (in parentheses) and s-t cut $\langle S,T\rangle$ shown in Figure 11.1.5 illustrate Fact 3. In particular, $val(f) = 6$, and the flow across the cut, $f\langle S,T\rangle$, equals $f(x,w) + f(v,t) - f(w,v) = 2 + 5 - 1 = 6$.

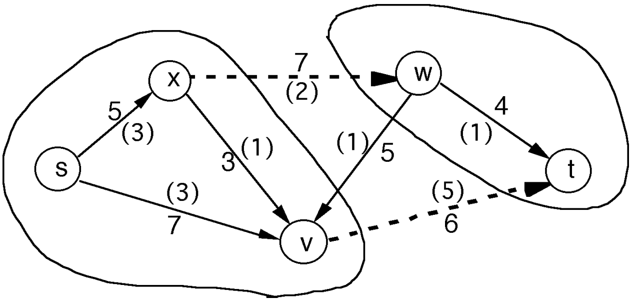


Figure 11.1.5 $val(f) = f\langle S,T\rangle = 6$.

11.1.3 Max-Flow Min-Cut Theorem

The relationship between the maximum-flow problem and its dual, the minimum-cut problem, is an example of *strong max-min duality* that occurs between certain optimization problems and their dual problems. Two other instances are König's theorem (Section 11.3), which states that the size of a minimum vertex cover in a bipartite graph equals the size of a maximum matching, and Menger's theorem (Section 4.1), which relates the local connectivity between two vertices of any graph and the number of internally–disjoint paths between them.

The Residual Network and Flow-Augmenting Paths

Algorithms that iteratively increase the flow in a network via *flow-augmenting paths* often use an associated digraph, called a *residual network*, for finding these paths more easily.

DEFINITIONS

D9: Let f be the current flow in a network $G = (V, E, s, t, cap)$. An arc $(v, w) \in E$ is ***increasable*** if $f(v, w) < cap(v, w)$ and is ***reducible*** if $f(v, w) > 0$.

NOTATION: Let I denote the set of all increasable arcs, and let R be the set of all reducible arcs. (Of course, in general, $I \cap R \neq \emptyset$.)

D10: Given a flow f in a network $G = (V, E, s, t, cap)$, the ***residual network*** $G_f = (V, E_f, s, t, r_f)$ has vertex-set V, and the arc-set E_f is constructed from network G as follows: for each arc $(v, w) \in E$, if arc $(v, w) \in I$, then create an arc (v, w) in G_f, and label it with a ***residual capacity*** $r_f(v, w) = cap(v, w) - f(v, w)$; if arc $(v, w) \in R$, then create an arc (w, v) in G_f, and label it $r_f(w, v) = f(v, w)$.

D11: Given a flow f in a network $G = (V, E, s, t, cap)$, a ***flow-augmenting path*** P for network G is a directed s-t path in the residual network G_f. The ***capacity of flow-augmenting path*** P, denoted Δ_P, is given by $\Delta_P = \min_{(v,w) \in E_f} r_f(v, w)$.

REMARK

R1: It follows from the definitions that the capacity Δ_P is always positive.

FACTS

F6: [Flow Augmentation] Let f be a flow in a network $G = (V, E, s, t, cap)$, and let P be a flow-augmenting path with capacity Δ_P. Let f_P be defined as follows:

$$f_P(v, w) = \begin{cases} f(v, w) + \Delta_P & \text{if } (v, w) \in E(P) \cap I \\ f(v, w) - \Delta_P & \text{if } (w, v) \in E(P) \text{ and } (v, w) \in R \\ f(v, w) & \text{otherwise} \end{cases}$$

Then f_P is a feasible flow in network G and $val(f_P) = val(f) + \Delta_P$.

F7: [Characterization of Maximum Flow] Let f be flow in a network G. Then f is a maximum flow if and only if there does not exist an f-augmenting path in G.

F8: **Max-Flow Min-Cut** [FuDa55, ElFeSh56, FoFu56] For a given network, the value of a maximum flow is equal to the capacity of a minimum cut.

EXAMPLES

E5: The current flow f in the network G shown at the top in Figure 11.1.6 has value 9. The corresponding residual network G_f is shown at the bottom.

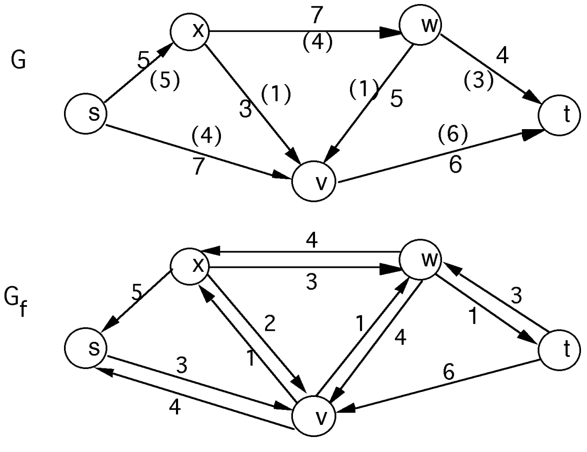


Figure 11.1.6 **A network G with flow f and its residual network G_f.**

There are exactly two directed paths from s to t in the residual network, each corresponding to a different flow-augmenting path for increasing the flow. One of them is the directed path $s \to v \to w \to t$ in the network G_f. The arc (v, w) in path P corresponds to the reducible arc (w, v) in network G. Notice that $\Delta_P = 1$.

E6: If the flow f given in Figure 11.1.6 above is augmented by the flow-augmenting path P in Example 5, then the resulting flow is as shown in Figure 11.1.7. Observe that the one reducible arc, (w, v), of the flow-augmenting path P reduces the flow on that arc by 1.

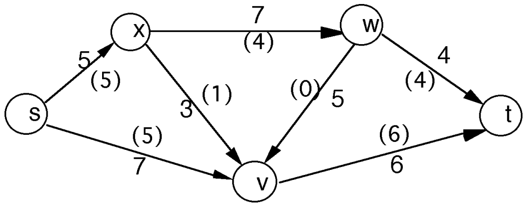


Figure 11.1.7 **Increasing the flow in Example 5 by 1.**

A simple application of Fact 5 shows that f_P is a maximum flow. Also straightforward to show is the non-existence of a flow-augmenting path corresponding to f_P.

11.1.4 Algorithms for Maximum Flow

It is possible to compute a maximum flow in polynomial time. There are three popular approaches. The first uses flow-augmenting paths, the second is to use a *push-relabel* method, and the third uses *linear programming.* We discuss the first two methods briefly.

Ford-Fulkerson Algorithm

The first published maximum-flow algorithm, due to Ford and Fulkerson [FoFu62], is essentially a *greedy* method: we iteratively push flow along flow-augmenting paths from source to sink.

Algorithm 11.1.1: Ford-Fulkerson Algorithm

```
Input: a flow network $G = (V, E, s, t, cap)$
Output: a maximum flow $f$

    Initialize $f(v, w) = 0$ for all $(v, w) \in E$.
    Calculate residual network $G_f$
    While an augmenting path in $G_f$ exists
        Let $P$ be an augmenting path in $G_f$ with capacity $\Delta_P$
        Obtain increased flow $f_P$ using flow-augmenting path $P$ (Fact 6).
        $f := f_P$
        Update residual network $G_f$
```

COMPUTATIONAL NOTE: The efficiency of the Ford-Fulkerson algorithm depends on which augmenting path is chosen, and on the data structures used to facilitate the computation. If an arbitrary augmenting path is chosen, the algorithm may not run in polynomial time [Za72]. However, many natural choices of a path lead to a polynomial time algorithm. The first such algorithm, due to Edmonds and Karp [EdKa72], and Dinitz [Di70], always chooses the shortest augmenting path, where the length of a path is defined to be the number of arcs on the path. This algorithm runs in $O(|E||V|)$ iterations of the main loop. Each iteration requires a breadth-first search and some updates of flow variables, which can be done in $O(|E|)$ time, and hence the algorithm runs in $O(|E|^2|V|)$ time. Many further improvements are possible with data structures such as dynamic trees [SlTa83] or by augmenting flow on several shortest paths simultaneously [Ka74]. (These are known as blocking flows.) The current fastest running time using this approach is due to Goldberg and Rao [GoRa98].

Preflow-Push Algorithms

An alternative approach to computing a maximum flow, called a *preflow-push* algorithm, was introduced by Goldberg [Go87] and Goldberg and Tarjan [GoTa88]. It uses a *push-relabel* strategy, which pushes flow over individual arcs, rather than paths. To do so, it allows flow to "accumulate" at some vertices, creating an *excess* at those vertices. The *push* operation selects one of these *active* vertices and tries to remove its excess by pushing flow to neighbors that are"closer" to the sink. The *relabel* operation maintains distance labels that help keep track of these neighbors.

DEFINITIONS

D12: A ***preflow*** is a relaxed version of a flow, a function $f : E \to R^+$ which obeys three types of constraints:

capacity constraints: $f(v, w) \leq cap(v, w)$, for each arc $(v, w) \in E$.

relaxed conservation constraints:

$$\sum_{(w,v)\in E} f(w, v) - \sum_{(v,w)\in E} f(v, w) \geq 0 \text{ for each vertex } v \in V - \{s, t\}.$$

nonnegativity constraints: $f(v, w) \geq 0$, for each arc $(v, w) \in E$.

D13: Let f be a preflow in an s-t network $G = (V, E, s, t, cap)$. The ***excess*** at vertex v, denoted $e(v)$, is given by $e(v) = \sum_{(w,v)\in E} f(w,v) - \sum_{(v,w)\in E} f(v,w)$. A vertex v with $e(v) > 0$ is called an ***active*** vertex.

TERMINOLOGY: Given a preflow f, the sets of *increasable* arcs, I, and *reducible* arcs, R, are defined as in the case when f is a feasible flow (Definition 9). Similarly, the residual network $G_f = (V, E_f, s, t, r_f)$ is defined in the same way as before (Definition 10).

D14: Let f be a preflow in an s-t network $G = (V, E, s, t, cap)$. A ***distance function*** $h : V \to N$ satisfies:

$h(s) = |V|$,

$h(t) = 0$,

$h(v) \leq h(w) + 1$ for each arc $(v, w) \in E_f$ in the residual graph G_f.

D15: For a given preflow f and distance function h, an arc (v, w) in the residual network G_f is ***admissible*** if $h(v) = h(w) + 1$.

Algorithm 11.1.2: Preflow-Push Algorithm

```
Input: a flow network $G = (V, E, s, t, cap)$
Output: a maximum flow $f$
    {Initialization}
    $f(v, w) := 0$ for all $(v, w) \in E$.
    $h(v) := 0$ for all $v \in V$
    $h(s) := |V|$
    $f(s, w) := cap(s, w)$ for all arcs $(s, w) \in E$.
    Compute residual network $G_f$ and excesses $e$.
    While there are active vertices
        Select an active vertex $v$.
        If $G_f$ contains an admissible arc $(v, w)$
            {Push flow on arc $(v, w)$.}
            $\Delta := \min\{e(v), r_f(v, w)\}$
            $e(v) := e(v) - \Delta$
            $e(w) := e(w) + \Delta$
            If arc $(v, w) \in I$
                $f(v, w) := f(v, w) + \Delta$
            Else
                $f(v, w) := f(v, w) - \Delta$
        Else {Relabel $v$.}
            $h(v) := \min\{h(w) : (v, w) \in E_f$ and $r_f(v, w) > 0\}$
        Update residual network $G_f$.
```

COMPUTATIONAL NOTE: The preflow-push algorithm terminates in $O(|V|^2|E|)$ time using simple data structures. More careful selection of operations and careful use of data structures leads to algorithms with running times of $O(|V|^3)$ or $O(|V||E| \log(|V|^2/|E|))$ [GoTa88]. The fastest preflow-push algorithm is slightly faster than these and is due to King, Rao and Tarjan [KiRaTa94].

COMPUTATIONAL NOTE: In practice a good implementation of a preflow-push algorithm seems to be faster than a good implementation of an augmenting path algorithm. Two heuristics are essential ingredients in implementing a push relabel algorithm well. First, a backwards breadth-first search of the residual graph is performed periodically, in order to update distance labels. Second, the *gap heuristic*, which quickly identifies vertices that must be on the sink side of the minimum s-t cut, is employed [GoCh97].

11.1.5 Variants and Extensions of Maximum Flow

We briefly mention some variations and extensions of the basic maximum-flow problem. For more extensive coverage, see, e.g., [AhMaOr93, EvMi92].

FACTS

F9: The convention that if arc $(v, w) \in E$ then the reverse arc $(w, v) \notin E$ is without loss of generality. Any maximum flow algorithm can easily be extended to handle this case. Alternatively, arc (v, w) can be converted to two arcs (v, x) and (x, w) each with capacity $cap(v, w)$, and arc (w, v) can be converted to two arcs (w, y) and (y, v) each with capacity $cap(w, v)$.

F10: [multiple-source multiple-sink] Suppose that we have a flow network G with multiple sources $\{s_1, \ldots, s_k\}$ and multiple sinks, $\{t_1, \ldots, t_\ell\}$. We can still find a maximum flow in this network by the following transformation. We create a new supersource s and supersink t, and add an arc (s, s_i) with $cap(s, s_i) = \infty$ for each source s_i, and an arc (t_i, t) with $cap(t_i, t) = \infty$ for each sink t_i. A maximum flow in this network is easily interpreted as a maximum flow for the multiple-source multiple-sink problem.

F11: Maximum flow can be used to find a maximum matching in a bipartite graph. (Matchings are discussed in Section 11.3.)

F12: If each arc has a cost, then we obtain the *minimum-cost flow problem* (see Section 11.2).

F13: [flow on undirected edges] The network G can contain undirected as well as directed edges. An undirected edge (v, w) with capacity $cap(v, w)$ is understood to be an edge that can carry up to $cap(v, w)$ units of flow in either direction. In any flow, it is only necessary that flow be carried in one direction or the other, as flow in opposing directions cancel each other out. For example, an undirected edge with flow $f(v, w) = 3$ and $f(w, v) = 2$ is equivalent to an edge with $f(v, w) = 1$ and no flow in the (w, v) direction. We can therefore convert an undirected edge to two oppositely directed arcs, each with capacity $cap(v, w)$.

F14: [lower bounds on flow] The network G can also contain lower bounds on the flow over an arc. It is still possible to find a maximum flow in such a graph, providing that one exists (see, e.g., [AhMaOr93, EvMi92]).

Multicommodity Flow

Perhaps the most important extension of maximum flow is the extension to the case of multiple commodities.

DEFINITIONS

D16: A ***commodity*** i is a triple (s_i, t_i, d_i) where s is a source, t is a sink, and d is a ***demand***, or amount of flow to be routed.

D17: A ***multicommodity flow network*** $G = (V, E, K, cap)$ is a directed graph with vertex set V and arc-set E, commodity set K, and a nonnegative ***capacity*** function $cap : E \to N$. We adopt the convention that if arc $(v, w) \in E$ then the reverse arc $(w, v) \notin E$. The commodities are indexed by the integers $1, 2, \ldots, k$.

D18: A ***multicommodity flow*** in a multicommodity flow network $G = (V, E, K, cap)$ is a set of $k = |K|$ functions $f_i : E \to R^+$ satisfying the following conditions:

joint capacity constraints: $\sum_{i=1}^{k} f_i(v, w) \leq cap(v, w)$, for each arc $(v, w) \in E$.

conservation constraints:

$\sum_{(w,v)\in E} f_i(w, v) = \sum_{(v,w)\in E} f_i(v, w)$ for each $v \in V - \{s_i, t_i\}$ and $i = 1, \ldots, k$, and $\sum_{(w,t_i)\in E} f_i(w, t_i) - \sum_{(t_i,w)\in E} f_i(t_i, w) = d_i$ for each $i = 1, \ldots, k$.

nonnegativity constraints: $f_i(v, w) \geq 0$, for each arc $(v, w) \in E$ and each $i = 1, \ldots, k$.

Variants of Multicommodity Flow Problems

DEFINITIONS

D19: In the ***feasible multicommodity flow problem***, we are given a multicommodity flow network G, and wish to know if a multicommodity flow exists. We call such a flow a ***feasible multicommodity flow***.

D20: In the ***concurrent flow problem***, we are given a multicommodity flow network G, and we wish to compute the maximum value z for which there is a feasible multicommodity flow in the network with all demands multiplied by z.

D21: In the ***maximum multicommodity flow problem***, we are given a multicommodity flow network, except that for each commodity, we are *not* given a demand. We wish to find, for each commodity i, a flow f_i of value (demand) $val(f_i)$ such that $\sum_{i=1}^{k} val(f_i)$ is maximized.

COMPUTATIONAL NOTE: If we do not require that the flows be integral (even though capacities and demands are integral), then all the above multicommodity flow problems can be solved in polynomial time via linear programming. More efficient combinatorial algorithms that compute approximately optimal solutions also exist [LeMaPlStTaTr94, Yo95, and GaKo98].

COMPUTATIONAL NOTE: If we require that all flows be integral, then all the above problems are NP-hard. The degree to which we can approximate them varies from problem to problem. (See [Va99] for a survey.) Note that integral multicommodity flow generalizes the *disjoint-paths problem.*

D22: In an ***unsplittable flow problem***, we have the additional restriction that each commodity must be routed on one path. All variants of this problem are NP-hard, but constant factor approximation algorithms exist for single-source multiple-sink variants [Kl96, KoSt97, DiGaGo98].

References

[AhMaOr93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, 1993.

[CoCuPuSc98] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, John Wiley and Sons, New York, 1998.

[ChGo97] B. V. Cherkassky and A. V. Goldberg, On implementing the push-relabel method for the maximum flow problem, *Algorithmica* 19(4) (1997), 390–410.

[Di70] E. A. Dinic, Algorithm for solution of a problem of maximum flow in networks with power estimation, *Soviet Math. Dokl.* 11 (1970), 1277–1280.

[DiGaGo98] Y. Dinitz, N. Garg, and M. Goemans, On the single source unsplittable flow problem, In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science* (1998), 290–299.

[EdKa72] J. Edmonds and R. M. Karp, Theoretical improvements in the algorithmic efficiency for network flow problems, *Journal of the ACM* 19 (1972), 248–264.

[ElFeSh56] P. Elias, A. Feinstein, and C. E. Shannon, Note on maximum flow through a network, *IRE Transactions on Information Theory IT-2* 19 (1956), 117–119.

[Ev79] S. Even, *Graph Algorithms*, Computer Science Press, 1979.

[EvMi92] *Optimization Algorithms for Networks and Graphs*, J. R. Evans and E. Minieka, Dekker, 1992.

[FoFu56] L. R. Ford, Jr. and D. R. Fulkerson, Maximal flow through a network, *Canadian J. of Math.* 8 (1956), 399–404.

[FoFu62] L. R. Ford, Jr. and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[FuDa55] D. R. Fulkerson and G. B. Dantzig, Computation of maximum flow in networks, *Naval Research Logistics Quarterly* 2 (1955), 277–283.

[GaKo98] N. Garg and J. Konemann, Faster and simpler algorithms for multicommodity flow and other fractional packing problems, In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science* (1998), 300–309.

[GoTaTa90] A. V. Goldberg, É. Tardos, and R. E. Tarjan, Network flow algorithms, In B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, editors, *Paths, Flows, and VLSI-Layout*, pp. 101–164, Springer-Verlag, 1990.

[Go87] A. V. Goldberg, *Efficient graph algorithms for sequential and parallel computers*, PhD thesis, MIT, Cambridge, MA, January 1987.

[GoRa98] A. V. Goldberg and S. Rao, Beyond the flow decomposition barrier. *Journal of the ACM* 45 (1998), 783–797.

[GoTa88] A. V. Goldberg and Robert E. Tarjan, A new approach to the maximum flow problem, *Journal of the ACM* 35 (1988), 921–940.

[Ka74] A. V. Karzanov, Determining the maximal flow in a network by the method of preflows, *Soviet Math. Dokl.* 15 (1974), 434–437.

[KiRaTa94] V. King, S. Rao, and R. E. Tarjan, A faster deterministic maximum flow algorithm, *Journal of Algorithms* 17 (1994), 447–474.

[Kl96] J. M. Kleinberg, Single-source unsplittable flow, In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science* (1996), 68–77.

[KoSt97] S. G. Kolliopoulos and C. Stein, Improved approximation algorithms for unsplittable flow problems, In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science* (1997), 426–436.

[La76] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.

[LeMaPlStTaTr95] T. Leighton, F. Makedon, S. Plotkin, C. Stein, É. Tardos, and S. Tragoudas, Fast approximation algorithms for multicommodity flow problems, *Journal of Computer and System Sciences* 50 (1995), 228–243.

[PaSt82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, 1982.

[SlTa83] D. Sleator and R. E. Tarjan, A data structure for dynamic trees, *Journal of Computer and System Sciences* 26 (1983), 362–391.

[Va01] V. Vazirani, *Approximation Algorithms*, Springer-Verlag, 2001.

[Yo95] N. Young, Randomized rounding without solving the linear program. In *Proceedings of the 6th ACM-SIAM Symposium on Discrete Algorithms* (1995), 170–178.

[Za72] N. Zadeh, Theoretical Efficiency of the Edmonds-Karp algorithm for computing maximal flows, *Journal of the ACM* 19 (1972), 184–192.

11.2 MINIMUM COST FLOWS

Lisa Fleischer, IBM/Carnegie Mellon University

11.2.1 Basic Model and Definitions
11.2.2 Optimality Conditions
11.2.3 The Dual Problem
11.2.4 Algorithms for Minimum Cost Flow
11.2.5 Extensions of Minimum Cost Flow
References

Introduction

Minimum cost flows are a powerful and useful network flow model that is distinguished by supply nodes, demand nodes, and linear flow costs on the edges of directed network. They are used to model complex problems occurring in transportation, transshipment, manufacturing, telecommunications, graph drawing, human resources, statistics, numerical algebra, physics, and many other engineering disciplines. Minimum cost flows generalize many other network problems. They lie on the tractable side of a boundary between computable and intractable problems: while there exist efficient algorithms to find integer minimum cost flows, most generalizations of integer minimum cost flow problems are NP-hard.

The book "Network Flows" by Ahuja, Magnanti, and Orlin provides thorough coverage of minimum cost flows, applications, and related topics [AhMaOr93]. Other recent texts and surveys with significant coverage of minimum cost flows include [CoCuPuSc98,GoTaTa90,IwMcSh00,Sc03].

11.2.1 The Basic Model and Definitions

DEFINITIONS

D1: A ***(standard) flow network*** $G = (V, A, cap, c, b)$ is a directed graph with vertex-set V, arc-set A, a nonnegative capacity function $cap : A \to N$, a linear cost function $c : A \to Z$, and an integral supply vector $b : V \to Z$ that satisfies $\sum_{w \in V} b(w) = 0$.

D2: An ***s-t flow network*** (or ***single-source single-sink network***) is a flow network $G = (V, A, cap, c, b)$ that contains two distinguished vertices s and t such that $b(v) = 0$ for all $v \in V - \{s, t\}$ and $b(s) = -b(t) > 0$.

D3: An ***extended flow network*** $G' = (V', A', cap')$ of $G = (V, A, cap, c, b)$ is an s-t network with vertex-set $V' = V \cup \{s, t\}$, arc-set $A' = A \cup \{(s, v) | b(v) > 0\} \cup \{(w, t) | b(w) < 0\}$ and capacity function cap' defined by

$$cap'(v,w) = \begin{cases} cap(v,w), & \text{if } (v,w) \in A \\ b(v), & \text{if } v = s \\ -b(w), & \text{if } w = t \end{cases}$$

D4: A ***transshipment network*** is a flow network in which all arcs have infinite capacity.

D5: A ***(standard) flow*** (also called a ***(standard) feasible flow***) is a function $f : A \to Z$ that satisfies

capacity constraints: $f(v,w) \leq cap(v,w)$ for all $(v,w) \in A$,

nonnegativity constraints: $f(v,w) \geq 0$ for all $(v,w) \in A$,

flow conservation constraints: $\sum\limits_{w}[f(v,w) - f(w,v)] = b(v)$ for each $v \in V$.

D6: A flow f in an s-t flow network $G = (V, A, cap, c, b)$ is called an ***s-t flow*** and is said to have ***volume*** $b(s)$.

D7: A ***minimum cost flow*** is a flow f with minimum $c^T f$ value among all flows.

D8: A ***circulation*** is a flow for the supply vector $b \equiv \mathbf{0}$. A ***minimum cost circulation*** is a circulation f with minimum $c^T f$ value among all circulations.

NOTATION: Sometimes, we use the subscript notation f_{uv}, c_{uv}, cap_{uv} or b_v instead of $f(u,v)$, $c(u,v)$, $cap(u,v)$, or $b(v)$, respectively.

REMARK

R1: In the definitions above, we can extend the definition of the capacity function cap to *all* ordered pairs in $V \times V$ by defining $cap(v,w) = 0$ for $(v,w) \notin A$. This has the effect of extending any flow function f to all such ordered pairs as well. This extended view of the capacity and flow functions is notationally convenient for interpreting expressions like the flow conservation constraints given above and for various other expressions appearing later in this section.

EXAMPLE

E1: An example of a flow network appears on the left in Figure 11.2.1. The supply at each node is indicated in brackets. In parentheses at each arc is the arc cost followed by the arc capacity. A feasible flow in this network is indicated by the italic numerals in the figure on the right.

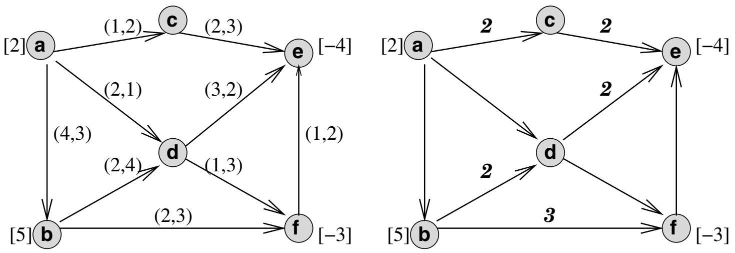


Figure 11.2.1 **An example of a flow network and a feasible flow.**

NOTATION: The following notation is used throughout this section:
(a) $n = |V|$ and $m = |A|$ (number of vertices and arcs, respectively, in the network).
(b) $C = \max\limits_{e \in A} |c(e)|$ (largest arc cost).
(c) $U = \max\limits_{e \in A}\{cap(e) | cap(e) < \infty\}$ (largest finite capacity).
(d) $B = \max\limits_{v \in V} |b(v)|$ (largest magnitude of a supply [or demand]).

(e) For an arc subset $F \subseteq A$, $c(F) \doteq \sum_{e \in F} c(e)$.

(f) $M = \sum_{e} c(e) cap(e)$

(g) $\mathcal{S}(n, m, C)$ denotes the time complexity of computing single source shortest paths from a fixed vertex in a directed network with n vertices m arcs and arc lengths (costs) bounded by C.

ASSUMPTIONS

A1: G has no parallel arcs and no oppositely directed pairs of arcs.

A2: No arc in G has negative cost.

FACTS

F1: The quantity M is an upper bound on the cost of a flow.

F2: Assumption 1 is without loss of generality and is made for notational convenience only. Alternatively, we can remove a parallel or opposite arc (v, w) by introducing a new node z and replacing (v, w) with (v, z) and (z, w) where (v, z) has same cost and capacity as (v, w) and (z, w) has 0 cost and infinite capacity.

F3: Assumption 2 is without loss of generality. If arc $(v, w) \in A$ has negative cost, it can be removed by ***saturating*** the arc: modify functions *cap*, c and b by setting $cap(w, v) = cap(v, w)$, $c(w, v) = -c(v, w)$, $b(v) = b(v) - cap(v, w)$, $b(w) = b(w) + cap(v, w)$. If the resulting problem yields a flow with value of $f'(w, v)$ on arc (w, v), then the value of $f(v, w) = cap(v, w) - f'(w, v)$.

F4: A minimum cost flow may also be defined in an undirected graph. By replacing each undirected arc by two oppositely directed arcs of the same cost and capacity, the undirected problem may be solved in the directed graph. If f' is the flow in the directed graph, the flow f in the undirected graph is obtained as follows: for all ordered pairs (v, w), $f(v, w) = \max\{0, f'(v, w) - f'(w, v)\}$.

F5: A (feasible) flow in flow network G can be found by finding a maximum s-t flow in the extended flow network G'. (See the multiple-source multiple-sink extension in subsection 11.1.5.)

F6: The minimum cost flow problem in G is equivalent to the minimum cost circulation problem in the network obtained by adding an infinite capacity arc (t, s) to G' with cost $-mC$.

F7: The following graph optimization problems are all special cases of minimum cost flows. The directed versions are described, but the undirected versions can be treated similarly.

Maximum flows: for $b_s = \beta = -b_t$ and $b_v = 0$ for $v \in V \backslash \{s, t\}$, find a flow that maximizes β. Using $b \equiv \mathbf{0}$, $c \equiv \mathbf{0}$, $A \leftarrow A \cup \{(t, s)\}$, $c_{ts} = -1$, and $cap_{ts} = \infty$, a minimum cost circulation in $G = (V, A, cap, c, b)$ is a maximum flow in the original network. (See Section 11.1.)

Single-source shortest paths: for a given vertex s, find the shortest path using arc lengths l from s to every other node. Using $b_s = |V| - 1$, $b_v = -1$, $cap \equiv \infty$, $c \equiv l$, a minimum cost flow in this graph describes a shortest path tree by taking all arcs with positive flow. (See Section 10.1.)

***k* arc-disjoint *s*-*t* paths**: given vertices s and t, arc cost function c, and an integer k, find k paths from s to t such that no two share an arc. Using $b_s = k = -b_t$, $b_v = 0$, $v \in V \backslash \{s,t\}$, $cap \equiv 1$, and c as given, the minimum cost flow yields a solution to this problem by taking all arcs with positive flow. (See Section 4.1, Menger's theorems.)

Maximum-weight bipartite matching: given a bipartite graph $G = (V_1 \cup V_2, A)$ with arc weights w, find a maximum weight subset of arcs such that no two share an end point. Using $b_v = 1$, $\forall v \in V_1$, $b_v = -1$, $\forall v \in V_2$, $c = -w$, and $cap \equiv 1$, a minimum cost flow yields a maximum weight matching by taking all arcs with positive flow. (See Section 11.3.)

Minimum cost transshipment: a minimum cost flow in a transshipment network.

Residual Networks

DEFINITIONS

D9: Given a flow f in a network G, let $F = \{e \in A | f_e < cap_e\}$ and $B = \{e \in A | f_e > 0\}$. Note that F and B may intersect. The ***residual network*** of G (with respect to) f, denoted G_f, is the network that contains for each arc $e = (v,w) \in F$ a ***forward arc*** (v,w) with ***residual capacity*** $cap_e - f_e$ and cost c_e, and for each arc $e = (v,w) \in B$ a ***backward arc*** (w,v) with residual capacity f_e and cost $-c_e$. Denote the arc-set of G_f by A_f.

D10: A set $S \subseteq A_f$ of arcs is ***augmenting*** if each arc in S has positive residual capacity. An ***augmenting path*** is a path whose set of arcs is augmenting.

D11: A flow f in a network G is augmented by u on arc set $S \subseteq A_f$ as follows: for each forward arc $(v,w) \in S$, set $f(v,w) = f(v,w) + u$, and for each backward arc $(v,w) \in S$, set $f(w,v) = f(w,v) - u$.

D12: An arc set $S \subseteq A_f$ is ***saturated*** if f is augmented on S by a quantity equal to the minimum capacity of an arc in S.

FACT

F8: Let f be a flow in G and f' be a circulation in G_f. Then $f + f'$ defined as $(f + f')(v,w) = f(v,w) + f'(v,w) - f'(w,v)$ is a flow in G.

EXAMPLE

E2: Figure 11.2.2 shows the residual network of the flow appearing in Figure 11.2.1.

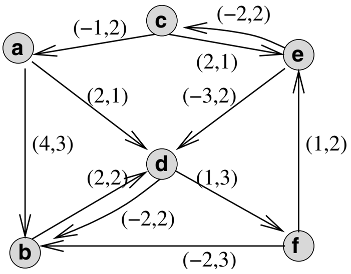


Figure 11.2.2 **The residual network of the flow in Figure 11.2.1.**

11.2.2 Optimality Conditions

DEFINITION

D13: Given a labeling $\pi : V \rightarrow Z$, the ***reduced cost*** vector c^π is defined as $c^\pi_{vw} := c_{vw} + \pi_v - \pi_w$.

FACTS

F9: If f is a minimum cost flow, then G_f has no negative cost cycles.

F10: If there exists a minimum cost flow, then there exists one such that the set of arcs in $F \cap B$ form a forest.

F11: If G_f has no negative cost cycles, then any $\pi : V \rightarrow Z$ obtained as shortest path distance labels from a selected vertex r by applying Dijkstra's algorithm (see Section 10.1) satisfies $c^\pi_{vw} \geq 0$ for all $(v, w) \in G_f$.

F12: For any flow f and labeling π, we have

(i) $c^\pi(\mathcal{C}) = c(\mathcal{C})$ for any cycle $\mathcal{C}$ in G.

(ii) $c^\pi(\mathcal{P}) = c(\mathcal{P}) + \pi_s - \pi_t$ for any s to t path $\mathcal{P}$ in G.

(iii) $\sum_{v,w} c^\pi_{vw} f_{vw} = \sum_{v,w} c_{vw} f_{vw} + \sum_v \pi_v b_v$.

Thus f minimizes $\sum_{v,w} c_{vw} f_{vw}$ if and only if f minimizes $\sum_{v,w} c^\pi_{vw} f_{vw}$.

F13: (Reduced-cost optimality conditions) If there is a labeling $\pi : V \rightarrow Z$ such that $c^\pi_{vw} \geq 0$ for all $(v, w) \in G_f$, then f is a minimum cost flow.

F14: (Complementary Slackness) If there is a labeling $\pi : V \rightarrow Z$ such that

$$\begin{aligned} c^\pi_{vw} &< 0 \;\Rightarrow f_{vw} = cap_{vw} \\ c^\pi_{vw} &> 0 \;\Rightarrow f_{vw} = 0 \\ c^\pi_{vw} &= 0 \;\Rightarrow 0 \leq f_{vw} \leq cap_{vw} \end{aligned}$$

then f is a minimum cost flow. In this case, f and π are ***complementary***.

REMARKS

R2: There is economic interpretation of the reduced-cost optimality conditions: if $c(v, w)$ is the cost of shipping one unit from v to w and π_v is the sales price of one unit at v, then the reduced cost for (v, w) is the cost associated with buying an item at v, shipping it to w, and selling it at w. If this is negative, it is worth doing, and thus the arc should be saturated. If it is positive, it is not worth doing, and thus the arc should not carry flow.

TERMINOLOGY: Consistent with the economic interpretation of Fact 13, the vertex labels π are often called ***dual prices***.

R3: Fact 14 is a restatement of Fact 13.

EXAMPLE

E3: Figure 11.2.3 highlights a negative cost cycle in the residual network appearing in Figure 11.2.2. On the left of Figure 11.2.4 is the new flow obtained by saturating this cycle, and on the right is the new residual graph with node labels in italics. The node labels and the residual network together show that the flow on the left is a minimum cost flow in the network in Figure 11.2.1.

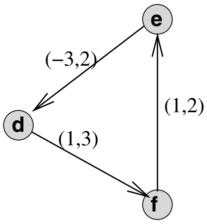


Figure 11.2.3 **A negative cost cycle.**

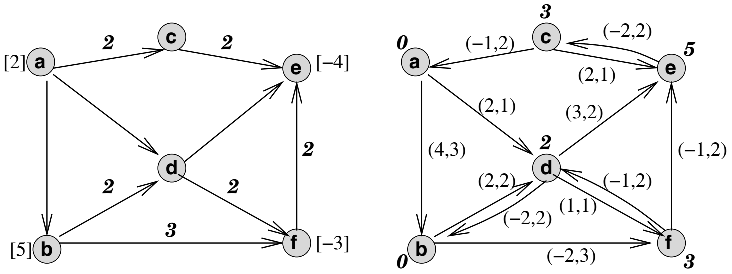


Figure 11.2.4 **A minimum cost flow, residual network, and node labels.**

A Basic Cycle-Canceling Algorithm

The correctness of the following simple algorithm for finding minimum cost flows rests on Facts 9, 11, and 13.

Algorithm 11.2.1: **Basic Cycle Canceling**

Input: a flow network G
Output: a minimum cost flow f.

```
Find a feasible flow $f$ and compute the residual network $G_f$.
While $G_f$ contains a negative cost cycle
    Find a negative cost cycle $\mathcal{C}$.
    Augment flow $f$ by saturating the arc-set of cycle $C$.
    Compute $G_f$.
```

FACTS

F15: If *cap* is integral, then the initial feasible flow is integral, and each augmentation modifies the existing flow by integral amounts. If in addition c is integral, each augmentation decreases the cost of the initial flow by at least -1.

F16: Negative cycles in G_f can be found using the algorithm of Bellman and Ford [Be58,Fo56]. (See also the references suggested in the introduction to this section.)

F17: There is a minimum cost flow in G if and only if G has no negative cost cycles with infinite capacity. In this case, the cycle-cancelling algorithm finds an integer minimum cost flow after at most M augmentations.

COMPUTATIONAL NOTE: (a) Zadeh [Za73] demonstrated by a family of examples that the basic cycle-canceling algorithm may require $O(U)$ augmentations.
(b) A modified cycle-canceling algorithm that simultaneously augments on a set of node-disjoint cycles that decrease the cost of the flow the most, finds a minimum cost flow after at most $O(m\log(nCU))$ augmentations [BaTa89].
(c) A modified algorithm that augments on a *minimum-mean cycle* (a cycle $\mathcal{C}$ that minimizes $c(\mathcal{C})/|\mathcal{C}|$) can be implemented to find a minimum cost flow after at most $O(mn\log(nC))$ augmentations [GoTa89].

REMARK

R4: Further discussion of useful ways to select cycles, and comparisons with dual algorithmic approaches that cancel cuts can be found in [IwMcSh00].

11.2.3 The Dual Problem

Frequently, problems arise that can be solved using minimum cost flow methods, but their equivalence to a minimum cost flow problem may not be immediately obvious. Certainly, recognizing such problems is important. The *linear programming dual* of minimum cost flow problem is one such type of problem.

DEFINITIONS

D14: The ***dual*** of the minimum cost flow problem (DualMCF) defined on flow network G is

$$\text{Minimize} \sum_{v,w} cap_{vw} z_{vw} \;+\; \sum_{v} \pi_v b_v$$

Subject to

$$\begin{aligned} c_{vw} + \pi_v - \pi_w + z_{vw} &\geq 0 \quad \text{for all } (v,w) \in A \\ z_{vw} &\geq 0 \quad \text{for all } (v,w) \in A \end{aligned}$$

D15: The pair of vectors z and π are ***dual feasible*** if they satisfy the constraints of DualMCF.

FACTS

F18: Given an arbitrary vector $\pi \in \mathbb{R}^n$, the pair (π, z^π) is dual feasible if $z^\pi \in \mathbb{R}^m$ is defined by $z^\pi_{vw} = \max\{0, -c^\pi_{vw}\}$.

F19: $$\min_{z,\pi \text{ dual feasible}} \left\{ \sum_{v,w} cap_{vw} z_{vw} + \sum_{v} \pi_v b_v \right\} + \min_{f \text{ a flow}} \left\{ \sum_{v,w} c_{vw} f_{vw} \right\} = 0.$$

F20: The vectors z and π are optimal for DualMCF if and only if there exists a flow $\hat{f}$ with $\sum_{v,w} cap_{vw} z_{vw} + \sum_{v,w} c^\pi_{vw} \hat{f}_{vw} = 0$.

F21: If flow $\hat{f}$ satisfying the conditions in Fact 20 exists, it can be found by fixing $\hat{f}$ on all arcs (v,w) with $c^\pi_{vw} \neq 0$ to $\hat{f}_{vw} = -\frac{cap_{vw} z_{vw}}{c^\pi_{vw}}$, $\hat{b}_v = b_v - \sum_w \hat{f}_{vw} + \sum_w \hat{f}_{wv}$, and setting $\hat{f}$ on all remaining arcs to the value of a feasible flow for $\hat{b}$ in the network $\hat{G} = (V, \hat{A})$ with $\hat{A} = \{(v,w) | c^\pi_{vw} = 0\}$.

11.2.4 Algorithms for Minimum Cost Flow

There are a wide variety of algorithms designed to solve minimum cost flow problems. Some algorithms, like *network simplex*, are based on linear programming. The cycle-canceling algorithm of subsection 11.2.2 can be interpreted as a primal algorithm: it starts with a feasible flow, and improves this solution until the complementary dual solution (in the sense of Fact 14) is feasible. In this subsection, we discuss two algorithms: a (primal-dual) augmenting-path algorithm that maintains primal and dual solutions that satisfy complementary slackness and works to make both solutions feasible; and a *push-relabel* algorithm based on the *preflow-push* maximum flow algorithm in Section 11.1. In the process we describe two powerful ideas that help algorithms achieve polynomial run-time: *capacity scaling* and *cost scaling.*

The first algorithm is simpler to describe and discuss if we assume capacities are infinite and G is a *complete* digraph (each pair of nodes has two oppositely directed arcs between them). Thus, we begin by discussing the relation between the transshipment problem and the minimum cost flow problem.

A Transshipment Problem Associated with a Minimum Cost Flow Problem

DEFINITIONS

D16: Let $G = (V, A, cap, c, b)$ be a flow network with nonnegative costs. The ***associated transshipment network*** is the network obtained by replacing each arc $e = (v, w)$ by three arcs (v, x_e), (y_e, x_e), and (y_e, w) having infinite capacity and with costs c_e, 0, and 0, repectively. Also, the supplies at the new nodes x_e and y_e are defined to be $b(x_e) = -cap_e$ and $b(y_e) = cap_e$.

D17: The ***completion*** of a transshipment network $G = (V, A, cap, c, b)$ is the complete transshipment network obtained by adding all missing arcs and giving each of them infinite capacity and arc cost $M + 1$.

FACTS

F22: Let G be a flow network and let $\hat{G}$ be its associated transshipment network. Then G has a feasible flow f if and only if $\hat{G}$ has a feasible flow $\hat{f}$ such that for each arc $e = (v, w)$, $\hat{f}_{vx_e} = fe$, $\hat{f}_{y_ex_e} = cap_e - f_e$, and $\hat{f}_{y_ew} = f_e$. Moreover, the cost of $\hat{f}$ equals the cost of f.

F23: Let G be a flow network and let G^* be the completion of its associated transshipment network. If the minimum cost transshipment uses an arc with cost $M + 1$, then the original minimum cost flow problem is not feasible.

F24: Let G be a flow network and let G^* be the completion of its associated transshipment network. Any algorithm that finds a minimum cost transshipment in G^* yields a solution to the minimum cost flow problem for G.

A Primal-Dual Algorithm

Edmonds and Karp gave the first polynomial time algorithm to find a minimum cost flow [EdKa72]. Their algorithm introduces the idea of capacity scaling. As mentioned above, the algorithm presented here is a simplified version designed for the transshipment problem on a complete transshipment network.

DEFINITION

D18: Given a flow f and a vertex v, the ***excess*** at v, denoted excess(v), is the initial supply at v minus the net flow out of v, i.e., excess$(v) = b(v) - \sum_{w \in V} [f(v,w) - f(w,v)]$.

NOTATION: For a length (or cost) function c, a flow f, and vertex labels π, dist(s,t) denotes the shortest (or least cost) augmenting path from v to w using c^{π}. By Fact 12, this is the same as the shortest (or least cost) augmenting path using c.

Algorithm 11.2.2: Capacity-Scaling Algorithm

```
Input: a complete transshipment network $G = (V, A, b, c)$
Output: a minimum cost transshipment $f$ and optimal dual prices $\pi$.
    Initialize $\pi(v) = 0$ for all $v \in V$; $f(v,w) = 0$ for all $(v,w) \in A$; $\Delta = 2^{\lfloor \log_2 B \rfloor}$.
    While $\Delta \geq 1$
        While $\max_{s \in V}\{\text{excess}(s)\} \geq \Delta$ and $\min_{t \in V}\{\text{excess}(t)\} \leq -\Delta$
            Select $s$ with excess$(s) \geq \Delta$ and $t$ with excess$(t) \leq -\Delta$.
            Compute dist$(s,v)$ for all $v \in V$.
            Augment flow by $\Delta$ on a least cost $s$-$t$ augmenting path.
            Update dual price $\pi(v) = \pi(v) + \text{dist}(s,v)$ for all $v \in V$.
        $\Delta := \Delta/2$.
```

TERMINOLOGY: A ***Δ'-phase*** is the set of algorithmic steps while $\Delta = \Delta'$.

FACTS

F25: Throughout the algorithm, f is nonnegative. After each augmentation and label update, f and π are complementary.

F26: When the algorithm terminates, f is feasible. Thus, since f and π are complementary, f is minimum cost.

F27: The number of augmentations in a Δ-phase is at most $n-1$.

COMPUTATIONAL NOTE: The capacity-scaling algorithm finds a minimum cost transshipment in $O(n(\log B)\mathcal{S}(n,m,C))$ time. Moreover, if G is a standard flow network with n vertices and m edges, then the number of vertices in the completion of its associated transshipment network (described in Definitions 16 and 17) equals $n+m$. Thus, the number of augmentations necessary to find a minimum cost flow in that complete transshipment network (which, by Fact 24, finds one in the original network G) equals $m+n-1$, which is $O(m)$. Since the complexity of a shortest-path computation is unaffected by vertices of degree 2 (added in the transformation in Definition 16), it follows that the run-time complexity of the capacity-scaling algorithm for solving a minimum cost flow problem is $O(m(\log B)\mathcal{S}(n,m,C))$.

COMPUTATIONAL NOTE: Algorithm 11.2.2 can be easily modified to work directly with a network with finite capacities. To ensure that the final f and π are complementary (so that the final f is optimal), the following condition must be maintained throughout the algorithm: $c^{\pi}(v,w) \geq 0$ if residual capacity of (v,w) is at least Δ. In the modified algorithm, the search for augmenting paths in a Δ-phase is restricted to arcs with residual capacity at least Δ. To maintain the relaxed complementary slackness, immediately after Δ is decreased, each arc with negative reduced cost and residual capacity at least $\Delta/2$ is saturated.

A Push-Relabel Algorithm

The push-relabel method was initially introduced for maximum flows (see the preflow-push algorithm in Section 11.1). We describe here a modification for minimum cost flows that runs in polynomial time. It relies on cost scaling and is due to Goldberg and Tarjan [GoTa89].

DEFINITIONS

D19: The pair (f, π) defined on $A \times V$ is ***ϵ-optimal*** if f is nonnegative and $c^\pi(v, w) \geq -\epsilon$ for all arcs with positive residual capacity.

D20: Arc (v, w) is ***admissible*** if it has positive residual capacity and $c^\pi(v, w) < 0$.

D21: Given a current flow f in a network G, the operation ***push***(v, w) assigns a flow of $\min\{cap(v, w), f(v, w) + \text{excess}(v)\}$ to arc(u, v).

D22: The operation push(v, w) is ***saturating*** if the resulting flow on arc (v, w) equals $cap(v, w)$; otherwise it is ***non-saturating***.

Algorithm 11.2.3: Push-Relabel Algorithm

Input: a flow network $G = (V, A, u, b, c)$
Output: a minimum cost flow f, and optimal dual prices π.

```
Initialize $\pi(v) = 0$ for all $v \in V$; $f(v, w) = 0$ for all $(v, w) \in A$; $\epsilon = C$.
While $\epsilon \geq \frac{1}{n}$
    Saturate each arc $(v, w)$ satisfying $c^\pi(v, w) < 0$.
    While $\max_{v \in V}\{\text{excess}(v)\} > 0$,
        Select $v$ with excess$(v) > 0$.
        If there is $w$ with $(v, w)$ admissible    {then push$(v, w)$}
            $f(v, w) := \min\{cap(v, w),\ f(v, w) + \text{excess}(v)\}$
        Else    {relabel$(v)$}
            $\pi(v) := \pi(v) - \epsilon$
    $\epsilon := \epsilon/2$
```

TERMINOLOGY: An ***ϵ'-phase*** is the set of algorithmic steps while $\epsilon = \epsilon'$.

FACTS

F28: If f is nonnegative and $\pi(v) = 0$ for all $v \in V$, then (f, π) is C-optimal. If f is a flow and (f, π) is ϵ-optimal for $\epsilon < \frac{1}{n}$, then f is a minimum cost flow.

F29: After saturation at the start of the ϵ-phase, the pair (f, π) is ϵ-optimal, and it remains ϵ-optimal throughout the phase. At the end of each ϵ-phase, f is a flow. Thus at the end of the last phase, f is a minimum cost flow.

F30: The network of admissible arcs is acyclic throughout the algorithm.

F31: π is monotone non-increasing throughout the algorithm.

F32: There are at most $3n$ relabels per vertex in an ϵ-phase.

F33: There are $O(n)$ saturating pushes per arc in an ϵ-phase.

COMPUTATIONAL NOTE: The run-time complexity of the algorithm depends on how v is selected and, for a given v, how w is selected. Goldberg and Tarjan [GoTa87], and independently Bertsekas and Eckstein [BeEc88] suggest selecting v to be the first vertex in a topological order implied by the graph of admissible arcs, and then selecting w by cycling through the adjacency list of v. They show that using such protocols, the number of non-saturating pushes is $O(n^3)$ per ϵ-phase. Using more sophisticated data structures, Goldberg and Tarjan [GoTa87] show that an implementation of this algorithm runs in $O(nm \log(n^2/m) \log(nC))$ time. Coupled with heuristic improvements, such as set relabels, *price buckets*, *flow fixing*, and *push look-ahead*, this algorithm has been shown to be experimentally competitive [Go97].

Strongly Polynomial Algorithms

There are numerous algorithms to solve the minimum cost flow problem, and detailed descriptions and comparisons already exist in [AhMaOr93,IwMcSh00,Sc03]. Both algorithms presented here have run-time complexities that depend on the size of numbers in the input data. The first *strongly polynomial time algorithm* (an algorithm with a run-time complexity that does not depend on the size of the numbers in the input data) to solve minimum cost flows is due to Tardos [Ta85] and introduces the idea of fixing flows on arcs. Most subsequent strongly polynomial time algorithms fix either arc flows or vertex labels. The minimum-mean cycle-canceling algorithm of Goldberg and Tarjan mentioned in subsection 11.2.2 can also be made to run in strongly polynomial time [GoTa89]. The fastest strongly polynomial algorithm is due to Orlin [Or88], is based on capacity scaling and arc-flow fixing, and has run-time complexity $O(m \log n \mathcal{S}(n, m, C))$.

11.2.5 Extensions to Minimum Cost Flow

Convex Cost Flows

Separable convex flow costs model some natural phenomena not captured by linear costs, for example traffic flows and matrix balancing.

DEFINITIONS

D23: Given separable convex functions $c_e : \mathbb{R} \to \mathbb{R}$ for all $e \in A$, and a flow network G, the ***minimum convex cost flow*** is a flow that minimizes $\sum_{e \in A} c_e(f_e)$ over all feasible flows.

D24: Convex costs in the ***discrete model*** are defined by piecewise linear curves, where each cost curve c_e is defined by at most p linear segments. The linear segments meet at ***breakpoints***. Convex costs in the ***continuous model*** are defined in functional form.

FACT

F34: In the discrete model, breakpoints of the cost curve can be assumed to be integers. This problem can be solved in polynomial time by transforming the convex cost flow problem into a minimum cost flow problem.

COMPUTATIONAL NOTE: In the continuous model, there is an issue of error introduced due to approximation of the values of the continuous function. This can be handled either by restricting the search for optimal flows to integral optimal flows, or by obtaining a close to optimal solution, within a tolerance dictated by the accuracy of the oracle for the continuous function. In this sense, the continous problem can be solved using techniques from nonlinear programming, or by extending various algorithms for minimum linear cost flows (e.g., capacity-scaling augmenting-path algorithm [Mi86]; minimum-mean cycle-cancelling algorithm [KaMc97]; cost-scaling algorithm [BePoTs97]).

Flows Over Time

Flows over time model problems where time plays a crucial role, such as transportation and telecommunications. They are also called *dynamic flows*.

DEFINITIONS

D25: A ***flow-over-time network*** is a flow network $G_\tau = (V, A, cap, \tau, b)$ where each arc $(v, w) \in A$ has an associated ***transit time*** τ_{vw}. The transit time τ_{vw} represents the amount of time that elapses between when flow enters arc (v, w) at v and when the same flow arrives at w.

D26: A ***flow over time*** x on G_τ with time horizon T is a collection of Lebesgue-measurable functions $x_e : [0, T) \to \mathbb{R}$ where $x_e(\theta)$ is the rate of flow (per time unit) entering arc e at time θ. For notational convenience, define $x_e(\theta) = 0$ for all θ outside the interval $[0, T)$. A flow over time satisfies the following conditions:

nonnegativity constraints: $x_{vw}(\theta) \geq 0$ for all $(v, w) \in A$ and $\theta \in [0, T]$.

flow conservation: The flow entering arc (w, v) at time θ arrives at v at time $\theta + \tau_{wv}$, i.e.,

$$\int_0^\xi \left(\sum_{w \in V} x_{vw}(\theta) \; - \; \sum_{w \in V} x_{wv}(\theta - \tau_{wv}) \right) d\theta \; = \; 0$$

for all $\xi \in [0, T)$, and $v \in V$ with $b_v = 0$.

time horizon: There is no flow after time T: $x_e(\theta) = 0$ for $\theta \in [T - \tau_e, T)$; and at time T, no flow should remain in the network, i.e.,

$$\int_0^T \left(\sum_{w \in V} x_{vw}(\theta) \; - \; \sum_{w \in V} x_{wv}(\theta - \tau_{wv}) \right) d\theta \; = \; b_v, \quad \text{for all } v \in V$$

(flow rate) capacity constraints: $x_e(\theta) \leq cap_e$, for all $\theta \in [0, T)$ and $e \in A$.

D27: In the setting with costs, the cost of a flow over time x is defined as

$$c(x) := \sum_{e \in A} \int_0^T c_e x_e(\theta) \; d\theta$$

D28: A ***maximum s-t flow over time*** with time horizon T is a flow over time for which $b_v = 0$ for all $v \in V - \{s, t\}$ and b_s is maximum among all such flows over time with time horizon T.

D29: A ***path flow*** (or ***cycle flow***) f in a flow network G is a flow on a set of arcs S that forms a path (or cycle) such that $f(e) = \nu$ on $e \in S$ and $f(e) = 0$ on $e \notin S$. In this case, the ***volume*** of the path flow (or cycle flow) is ν.

D30: Let f be a standard flow in a network $G = (V, A, cap, c, b)$, and let Γ be a set of path flows and cycle flows in G. The set Γ is a ***flow decomposition*** of the flow f if for each arc $(v, w) \in A$, and for each $\gamma \in \Gamma$, $\gamma(v, w) > 0$ only if $f(v, w) > 0$, and the flow $f(v, w) = \sum_{\gamma \in \Gamma} \gamma(v, w)$.

NOTATION: For a given flow-over-time network G_τ, the standard flow network obtained by ignoring transit times τ is denoted G.

NOTATION: For a path or cycle flow $\gamma \in \Gamma$, denote by $\nu(\gamma)$ the volume of γ, and denote by $P(\gamma)$ the path of arcs corresponding to γ. If Γ is a flow decomposition of flow in the standard flow network obtained from G_τ, let $\tau(\gamma)$ denote the sum of the transit times of arcs in $P(\gamma)$.

FACTS

F35: For any standard flow f in $G = (V, A, cap, c, b)$, there exists a flow decomposition Γ of flows on simple paths and cycles, where $|\Gamma| \leq m$.

F36: [FoFu58] Let G be an s-t flow network and let $\hat{G}$ be the augmented network obtained by adding an infinite capacity arc (t, s) with cost $c(t, s) = -T$. Suppose f is a minimum cost circulation in $\hat{G}$ and Γ is a flow decomposition of f restricted to G. Then $\tau(\gamma) \leq T$ for all path flows $\gamma \in \Gamma$. Define a flow over time as follows: for each path flow $\gamma \in \Gamma$, send flow along $P(\gamma)$ by inserting flow into the first arc in $P(\gamma)$ at rate $\nu(\gamma)$ from time 0 until time $T - \tau(\gamma)$. This flow will arrive at the end of $P(\gamma)$ in the interval $[\tau(\gamma), T)$. Ford and Fulkerson [FoFu58] showed that this flow over time is a maximum s-t flow over time with time horizon T.

F37: Unlike the case with a standard multiple-source multiple-sink, maximum flow problem (subsection 11.1.5), there is no simple transformation of a multiple-source, multiple-sink flow-over-time problem into an s-t flow-over-time problem. However, there is a combinatorial algorithm that solves this problem in polynomial time [HoTa00].

F38: Finding a minimum cost flow-over-time with time horizon T is NP-hard [KlWo95], but, for any fixed $\epsilon > 0$, an integral solution with time horizon $T(1 + \epsilon)$ and cost at most the cost of the minimum cost flow-over-time with time horizon T can be found in polynomial time (where the run time complexity depends linearly on ϵ^{-2}) [FlSk03].

Flows with Losses and Gains

Flows with losses and gains model flow problems where leakage or loss may occur. Some examples of this are financial transactions, shipping, conversion of raw materials into products, and machine loading. These flows with losses and gains are also called *generalized flows.*

DEFINITIONS

D31: A ***gain network*** $\bar{G} = (V, A, cap, \gamma, c, s, t)$ is a network $G = (V, A, cap, c, b)$ with positive-valued gain function $\gamma : E \to \mathbb{R}^+$ and supply function $b_v = 0$ for all $v \in V - \{s, t\}$. The ***gain factor*** $\gamma(e) > 0$ for arc e enforces that for each unit of flow that enters the arc, $\gamma(e)$ units exit. If $\gamma(e) < 1$, the arc e is ***lossy***; if $\gamma(e) > 1$, the arc e is ***gainy***. In general, the term gains is used to denote the gain functions of both lossy and gainy arcs. For standard network flows, the gain factor of every arc is one.

D32: A ***flow with gains*** is a function $f : A \to \mathbb{R}$ that satisfies:

capacity constraints: $f(v,w) \le cap(v,w)$ for all $(v,w) \in A$,

nonnegativity constraints: $f(v,w) \ge 0$ for all $(v,w) \in A$,

flow conservation constraints:

$$\sum_{w \in V} [f(v,w) - f(w,v)\gamma(w,v)] = 0, \quad \text{for each } v \in V - \{s,t\}.$$

D33: A ***maximum flow with gains*** is a flow with gains that maximizes the amount of flow reaching t given an unlimited supply at s.

D34: A ***minimum cost maximum flow with gains*** is a maximum flow with gains that minimizes $\sum\limits_{e \in A} c(e)g(e)$.

FACT

F39: There are combinatorial optimality conditions for flows with losses and gains that generalize the optimality conditions for standard flows described in §11.2.2 [On67, Tr77].

COMPUTATIONAL NOTE: Maximum and minimum cost flows with gains can be solved by linear programming. The versions in which f is restricted to be integral are NP-hard. Combinatorial, polynomial algorithms for maximum flow with gains are based on *high capacity* paths (paths that send the most to the sink) [GoPlTa91], or on high gain paths [GoJiOr97]. A combinatorial, polynomial time algorithm for minimum cost flow with gains is based on a generalization of the minimum-mean cycle-canceling algorithm for minimum cost flow [Wa02]. It is an interesting open question if there exists a strongly polynomial algorithm for this problem.

References

[AhMaOr93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ, 1993.

[BaTa89] F. Barahona and É. Tardos, Note on Weintraub's minimum-cost circulation algorithm, *SIAM J. Comput.* 18 no. 3 (1989), 579–583.

[Be58] R. E. Bellman, On a routing problem, *Quart. Appl. Math.* 16 (1958), 87–90.

[BeEc88] D. P. Bertsekas and J. Eckstein, Dual coordinate step methods for linear network flow problems, *Math. Programming* 42 (Ser. B) no. 2 (1988), 203–243.

[BePoTs97] D. P. Bertsekas, L. C. Polymenakos, and P. Tseng, An ϵ-relaxation method for separable convex cost network flow problems, *SIAM J. Optim.* 7, no. 3 (1997), 853–870.

[CoCuPuSc98] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, John Wiley and Sons, New York, 1998.

[EdKa70] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *J. ACM* 19 (1972), 248–264.

[FlSk03] L. Fleischer and M. Skutella, Minimum cost flows over time without intermediate storage. *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms* (2003) 66–75.

[Fo56] L. R. Ford, Jr. Network flow theory, The Rand Corp., P-923, August 1956.

[FoFu58] L. R. Ford and D. R. Fulkerson, Constructing maximal dynamic flows from static flows, *Operations Res.* 6 (1958), 419–433.

[Go97] A. V. Goldberg, An efficient implementation of a scaling minimum-cost flow algorithm, *J. Algorithms* 22, no. 1 (1997), 1–29.

[GoPlTa91] A. V. Goldberg, S. A. Plotkin, and É. Tardos, Combinatorial algorithms for the generalized circulation problem, *Math. Oper. Res.* 16, no. 2 (1991), 351–381.

[GoTa89] A. V. Goldberg and R. E. Tarjan, Finding minimum-cost circulations by canceling negative cycles, *J. Assoc. Comput. Mach.* 36, no. 4 (1989), 873–886.

[GoTa87] A. V. Goldberg and R. E. Tarjan, Finding minimum-cost circulations by successive approximation, *Math. Oper. Res.* 15, no. 3 (1990), 430–466.

[GoTaTa88] A. V. Goldberg, É. Tardos, and R. E. Tarjan, Network flow algorithms, *Paths, Flows, and VLSI-layout* (Bonn, 1988), 101–164.

[GoJiOr97] D. Goldfarb, Z. Y. Jin, and J. B. Orlin, Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem, *Math. Oper. Res.* 22, no. 4 (1997), 793–802.

[HoTa00] B. Hoppe and É. Tardos, The Quickest Transshipment Problem, *Math. Oper. Res.* 25, no. 1 (2000), 36–62.

[IwMcSh00] S. Iwata, S. T. McCormick, and M. Shigeno, Relaxed most negative cycle and most positive cut canceling algorithms for minimum cost flow, *Math. Oper. Res.* 25, no. 1 (2000), 76–104.

[KaMc97] A. V. Karzanov and S. T. McCormick, Polynomial methods for separable convex optimization in unimodular linear spaces with applications, *SIAM J. Comput.* 26, no. 4 (1997), 1245–1275.

[KlWo95] B. Klinz and G. J. Woeginger, Minimum cost dynamic flows: the series-parallel case, *Integer programming and combinatorial optimization (Copenhagen, 1995)* 329–343, Lecture Notes in Comput. Sci., 920, Springer, Berlin, 1995.

[Mi86] M. Minoux, Solving integer minimum cost flows with separable convex cost objective polynomially. Netflow at Pisa (Pisa, 1983), *Math. Programming Stud.* No. 26 (1986), 237–239.

[On67] K. Onaga, Optimum flows in general communication networks, *J. Franklin Inst.* 283 (1967) 308–327.

[Or88] J. B. Orlin, A faster strongly polynomial minimum cost flow algorithm, *Oper. Res.* 41 (1993) 338–350.

[Sc03] A. Schrijver. *Combinatorial Optimization*, Springer-Verlag, Berlin, 2003.

[Ta85] É. Tardos, A strongly polynomial minimum cost circulation algorithm, *Combinatorica* 5, no. 3 (1985), 247–255.

[Tr77] K. Truemper, On max flows with gains and pure min-cost flows, *SIAM J. Appl. Math.* 32, no. 2 (1977), 450–456.

[Wa02] K. D. Wayne, A polynomial combinatorial algorithm for generalized minimum cost flow, *Math. Oper. Res.* 27, no. 3 (2002), 445–459.

[Za73] N. Zadeh, A bad network problem for the simplex method and other minimum cost flow algorithms, *Math. Program.* 5 (1973), 255–266.

11.3 MATCHINGS AND ASSIGNMENTS

Douglas R. Shier, Clemson University

11.3.1 Matchings
11.3.2 Matchings in Bipartite Graphs
11.3.3 Matchings in Nonbipartite Graphs
References

Introduction

In an undirected graph, the maximum matching problem requires finding a set of nonadjacent edges having the largest total size or largest total weight. This graph optimization problem arises in a number of applications, often involving the optimal pairing of a set of objects.

11.3.1 Matchings

Matchings are defined on undirected graphs, in which the edges can be weighted. Matchings are useful in a wide variety of applications, such as vehicle and crew scheduling, sensor location, snowplowing streets, scheduling on parallel machines, among others.

DEFINITIONS

D1: Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. Each edge $e \in E$ has an associated ***weight*** w_e.

D2: A ***matching*** in $G = (V, E)$ is a set $M \subseteq E$ of pairwise nonadjacent edges.

D3: A ***vertex cover*** in G is a set C of vertices such that every edge in G is incident on at least one vertex in C.

D4: A ***perfect matching*** in $G = (V, E)$ is a matching M in which each vertex of V is incident on exactly one edge of M.

TERMINOLOGY: A perfect matching of G is also called a ***1-factor*** of G; see §5.3.

D5: The ***size*** (***cardinality***) of a matching M is the number of edges in M, written $|M|$. The ***weight*** of a matching M is $wt(M) = \sum_{e \in M} w_e$.

D6: A ***maximum-size matching*** of G is a matching M having the largest size $|M|$.

D7: A ***maximum-weight matching*** of G is a matching M having the largest weight $wt(M)$.

D8: Relative to a matching M in $G = (V, E)$, edges $e \in M$ are ***matched*** edges, while edges $e \in E - M$ are ***free*** edges. Vertex v is ***matched*** if it is incident on a matched edge; otherwise vertex v is ***free*** (or ***unmatched***).

D9: Every matched vertex v has a ***mate***, the other endpoint of the matched edge incident on v.

D10: With respect to a matching M, the ***weight*** $wt(P)$ of path P is the sum of the weights of the free edges in P minus the sum of the weights of the matched edges in P.

D11: An ***alternating*** path has edges that are alternately free and matched. An ***augmenting*** path is an alternating path that starts at a free vertex and ends at another free vertex.

NOTATION: Throughout this section, edges are represented as ordered pairs of vertices, and when discussing matchings, paths are represented as edge sets.

EXAMPLES

E1: Figure 11.3.1 shows a graph G together with the matching $M_1 = \{(2,3),(4,5)\}$ of size 2; the matched edges are highlighted. The mate of vertex 2 is vertex 3, and the mate of vertex 5 is vertex 4. Relative to matching M_1, vertices 1 and 6 are free vertices, and an augmenting path P from 1 to 6 is given by $P = \{(1,2),(2,3),(3,4),(4,5),(5,6)\}$. The matching M_1 is not a maximum-size matching; the matching $M_2 = \{(1,2),(3,4),(5,6)\}$ of size 3 is a maximum-size matching, in fact a perfect matching.

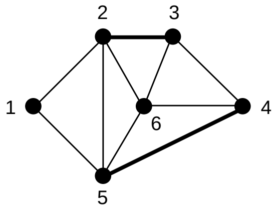


Figure 11.3.1 A matching in a graph.

E2: Figure 11.3.2 shows a graph G with 6 vertices. The matching $M = \{(2,4),(3,5)\}$ displayed is a maximum-size matching, of size 2. This graph G does not have a perfect matching.

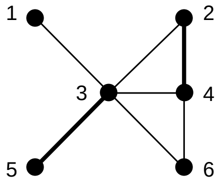


Figure 11.3.2 A maximum-size matching that is not perfect.

E3: In the weighted graph G of Figure 11.3.3 below, the weight w_e is shown next to each edge e. The weight of matching $M = \{(1,2),(3,5)\}$ is $wt(M) = 7$. Relative to this matching, the path $P = \{(1,2),(2,5),(3,5),(3,6)\}$ is an alternating path with $wt(P) = 7 + 1 - 2 - 5 = 1$. The path $\{(1,4),(1,2),(2,3),(3,5),(5,6)\}$ is an augmenting path, joining the free vertices 4 and 6.

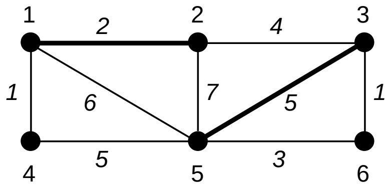


Figure 11.3.3 A matching in a weighted graph.

Some Fundamental Results

FACTS

F1: If M is a matching of $G = (V, E)$, then the number of matched vertices is $2|M|$ and the number of free vertices is $|V| - 2|M|$.

F2: If M is any matching in G, then $|M| \leq \left\lfloor \frac{|V|}{2} \right\rfloor$.

F3: (Weak Duality) The size of any vertex cover of G is an upper bound on the size of any matching in G.

F4: Every augmenting path has an odd number of edges.

F5: If M is a matching and P is an augmenting path with respect to M, then the symmetric difference $M \Delta P$ is a matching of size $|M| + 1$.

NOTE: The symmetric difference $M \Delta P$ is taken with respect to the *edge sets* defining M and P.

F6: (Augmenting Path Theorem) M is a maximum-size matching if and only if there is no augmenting path with respect to M. (See [Pe1891], [Be57], [NoRa59].)

F7: If M is a matching and P is an augmenting path with respect to M, then $wt(M \Delta P) = wt(M) + wt(P)$.

F8: Suppose M is a matching having maximum weight among all matchings of a fixed size k. If P is an augmenting path of maximum weight with respect to M, then $M \Delta P$ is a maximum-weight matching among all matchings of size $k + 1$.

F9: Let M_i be a maximum-weight matching among all matchings of a fixed size i, $i = 1, 2 \ldots, k$, and let P_i be a maximum-weight augmenting path with respect to M_i. Then $wt(P_1) \geq wt(P_2) \geq \cdots \geq wt(P_k)$.

F10: An immediate consequence of Facts 8 and 9 is a weighted-matching analogue to the Augmenting Path Theorem: A matching M is of maximum weight if and only if the weight of every augmenting path relative to M is nonpositive.

EXAMPLES

E4: In Figure 11.3.1, the path $P = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 6)\}$ is augmenting with respect to the matching $M_1 = \{(2, 3), (4, 5)\}$. As guaranteed by Fact 4, path P has an odd number of edges. The new matching $M_2 = M_1 \Delta P = \{(1, 2), (3, 4), (5, 6)\}$ has size one greater than M_1, and is a maximum-size matching. There are other maximum-size matchings, such as $\{(1, 2), (3, 6), (4, 5)\}$ and $\{(1, 5), (2, 6), (3, 4)\}$.

E5: In Figure 11.3.2, the set $S = \{3, 4\}$ is a vertex cover of G. Thus, by Fact 3, the size of any matching M satisfies $|M| \leq 2 = |S|$. On the other hand, $S = \{2, 3, 4, 5\}$ is a (minimum cardinality) vertex cover of the graph in Figure 11.3.1, yet a maximum-size matching M for this graph satisfies $|M| = 3 < |S|$.

E6: Figure 11.3.4(a) below shows a matching M_1 of size 1, with $wt(M_1) = 7$. Since edge $(2, 5)$ has maximum weight among all edges, M_1 is a maximum-weight matching of size 1. Relative to M_1, the augmenting path $P_1 = \{(1, 5), (2, 5), (2, 3)\}$ has weight $wt(P_1) = 6 + 4 - 7 = 3$, whereas the augmenting path $P_2 = \{(3, 6)\}$ has weight 1. It can be verified that P_1 is a maximum-weight augmenting path relative to M_1. Illustrating

Fact 8, $M_2 = M_1 \Delta P_1 = \{(1,5),(2,3)\}$ is a maximum-weight matching of size 2, with $wt(M_2) = wt(M_1) + wt(P_1) = 10$ (see Figure 11.3.4(b)). Relative to M_2 there are several augmenting paths between the free vertices 4 and 6:

$$\begin{aligned} Q_1 &= \{(1,4),(1,5),(5,6)\}, & wt(Q_1) &= 1+3-6 = -2,\\ Q_2 &= \{(1,4),(1,5),(2,5),(2,3),(3,6)\}, & wt(Q_2) &= 1+7+1-6-4 = -1,\\ Q_3 &= \{(4,5),(1,5),(1,2),(2,3),(3,6)\}, & wt(Q_3) &= 5+2+1-6-4 = -2. \end{aligned}$$

Path Q_2 is a maximum-weight augmenting path and so (by Fact 8) $M_3 = M_2 \Delta Q_2 = \{(1,4),(2,5),(3,6)\}$ is a maximum-weight matching of size 3, with $wt(M_3) = 9$. All augmenting paths relative to M_2 have negative weight, and so (by Fact 10) M_2 is a maximum-weight matching in G.

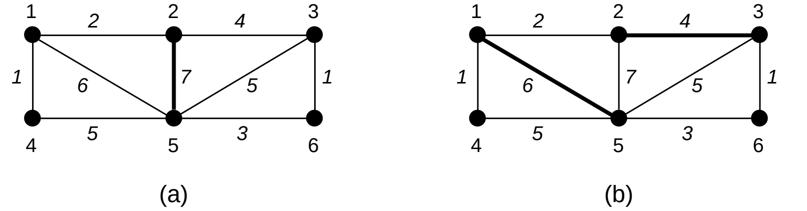


Figure 11.3.4 Maximum-weight matchings of sizes 1 and 2.

REMARKS

R1: Fact 6 was obtained independently by C. Berge [Be57] and by R. Z. Norman and M. O. Rabin [NoRa59]. This result was also recognized in an 1891 paper of J. Petersen [Pe1891].

R2: An historical perspective on the theory of matchings is provided in [Pl92].

R3: Plummer [Pl93] describes a number of variations on the standard matching problem, together with their computational complexity.

11.3.2 Matchings in Bipartite Graphs

Bipartite graphs arise in a number of applications (such as in assigning personnel to jobs or tracking objects over time). See the surveys [AhMaOr93], [AhMaOrRe95], and [Ge95] as well as the text [LoPl86] for additional applications. This section describes properties and algorithms for maximum-size and maximum-weight matchings in bipartite graphs.

DEFINITIONS

D12: Let $G = (X \cup Y, E)$ be a bipartite graph with n vertices, m edges, and edge weights w_e.

D13: If $S \subseteq X$ then $\Gamma(S) = \{y \in Y \mid (x,y) \in E \text{ for some } x \in S\}$ is the set of vertices in Y adjacent to some vertex of S.

D14: A ***complete*** (or ***X-saturating***) ***matching*** of $G = (X \cup Y, E)$ is a matching M in which each vertex of X is incident on an edge of M. Such a matching is also called an ***assignment*** from X to Y.

APPLICATIONS

A1: A drug company is testing n antibiotics on n volunteer patients in a hospital. Some patients have known allergic reactions to certain of these antibiotics. To determine if there is a feasible assignment of the n different antibiotics to n different patients, construct the bipartite graph $G = (X \cup Y, E)$, where X is the set of antibiotics and Y is the set of patients. An edge $(i, j) \in E$ exists when patient j is *not* allergic to antibiotic i. A complete matching of G is then sought.

A2: There are n applicants to be assigned to n jobs, with each job being filled with exactly one applicant. The weight w_{ij} measures the suitability (or productivity) of applicant i for job j. Finding a valid assignment (matching) achieving the best overall suitability is a weighted matching problem on the bipartite graph $G = (X \cup Y, E)$, where X is the set of applicants and Y is the set of jobs.

A3: The movements of n objects (such as submarines or missiles) are to be followed over time. The locations of the group of objects are known at two distinct times, though without identification of the individual objects. Suppose $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ represent the spatial coordinates of the objects detected at times t and $t + \Delta t$. If Δt is sufficiently small, then the Euclidean distance between a given object's position at these two times should be relatively small. To aid in identifying the objects (as well as their velocities and directions of travel), a pairing between set X and set Y is desired that minimizes the overall sum of Euclidean distances. This can be formulated as a maximum-weight matching problem on the complete bipartite graph $G = (X \cup Y, E)$, where edge (i, j) indicates pairing position x_i with position y_j. The weight of this edge is the negative of the Euclidean distance between x_i and y_j. A maximum-weight matching of size n in G then provides an optimal (minimum distance) pairing of observations at the two times t and $t + \Delta t$.

FACTS

F11: (König's Theorem) For a bipartite graph G, the maximum size of a matching in G is the minimum cardinality of a vertex cover in G. Thus for bipartite graphs the general inequality stated in Fact 3 can always be satisfied as an equality. (See [Bo90].)

F12: (Hall's Theorem) $G = (X \cup Y, E)$ has a complete matching if and only if $|\Gamma(S)| \geq |S|$ holds for every $S \subseteq X$. In words, a complete matching exists precisely when every set of vertices in X is adjacent to at least an equal number of vertices in Y. (See [Bo90, Gr99].)

F13: Suppose there exists some k such that $deg(x) \geq k \geq deg(y)$ holds in $G = (X \cup Y, E)$ for all $x \in X$ and $y \in Y$. Then G has a complete matching. (See [Gr99].)

EXAMPLES

E7: In the bipartite graph of Figure 11.3.5 below, $S = \{1, 3, b, d\}$ is a vertex cover of minimum cardinality, and $M = \{(1, a), (3, c), (4, b), (5, d)\}$ is a maximum-size matching. As guaranteed by König's Theorem, $|M| = |S|$. Also, by choosing $A = \{2, 4, 5\}$ we have $\Gamma(A) = \{b, d\}$. Since $|\Gamma(A)| < |A|$ holds, Hall's Theorem shows that there is no complete matching with respect to the set $X = \{1, 2, 3, 4, 5\}$. In fact, the maximum matching M above has size $4 < 5$.

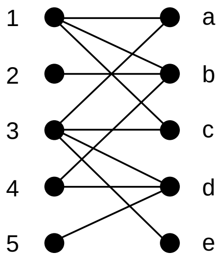


Figure 11.3.5 **Covers and matchings in a bipartite graph.**

E8: In the chessboard of Figure 11.3.6, we are to place non-taking rooks at certain allowable positions, those marked with an X. For example, we can place rooks at the *independent* positions $(1, 3), (2, 4), (4, 1)$: no two selected positions are in the same row or column. It turns out that three is the maximum number of rooks that can be so placed with regard to the allowable positions X. Also, notice that row 2, row 4, and column 3 are three *lines* in the chessboard containing all X entries; in fact, no fewer number of lines suffice. Here the maximum number of non-taking rooks among the X entries equals the minimum number of lines containing all the X entries. This is a manifestation of König's Theorem, obtained by constructing the bipartite graph $G = (X \cup Y, E)$ where X contains the rows $\{1, 2, 3, 4\}$ and Y contains the columns $\{1, 2, 3, 4, 5\}$; edge $(i, j) \in E$ indicates an X in row i and column j. In this context, independent positions correspond to a matching and covering lines correspond to a vertex cover in G.

	1	2	3	4	5
1			☒		
2		☒		☒	☒
3			☒		
4	☒		☒	☒	

Figure 11.3.6 **A chessboard with allowable X entries.**

REMARKS

R4: König's Theorem and Hall's Theorem can be derived from the Max-Flow Min-Cut Theorem of §11.1.

R5: Maximum-size matching problems in bipartite graphs can be formulated as maximum flow problems in unit capacity networks and solved using maximum flow algorithms (§11.1).

R6: Maximum-weight matching problems in bipartite graphs can be formulated as minimum cost flow problems in two-terminal flow networks and solved using minimum cost flow algorithms (§11.2).

Bipartite Maximum-Size Matching Algorithm

Algorithm 11.3.1, based on Fact 6, produces a maximum-size matching of the bipartite graph $G = (X \cup Y, E)$. Each iteration involves a modified breadth-first search of G, starting with all free vertices in set X. The vertices of G are structured into levels that alternate between free and matched edges. Algorithm algnum can be implemented to run in $O(nm)$ time (see [PaSt82]).

Algorithm 11.3.1: Bipartite Maximum-Size Matching

Input: Bipartite graph $G = (X \cup Y, E)$.
Output: Maximum-size matching M.

```
$M := \emptyset$
$DONE :=$ FALSE
While NOT $DONE$
    Let $FREE$ consist of all the free vertices of $G$.
    $S_X := X \cap FREE$
    $SEEN := \emptyset$
    $STILL\_LOOKING :=$ TRUE
    While $STILL\_LOOKING$ {for an augmenting path}
        $S_Y := \{y \mid y \notin SEEN$ and $(x,y) \in E,\ x \in S_X\}$
        If $S_Y \cap FREE \neq \emptyset$ {an augmenting path exists}
            Construct an augmenting path $P$ to $y^*$.   [*]
            $M := M \Delta P$
            $STILL\_LOOKING :=$ FALSE
        Else {continue looking for an augmenting path}
            $SEEN := SEEN \cup S_Y$
            $S_X := \{x \mid (y,x) \in M,\ y \in S_Y\}$
            If $S_X = \emptyset$
                $STILL\_LOOKING :=$ FALSE
                $DONE :=$ TRUE
```

REMARK

R7: The augmenting path at step [*] is constructed in reverse, starting at the free Y-vertex y^*. Choose a vertex $x \in S_X$ (adjacent to y^*) by which y^* was defined to be an element of S_Y. Then choose the vertex $y \in S_Y$ that is matched to x in M. Vertices from X and Y are alternately chosen in this way until an x is chosen from the initial S_X, which means that it is a free vertex.

EXAMPLE

E9: Algorithm 11.3.1 can be used to find a maximum-size matching in the bipartite graph of Figure 11.3.7 below. We begin with the matching $M = \{(1,a),(2,b)\}$ of size 2, shown in Figure 11.3.7(a). At the next iteration, $S_X = \{3,4\}$ and $S_Y = \{a,b\}$. Since both vertices of S_Y are matched, the algorithm continues with $S_X = \{1,2\}$ and $S_Y = \{c\}$. Since $c \in S_Y$ is free, with augmenting path $P = \{(3,a),(a,1),(1,c)\}$, the new matching produced is $M = \{(1,c),(2,b),(3,a)\}$; see Figure 11.3.7(b). The next iteration produces $S_X = \{4\}$, $S_Y = \{b\}$; $S_X = \{2\}$, $S_Y = \{a,c\}$; finally $S_X = \{1,3\}$, $S_Y = \emptyset$, $S_X = \emptyset$. No further augmenting paths are found, and Algorithm 11.3.1 terminates with the maximum-size matching $M = \{(1,c),(2,b),(3,a)\}$.

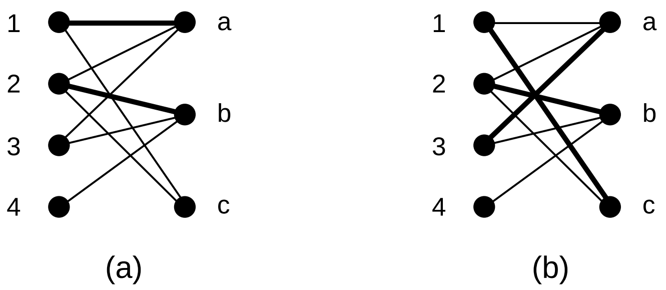


Figure 11.3.7 Maximum-size matching in a bipartite graph.

Bipartite Maximum-Weight Matching Algorithm

Algorithm 11.3.2, based on Facts 8, 9, and 10, produces a maximum-weight matching of $G = (X \cup Y, E)$. Each iteration finds a maximum-weight augmenting path relative to the current matching M. The algorithm terminates when the path has nonpositive weight. A straightforward implementation of Algorithm 11.3.2 runs in $O(n^2m)$ time.

NOTATION: The tentative largest weight of an alternating path from a free vertex in X to vertex j is maintained using the label $d(j)$.

Algorithm 11.3.2: Bipartite Maximum-Weight Matching

Input: Bipartite graph $G = (X \cup Y, E)$.
Output: Maximum-weight matching M.

```
$M := \emptyset$
$DONE :=$ FALSE
While NOT $DONE$
    Let $S_X$ consist of all the free vertices of $X$.
    Let $d(j) := 0$ for $j \in S_X$ and $d(j) := -\infty$ otherwise.
    While $S_X \neq \emptyset$
        $S_Y := \emptyset$
        For each edge $(x, y) \in E - M$ with $x \in S_X$
            If $d(x) + w_{xy} > d(y)$
                $d(y) := d(x) + w_{xy}$
                $S_Y := S_Y \cup \{y\}$
        $S_X := \emptyset$
        For each edge $(y, x) \in M$ with $y \in S_Y$
            If $d(y) - w_{yx} > d(x)$
                $d(x) := d(y) - w_{yx}$
                $S_X := S_X \cup \{x\}$
    Let $y$ be a free vertex with maximum label $d(y)$
     and let $P$ be the associated path.
    If $d(y) > 0$
        $M := M \Delta P$
    Else
        $DONE :=$ TRUE
```

EXAMPLE

E10: Algorithm 11.3.2 can be used to find a maximum-weight matching in the bipartite graph of Figure 11.3.8. If we begin with the empty matching, then the first iteration yields the augmenting path $P_1 = \{(3, a)\}$, with $wt(P_1) = 6$, and the maximum-weight matching (of size 1) $M = \{(3, a)\}$, with $wt(M) = 6$; see Figure 11.3.8(a). The next iteration starts with $S_X = \{1, 2\}$. The labels on vertices a, b, c are then updated to $d(a) = 4$, $d(b) = 4$, $d(c) = 5$, so $S_Y = \{a, b, c\}$. Using the matched edge $(a, 3)$, vertex 3 has its label updated to $d(3) = -2$ and $S_X = \{3\}$. No further updates occur, and the free vertex c with maximum label $d(c) = 5$ is selected. This label corresponds to the augmenting path $P_2 = \{(2, c)\}$, with $wt(P_2) = 5$. The new matching is $M = \{(2, c), (3, a)\}$, with $wt(M) = 11$; see Figure 11.3.8(b). At the next iteration, $S_X = \{1\}$ and vertices a, b receive updated labels $d(a) = 4$, $d(b) = 1$. Subsequent updates produce $d(3) = -2$, $d(c) = 3$, $d(2) = -2$, $d(b) = 2$. Finally, the free vertex b is selected with $d(b) = 2$, corresponding to the augmenting path $P_3 = \{(1, a), (a, 3), (3, c), (c, 2), (2, b)\}$, with $wt(P_3) = 2$. This gives the maximum-weight matching $M = \{(1, a), (2, b), (3, c)\}$, with $wt(M) = 13$, shown in Figure 11.3.8(c). As predicted by Fact 9, the weights of the augmenting paths are nonincreasing: $wt(P_1) \geq wt(P_2) \geq wt(P_3)$.

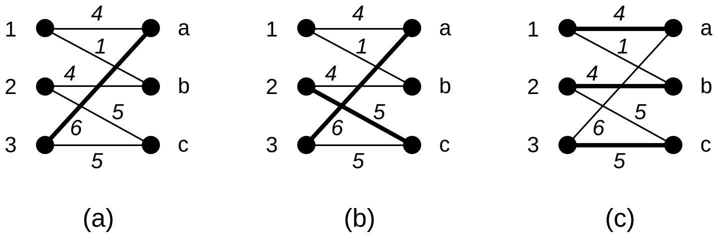


Figure 11.3.8 Maximum-weight matchings of sizes 1, 2, and 3.

11.3.3 Matchings in Nonbipartite Graphs

This section discusses matchings in more general (nonbipartite) graphs. Algorithms for constructing maximum-size and maximum-weight matchings are considerably more intricate than for bipartite graphs. The important new concept is that of a “blossom”.

DEFINITIONS

D15: Suppose P is an alternating path from a free vertex s in graph $G = (V, E)$. Then a vertex v on P is ***even*** if the subpath P_{sv} of P joining s to v has even length; it is ***odd*** if P_{sv} has odd length.

D16: Suppose P is an alternating path from a free vertex s to an even vertex v and edge $(v, w) \in E$ joins v to another even vertex w on P. Then $P \cup \{(v, w)\}$ contains a unique cycle, called a ***blossom***.

D17: A ***shrunken blossom*** results when a blossom B is collapsed into a single vertex b, whereby any edge (x, y) with $x \notin B$ and $y \in B$ is transformed into the edge (x, b). The reverse of this process gives an ***expanded blossom***.

FACTS

F14: A blossom B has odd length $2k+1$ and contains k matched edges, for some $k \geq 1$.

F15: A bipartite graph contains no blossoms.

F16: (Edmonds's Theorem) [Ed65a] Suppose graph G^B is formed from G by collapsing blossom B. Then G^B contains an augmenting path if and only if G does.

F17: (General Maximum-Size Matching) Algorithm 11.3.3, based on Fact 6, produces a maximum-size matching of G. At each iteration, a forest of trees is grown, rooted at the free vertices of G, in order to identify an augmenting path. As encountered, blossoms B are shrunk, with the search continued in the resulting graph G^B.

Algorithm 11.3.3: General Maximum-Size Matching

Input: Graph $G = (X \cup Y, E)$.
Output: Maximum-size matching M.

```
$M := \emptyset$
$DONE :=$ FALSE
While NOT $DONE$
    Mark all free vertices as even.
    Mark all matched vertices as unreached.
    Mark all free edges as unexamined.
    While there are unexamined edges and no augmenting path is found
        Let $(v, w)$ be an unexamined edge.
        Mark $(v, w)$ as examined.
      {Case 1}
        If $v$ is even and $w$ is unreached
            Mark $w$ as odd and its mate $z$ as even.
            Extend the forest by adding $(v, w)$ and matched edge $(w, z)$.
      {Case 2}
        If $v$ and $w$ are even and they belong to different subtrees
            An augmenting path has been found.
      {Case 3}
        If $v$ and $w$ are even and they belong to the same subtree
            A blossom $B$ is found.
            Shrink $B$ to an even vertex $b$.
    If an augmenting path $P$ has been found
        $M := M \Delta P$
    Else
        $DONE :=$ TRUE
```

F18: Algorithm 11.3.3 was initially proposed by Edmonds [Ed65a] with a time bound of $O(n^4)$. An improved implementation of Algorithm 11.3.3 runs in $O(nm)$ time; see [Ta83] and [Ge95].

F19: Maximum-size matchings in nonbipartite graphs can also be found using the algorithm of Gabow [Ga76], that runs in $O(n^3)$ time, and the algorithm of Micali and Vazirani [MiVa80], that runs in $O(m\sqrt{n})$ time.

F20: More complicated algorithms are required for solving weighted-matching problems in general graphs. The first such algorithm, also involving blossoms, was developed by Edmonds [Ed65b] and has a time bound of $O(n^4)$.

F21: Improved algorithms exist for the weighted-matching problem, with running times $O(n^3)$ and $O(nm \log n)$ respectively; see [AhMaOr93] and [Ge95]. An $O(nm + n^2 \log n)$ algorithm is given by Gabow [Ga90].

EXAMPLES

E11: In Figure 11.3.9(a), $P = \{(1,2),(2,3),(3,4),(4,5)\}$ is an alternating but not augmenting path, with respect to the matching $M = \{(2,3),(4,5)\}$. Relative to path P, vertices $1,3,5$ are even while vertices $2,4$ are odd. Since $(5,3)$ is an edge joining two even vertices on P, the blossom $B = \{(3,4),(4,5),(5,3)\}$ is formed. On the other hand, $Q = \{(1,2),(2,3),(3,5),(5,4),(4,6)\}$ is an augmenting path relative to M so that $M \Delta P = \{(1,2),(3,5),(4,6)\}$ is a matching of larger size—in fact a matching of maximum size. Notice that relative to path Q, vertices $1,3,4$ are even while vertices $2,5,6$ are odd.

E12: Shrinking the blossom B relative to path P in Figure 11.3.9(a) produces the graph G^B shown in Figure 11.3.9(b). The path $P^B = \{(1,2),(2,b),(b,6)\}$ is now augmenting in G^B. By expanding P^B so that $(2,3)$ remains matched and $(4,6)$ remains free, the augmenting path $Q = \{(1,2),(2,3),(3,5),(5,4),(4,6)\}$ in G is obtained.

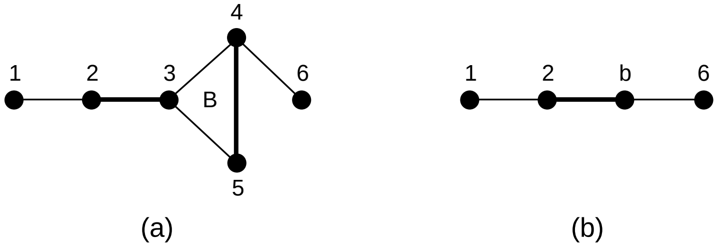


Figure 11.3.9 **Shrinking blossom B.**

E13: Algorithm 11.3.3 can be applied to the nonbipartite graph shown in Figure 11.3.10(a). Suppose the matching $M = \{(3,4),(6,8)\}$ of size 2 is already available.

Iteration 1: The free vertices $1,2,5,7$ are marked as even, and the matched vertices $3,4,6,8$ are marked as unreached. The initial forest consists of the isolated vertices $1,2,5,7$.

• If the free edge $(2,3)$ is examined then Case 1 applies, so vertex 3 is marked odd and vertex 4 even; the free edge $(2,3)$ and the matched edge $(3,4)$ are added to the forest.

• If the free edge $(7,4)$ is next examined then Case 2 applies, and the augmenting path $P = \{(2,3),(3,4),(4,7)\}$ is found. Using P the new matching $M = \{(2,3),(4,7),(6,8)\}$ of size 3 is obtained; see Figure 11.3.10(b).

Iteration 2: The forest is initialized with the free (even) vertices $1,5$.

• If the free edge $(1,2)$ is examined then Case 1 applies, so vertex 2 is marked odd and vertex 3 even; edges $(1,2)$ and $(2,3)$ are added to the forest.

• Examining in turn the free edges $(3,4)$ and $(7,6)$ makes $4,6$ odd vertices and $7,8$ even. Edges $(3,4),(4,7),(7,6),(6,8)$ are then added to the subtree rooted at 1.

- If edge $(8,7)$ is examined, then Case 3 applies, and the blossom $B = \{(7,6),(6,8),(8,7)\}$ is detected and shrunk; Figure 11.3.10(c) shows the resulting G^B. The current subtree rooted at 1 now becomes $\{(1,2),(2,3),(3,4),(4,b)\}$.

- If the free edge $(b,5)$ is examined, then Case 2 applies and the augmenting path $\{(1,2),(2,3),(3,4),(4,b),(b,5)\}$ is found in G^B. The corresponding augmenting path in G is $P = \{(1,2),(2,3),(3,4),(4,7),(7,8),(8,6),(6,5)\}$. Forming $M \Delta P$ produces the new matching $\{(1,2),(3,4),(5,6),(7,8)\}$, which is a maximum-size matching; see Figure 11.3.10(d).

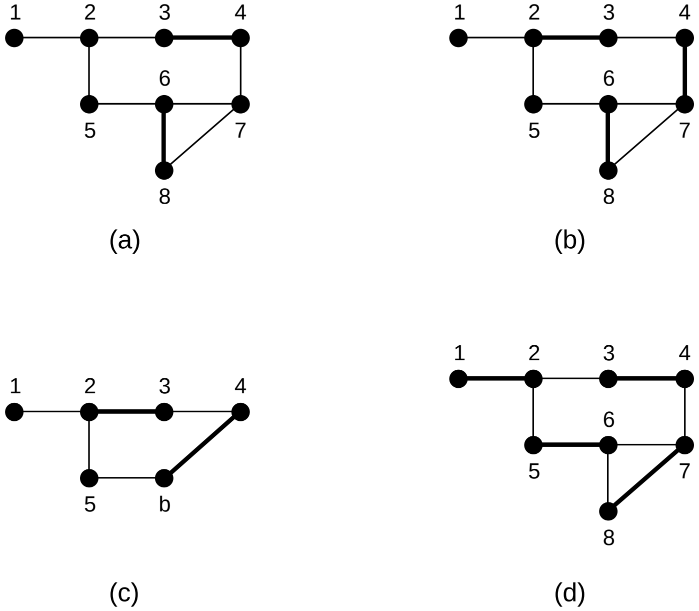


Figure 11.3.10 Illustrating Algorithm 11.3.3.

APPLICATIONS

A4: Pairs of pilots are to be assigned to aircraft serving international routes. Pilots i and j are considered compatible if they are fluent in a common language and have comparable flight training. Form the graph G whose vertices represent pilots and whose edges represent compatible pairs of pilots. The problem of flying the largest number of aircraft with compatible pilots is then a maximum-size matching problem on G.

A5: Bus drivers are hired to work two four-hour shifts each day. Union rules require a certain minimum amount of time between the shifts that a driver can work. There are also costs associated with transporting the driver between the ending location of the first shift and the starting location of the second shift. The problem of optimally combining pairs of shifts that satisfy union regulations and incur minimum total cost can be formulated as a maximum-weight matching problem. Namely, define the graph G with vertices representing each shift that must be covered and edges between pairs of compatible shifts (satisfying union regulations). The weight of edge (i,j) is the negative of the cost of assigning a single driver to shifts i and j. It is convenient also to add edges (i,i) to G to represent the possibility of needing a part-time driver to cover a single shift; edge (i,i) is given a sufficiently large negative weight to discourage single-shift assignments unless absolutely necessary. A maximum-weight perfect matching in G then provides a minimum-cost pairing of shifts for the bus drivers.

References

[AhMaOr93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, 1993.

[AhMaOrRe95] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and M. R. Reddy, Applications of network optimization, pages 1–83 in M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (Eds.), *Network Models*, North-Holland, 1995.

[Be57] C. Berge, Two theorems in graph theory, *Proc. Natl. Acad. Sci.* (U.S.A.) 43 (1957), 842–844.

[Bo90] K. P. Bogart, *Introductory Combinatorics*, Harcourt Brace Jovanovich, 1990.

[Ed65a] J. Edmonds, Paths, trees, and flowers, *Canadian Journal of Mathematics* 17 (1965), 449–467.

[Ed65b] J. Edmonds, Maximum matching and a polyhedron with $0,1$-vertices, *Journal of Research of the National Bureau of Standards* B-69 (1965), 125–130.

[Ga76] H. N. Gabow, An efficient implementation of Edmonds' algorithm for maximum matching on graphs, *Journal of the ACM* 23 (1976), 221–234.

[Ga90] H. N. Gabow, Data structures for weighted matchings and nearest common ancestors with linking, *Proceedings of the 1st Annual ACM-SIAM Symposium on Discrete Algorithms*, 1990, 434–443.

[Ge95] A. M. H. Gerards, Matching, pages 135–224 in M. Ball, T. Magnanti, C. Monma, and G. Nemhauser (Eds.), *Network Models*, North-Holland, 1995.

[Gr99] R. P. Grimaldi, *Discrete and Combinatorial Mathematics*, Addison Wesley-Longman, 1990.

[LoPl86] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland, 1986.

[MiVa80] S. Micali and V. V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, 1980, 17–27.

[NoRa59] R. Z. Norman and M. O. Rabin, An algorithm for a minimum cover of a graph, *Proceedings of the American Mathematics Society* 10 (1959), 315–319.

[PaSt82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice-Hall, 1982.

[Pe1891] J. Petersen, Die Theorie der regulären graphs, *Acta Mathematica* 15 (1891), 193–220.

[Pl92] M. D. Plummer, Matching theory—a sampler: from Dénes König to the present, *Discrete Mathematics* 100 (1992), 177–219.

[Pl93] M. D. Plummer, Matching and vertex packing: how "hard" are they?, pages 275–312 in J. Gimbel, J. W. Kennedy, and L. V. Quintas (Eds.), *Quo Vadis, Graph Theory?*, *Annals of Discrete Mathematics* 55, North-Holland, 1993.

[Ta83] R. E. Tarjan, *Data Structures and Network Algorithms*, SIAM, 1983.

11.4 COMMUNICATION NETWORK DESIGN MODELS

Prakash Mirchandani, University of Pittsburgh
David Simchi-Levi, Massachusetts Institute of Technology

11.4.1 General Network Design Model
11.4.2 Uncapacitated Network Design
11.4.3 Survivable Network Design
11.4.4 Capacitated Network Design
References

Introduction

Telecommunication network design represents a major application of graph theory. Although network design problems come in many different flavors that correspond to different telecommunication application contexts, the common theme across all such problems is that we need to connect a set of locations (for example, customers, computers, cities, or communication switches) using transmission links (which may or may not be capacitated) in order to satisfy the demand (consisting of voice, video, or data traffic) between pairs of locations at minimum cost. While simple to state, network design problems are challenging from the modeling, algorithmic, and computational perspectives, and even the simplest network design problems belong to the class of NP-hard problems. Many classical graph theory problems, for example, the Steiner Tree Problem, the Traveling Salesman Problem, and the two-connected problem, are special cases of the general network design problem.

In this chapter, we first present a general network design model and then consider a number of its special cases. Since the vast scope of network design prevents us from being comprehensive, our objective is to introduce some central network design models, along with their important structural properties and solution algorithms.

11.4.1 General Network Design Model

Preliminaries

DEFINITIONS

D1: A ***network*** is a graph $G = (N, E)$, where N is the set of nodes (or vertices) and E is the set of edges. The nodes denote geographical locations where communication demands originate or terminate, or locations where routing hardware and software are installed. The edges denote potential transmission links. Throughout this section, $|N| = n$ and $|E| = m$.

D2: A ***commodity***, denoted by k, refers to a communication ***demand*** of value d_k (measured in ***bits per second (bps)***) between the ***origin node*** $O(k) \in N$ and ***destination node*** $D(k) \in N$. Let $\mathcal{K}$ denote the set of all commodities and $K = |\mathcal{K}|$.

D3: A ***communication facility***, or simply a ***facility***, is a transmission medium (e.g., a copper cable, a fiber-optic cable, or a cellular tower-to-satellite connection) installed on a network edge. In our discussion, we assume that the facilities are ***undirected***, that is, if a facility of capacity C is installed on an edge, then it permits a total flow of C on the edge in each direction.

D4: If the facility allows unlimited flow of the commodities, it is said to be ***uncapacitated***, otherwise it is said to be ***capacitated***. The capacity measured in *bps* of a facility is referred to as its ***bandwidth***.

NOTATION: Let $\mathcal{L} = \{1, 2, \ldots, L\}$ denote the set of ***facility levels*** ordered by ***facility capacity***; the capacity of the facility level i is C_i, $i = 1, 2 \ldots, L$, where $C_1 \leq C_2 \leq \ldots \leq C_L$.

NOTATION: The (nonnegative) ***fixed cost*** of installing a facility of capacity C_l on the edge $\{i, j\} \in E$ is a_{ij}^l for $l = 1, 2, \ldots, L$, with $a_{ij}^l \leq a_{ij}^{l+1}$. The (nonnegative) ***variable cost*** of sending one unit of commodity k from node i to node j on edge $\{i, j\}$ is denoted b_{ij}^k.

D5: A ***switch*** routes or processes the data signal and can be installed on each node in a subset $M \subseteq N$. Each switch has capacity T, and for each node $i \in M$, t_i denotes the ***switch installation cost***.

D6: The ***network design problem*** is to install capacity, install switches, and route the commodities at minimum cost.

D7: A ***(linear) mixed-integer program (MIP)*** is a problem consisting of a linear multi-variable function to be minimized or maximized subject to linear inequalities involving those variables, where some of the variables are restricted to integer values. An ***integer program (IP)*** is one in which all variables are restricted to be integer-valued, and a ***linear program (LP)*** is one in which no variables are restricted to be integer (i.e., all variables are *continuous*).

NOTATION: The optimal value of any mathematical program P defined above (LP, IP, or MIP) is denoted z_P.

SUMMARY OF NOTATION

- Network $G = (N, E)$, n nodes and m edges.
- Set $\mathcal{K}$ of commodities, $k = 1, 2, \ldots, K$.
- Each commodity k has origin $O(k)$, destination $D(k)$, demand d_k bps, and a variable cost b_{ij}^k of sending one unit from node i to j on edge $\{i, j\}$.
- Set $\mathcal{L}$ of facility levels, $l = 1, 2, \ldots, L$.
- Each facility level l has capacity C_l and fixed installation cost a_{ij}^l on edge $\{i, j\}$.
- Each node $i \in M \subseteq N$ has a switch installation cost t_i.

General Edge-Based Flow Model

The network design problem defined above can be modeled by the mixed integer program shown below.

NOTATION: The ***edge design*** decision variables, upper bounds, any restrictions on the network topology are defined as follows:

- The continuous decision variable f_{ij}^k denotes the ***flow*** of commodity k on edge $\{i,j\}$ from i to j.
- The integer decision variable u_{ij}^l denotes the number of facilities of level l to install on edge $\{i,j\}$, with $u_{ij}^l \leq \mu_{ij}^l$, the upper bound.
- The binary decision variable v_i equals one if a switch is installed on node $i \in M$ and 0 otherwise.
- For each facility level l, U^l is a set specifying any topological restrictions on the design (for example, U^1 might specify that the chosen edge design variables u_{ij}^1 define a cycle). The m-component vector of u_{ij}^1's is denoted u^l.

General Model: Edge-flow [GM:EF]

$$\text{Minimize } \left\{ \sum_{\{i,j\}\in E, l\in\mathcal{L}} a_{ij}^l u_{ij}^l + \sum_{\{i,j\}\in E, k\in\mathcal{K}} b_{ij}^k f_{ij}^k + \sum_{i\in M} t_i v_i \right\}$$

subject to:

$$\sum_{j\in N:\{i,j\}\in E} f_{ji}^k - \sum_{j\in N:\{i,j\}\in E} f_{ij}^k = \begin{cases} -d_k & \text{if } i = O(k) \\ d_k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in N,\ \forall k \in K \qquad \text{(FC)}$$

$$\sum_{k\in\mathcal{K}} f_{ij}^k \leq \sum_{l\in\mathcal{L}} C_l u_{ij}^l \qquad \forall\{i,j\} \in E \qquad \text{(EF2)}$$

$$\sum_{k\in\mathcal{K}} f_{ji}^k \leq \sum_{l\in\mathcal{L}} C_l u_{ij}^l \qquad \forall\{i,j\} \in E \qquad \text{(EF3)}$$

$$\sum_{k\in\mathcal{K}} \sum_{j\in N:\{i,j\}\in E} f_{ji}^k \leq T v_i \qquad \forall i \in M \qquad \text{(EF4)}$$

$$u^l \in U^l \qquad \forall\, l \in \mathcal{L} \qquad \text{(EF5)}$$

$$f_{ij}^k \geq 0 \qquad \forall\{i,j\} \in E,\ \forall k \in \mathcal{K} \qquad \text{(EF6)}$$

$$0 \leq u_{ij}^l \leq \mu_{ij}^l \text{ and integer} \qquad \forall\{i,j\} \in E,\ \forall l \in \mathcal{L} \qquad \text{(EF7)}$$

$$v_i = 0 \text{ or } 1 \quad \forall\, i \in M \qquad \text{(EF8)}$$

REMARKS

R1: The constraints (FC) are the usual *flow conservation* constraints for each commodity at each node. Constraints (EF2) and (EF3) limit the total flow of all commodities in each direction of an edge by its installed capacity, and (EF4) constrains the total flow through nodes in M. Constraints (EF5) restrict the topology of the chosen design. Constraints (EF6) through (EF8) define the nonnegativity and integrality requirements.

R2: This formulation assumes that the initial capacity of the edges is zero. By modifying the right hand side of constraints (EF2) and (EF3), we can model situations where edges have some positive level of initial capacity.

R3: Constraints (EF2) and (EF3) limit the total flow of all commodities on an edge; hence, we refer to them as the *bundle* capacity constraints. In addition, some applications limit the flow for *individual* commodities on each edge. For an edge $\{i, j\}$, if C_{ij}^k denotes the maximum allowable flow of commodity k on the edge from i to j, then the *individual* capacity constraints are of the form:

$$f_{ij}^k \leq C_{ij}^k \quad \forall k \in \mathcal{K}, \ \forall \{i, j\} \in E.$$

R4: Formulation [GM:EF] allows fractional flows although some applications require integral flows. The formulation also allows a commodity to flow over multiple origin-destination paths, called a ***bifurcated*** flow. Some applications might require ***non-bifurcated*** flow where each commodity is required to flow on only one origin-destination path.

R5: Parallel edges in the graph can refer to different transmission technologies, for example, copper cables, fiber-optic cables, or wireless transmission.

R6: In subsequent models that have the flow conservation constraints, we simply write "Flow Conservation Constraints (FC)" instead of rewriting those constraints each time.

General Path-Flow Model

Instead of using the edge-based flow variables as defined in formulation [GM:EF] above, an alternate approach for modeling the network design problem is to use *path-based* flow variables. We replace each edge $\{i, j\}$ in the original network by two oppositely directed arcs (i, j) and (j, i) having the same costs as edge $\{i, j\}$.

NOTATION: The relevant notation for the model given below is as follows:

- For each commodity $k = 1, 2, \ldots, K$, P_k denotes the set of directed paths from origin $O(k)$ to destination $D(k)$.
- The flow (in bps) of commodity k on path $p \in P_k$ is denoted g_p^k.
- The cost of sending one unit of flow of commodity k from origin $O(k)$ to destination $D(k)$ on path p is denoted $c_p^k = \sum_{(i,j) \in p} b_{ij}^k$.

General Model: Path-flow [GM:PF]

$$\text{Minimize } \Big\{ \sum_{\{i,j\} \in E, l \in \mathcal{L}} a_{ij}^l u_{ij}^l + \sum_{k \in \mathcal{K}} \sum_{p \in P_k} c_p^k g_p^k + \sum_{i \in M} t_i v_i \Big\}$$

subject to:

$$\sum_{p \in P_k} g_p^k = d_k \quad \forall k \in \mathcal{K}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in P_k : (i,j) \in p} g_p^k \ \leq \ \sum_{l \in \mathcal{L}} C_l u_{ij}^l \quad \text{and}$$

$$\sum_{k\in\mathcal{K}}\sum_{p\in P_k:(j,i)\in p} g_p^k \leq \sum_{l\in\mathcal{L}} C_l u_{ij}^l \quad \forall\ \{i,j\}\in E$$

$$\sum_{k\in\mathcal{K}}\sum_{p\in P_k:i\in p} g_p^k \leq T v_i \quad \forall i\in M$$

$$u^l \in U^l \quad \forall\ l\in\mathcal{L}$$

$$g_p^k \geq 0 \quad \forall\{i,j\}\in E,\ \forall k\in\mathcal{K}$$

$$0\leq u_{ij}^l \leq \mu_{ij}^l \text{ and integer} \quad \forall\{i,j\}\in E,\ \forall l\in\mathcal{L}$$

$$v_i = 0 \text{ or } 1 \quad \forall i\in M$$

EXAMPLE

E1: Suppose there are seven nodes ($n = 7$), two facility levels (L=2) with capacities $C_1 = 1$ and $C_2 = 10$, $U^1 = U^2 = \emptyset$, and the internodal demand is as follows.

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0
3	10	10	0	0	0	0	0
4	20	0	30	0	0	0	0
5	1	0	0	0	0	0	0
6	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0

In the Figure 11.4.1 below, which gives a feasible solution for this situation, each thick line represents ten units of capacity and each thin line represents one unit of capacity.

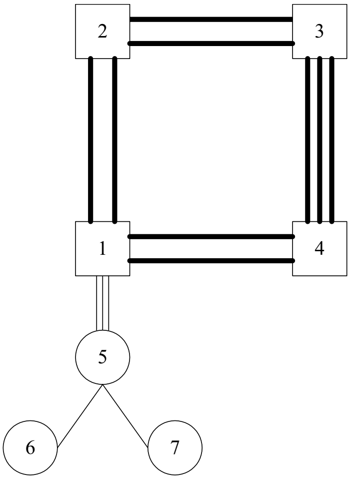


Figure 11.4.1 **Example showing a capacitated network.**

REMARKS

R7: Nodes 1, 2, 3 and 4 in Figure 11.4.1 above, represented by squares, are *hub* nodes, they are typically 2-edge connected by high capacity facilities and form the *backbone network*. Nodes 5, 6, and 7, represented by circles, are *end-office* or *terminal* nodes. Each end-office node is uniquely assigned to a hub. Each subset of end-office nodes assigned to the same hub forms the ***local access network***, which is usually a tree.

R8: Because configuring the entire telecommunication network simultaneously is computationally difficult, it is often done by decomposing the full problem into the backbone network and the local access network design problems.

11.4.2 Uncapacitated Network Design

In certain applications, we can assume that the edge facilities and nodes switches are uncapacitated. Three such scenarios are:

(1) If $\sum_{k \in \mathcal{K}} d_k \leq C_1$, then installing the lowest capacity facility on an edge permits us to send the maximum possible flow on it.

(2) In the telecommunications setting, because fiber-optic cables have high transmission capacity, their capacity may be practically unlimited for some applications.

(3) We may be initially interested in designing the topology of the network only. (A later analysis, if necessary, will determine the edge capacities and flow routes.)

DEFINITIONS

D8: The ***uncapacitated network design model (UND)*** is the network design model that results from the assumption that the edge facilities and node switches are uncapacitated.

D9: A ***linear programming relaxation*** LPR of an integer or mixed-integer program is the linear program that results from relaxing the integrality requirement.

D10: Let IP1 and IP2 denote two minimization integer or mixed-integer programs for a discrete optimization problem. If LP1 and LP2, respectively, denote their linear programming relaxations, then the LP1 relaxation is said to be ***at least as strong*** as the LP2 relaxation if $z_{LP1} \geq z_{LP2}$.

D11: The ***integrality gap*** for a minimization integer (or mixed-integer) program IP is the quantity $(z_{IP} - z_{LPR})/z_{LPR}$.

Uncapacitated Network Design [UND]

For any of the three scenarios mentioned above, we are able to make certain assumptions about our design problem. The last of the three listed below is an additional assumption that we make for the purpose of this presentation.

ASSUMPTIONS

- The design will use only (the least cost) facility 1.
- We can scale all the commodity demands d_k to one, and correspondingly, we can scale the variable flow costs to $b_{ij}^k \leftarrow b_{ij}^k d_k \ \forall k \in \mathcal{K}, \forall \{i, j\} \in E$, and set $C_1 = K$.

- We assume $U^l = \emptyset$ for all $l \in \mathcal{L}$, and we suppress the facility index l.

$$\text{Minimize}\left\{\sum_{\{i,j\}\in E} a_{ij}u_{ij} + \sum_{\{i,j\}\in E, k\in\mathcal{K}} b^k_{ij} f^k_{ij}\right\}$$

subject to:

Flow Conservation Constraints (FC)

$$\sum_{k\in\mathcal{K}} f^k_{ij} \le K u_{ij} \quad \text{and} \tag{UND1}$$

$$\sum_{k\in\mathcal{K}} f^k_{ji} \le K u_{ij} \quad \forall\{i,j\}\in E \tag{UND2}$$

$$f^k_{ij} \ge 0 \quad \forall\{i,j\}\in E,\ \forall k\in\mathcal{K} \tag{UND3}$$

$$u_{ij} = 0 \text{ or } 1 \quad \forall\{i,j\}\in E \tag{UND4}$$

FACTS

F1: When all the flow costs are zero, $K = n-1, O(k) = k$, and $D(k) = n$ for all $k \in \mathcal{K}$, the above formulation models the (polynomially solvable) minimum spanning tree problem.

F2: If the values of the design variables u_{ij} are known, then solving problem UND amounts to finding K shortest paths, one for each commodity k using cost b^k_{ij} on the directed graph $G(N, A)$, where A is the set of directed arcs $(i, j), (j, i)$ for which $u_{ij} = 1, \{i, j\} \in E$.

F3: Replacing the aggregate "***forcing***" constraints (UND1) and (UND2) by the disaggregate constraints

$$f^k_{ij} \le u_{ij} \quad \text{and} \tag{UND5}$$

$$f^k_{ji} \le u_{ij} \quad \forall\{i,j\}\in E,\ \forall k\in\mathcal{K} \tag{UND6}$$

results in an equivalent integer program, [UNDStr]. However, the linear programming relaxation obtained by eliminating the integrality restriction on the u_{ij} variables for Formulation [UNDStr] is at least as strong as the linear programming relaxation of Formulation [UND].

F4: Suppose the costs satisfy the mild flow-cost conditions:

(a) $b^k_{ij} = b^k_{ji} = b_{ij}$ for all $k \in \mathcal{K}$ and for all edges $\{i, j\} \in E$;

(b) $b_{ij} + b_{ji} \ge 0$ for all $\{i, j\} \in E$.

In this case [BaMaWo89], if two commodities k_1 and k_2 share the same origin or the same destination, then there exists an optimal [UND] solution in which k_1 and k_2 flow in the same direction on every edge, that is,

$$f^{k_1}_{ij} + f^{k_2}_{ji} \le u_{ij} \quad \forall k_1, k_2\ \big(O(k_1) = O(k_2) \text{ or } D(k_1) = D(k_2)\big),\ \forall\{i,j\}\in E \tag{UND7}$$

F5: Balakrishnan et al. [BaMaWo89] develop a very effective dual ascent method that approximately solves the dual of formulation [UNDStr] and includes constraints (UND7). Using local search (*add–drop*) heuristics on an initial solution provided by the dual heuristic results in low performance gaps on large-scale, randomly generated problems.

F6: The results in [BaMaWo89] indicate that the integrality gap of formulation [UNDStr] with constraints (UND7) is, on average, small for randomly generated problems. [BaMaMi98] conduct a theoretical analysis of the uncapacitated network design model and develop heuristics with a worst-case integrality gap of $\sqrt{K}$.

EXAMPLES

E2: Consider a complete graph on three nodes $1, 2, 3$ with three commodities defined by $O(1) = O(2) = 1, O(3) = 2, D(1) = 2, D(2) = D(3) = 3$. Let the fixed costs equal one and the flow costs equal zero for all three edges, i.e., $a_{12} = a_{13} = a_{23} = 1$ and $b_{12} = b_{13} = b_{23} = 0$. Setting $u_{12} = u_{13} = u_{23} = 1/3$ and $f^1_{12} = f^2_{13} = f^3_{23} = 1$ gives a feasible solution of cost 1 to the linear programming relaxation of formulation [UND]. However, this solution is not feasible to the linear programming relaxation of formulation [UNDStr]. The optimal solution to the linear programming relaxation of formulation [UNDStr] costs $3/2$ ($u_{12} = u_{13} = u_{23} = 1/2, f^1_{12} = f^1_{13} = f^1_{32} = f^2_{13} = f^2_{12} = f^2_{23} = f^3_{23} = f^3_{21} = f^3_{13} = 1/2$), which shows, in this case, that the relaxation [UNDstr] is stronger than the relaxation of [UND].

E3: Constraints (UND7) strengthen the linear programming relaxation of formulation [UNDStr]. Consider a 3-node, 2-commodity network with $O(1) = O(2) = 1, D(1) = 2, D(2) = 3$. As in Example E2, $a_{12} = a_{13} = a_{23} = 1$, and $b_{12} = b_{13} = b_{23} = 0$. The solution to the linear programming relaxation of the UNDStr model sets all three design variables to $1/2$; if the model is enhanced by adding constraints (UND7), the linear programming relaxation obtains the optimal solution by setting $u_{12} = u_{13} = 1$.

Multi-Level Network Design

Another uncapacitated network design that occurs in many telecommunication and transportation settings is one in which the nodes are classified into a hierarchy of groups based on their importance, and different grades of facilities are available. The more critical nodes need to be connected using higher grade (level) facilities. The ***Multi-Level Network Design (MLND)*** model considers topological network design applications in such hierarchical settings.

DEFINITIONS

D12: In a ***multi-level network design (MLND)*** problem, the facility levels are $l = 1, 2, \ldots, L$, the node-set N is partitioned into $N_1, N_2, ..., N_L$ non-empty levels (groups), and we assume that the higher indexed node groups are more critical than the lower indexed ones and the facility levels are indexed in increasing order of grade (and expense). The objective is to assign a facility level to each selected design edge subject to the constraint that every pair of nodes $i \in N_{l'}$ and $j \in N_{l''}$ can communicate along a path that uses only facilities at level at least $\min(l', l'')$. We assume that there are no flow costs and that the facilities are uncapacitated.

NOTATION: Installing a level-l facility on each edge $\{i, j\}$ of the network costs a^l_{ij}, where $a^1_{ij} \leq a^2_{ij} \leq \cdots \leq a^L_{ij}$.

D13: The MLND problem has a ***proportional cost structure*** if the ratio (a^l_{ij}/a^{l-1}_{ij}), $l = 2, \ldots, L$, is the same for all edges.

D14: The ***Steiner tree problem***: Given a weighted graph in which a subset of vertices are identified as terminals, find a minimum-weight connected subgraph that includes all the terminals. In an optimal solution, the non-terminal nodes are called ***Steiner nodes***.

Multi-Level Network Design Model [MLND]

In the formulation of the MLND problem shown below [BaMaMi94a], the variable u_{ij}^l equals 1 if a level-l facility is installed on edge $\{i, j\}$ and 0 otherwise. We also define commodities $k = 1, 2 \ldots, n-1$ such that for each commodity k, $O(k) = n, D(k) = k$, and $d_k = 1\ \forall k$.

$$\text{Minimize} \sum_{\{i,j\}\in E} \sum_{l=1}^{L} a_{ij}^l u_{ij}^l$$

subject to:

Flow Conservation Constraints (FC)

$$\begin{array}{ll} f_{ij}^k \le \sum\limits_{l\le l'\le L} u_{ij}^{l'} & \text{and} \\ f_{ji}^k \le \sum\limits_{l\le l'\le L} u_{ij}^{l'} & \forall\{i,j\}\in E,\ \forall k : D(k)\in N_l \\ f_{ij}^k \ge 0 & \forall\{i,j\}\in E,\ \forall k\in\mathcal{K} \\ u_{ij}^l = 0 \text{ or } 1 & \forall\{i,j\}\in E,\ \forall l = 1,2,\ldots,L \end{array}$$

FACTS

F7: Since the edge costs are nonnegative, the edges chosen by the optimal MLND solution define a tree.

F8: For each $l, l = 2, 3, \ldots, L$, the subtree in the optimal solution defined by facilities at level l is embedded in the (sub) tree defined by facilities at level $l-1$.

F9: When $L = 2$ and $a_{ij}^1 = 0$, the MLND problem is equivalent to the Steiner tree problem with N_2 defining the terminal nodes and N_1 defining the Steiner nodes.

F10: Since the MLND problem generalizes the Steiner problem, the MLND problem is NP-hard. The problem continues to be NP-hard when $L = 2, |N_2| = 2$ and (i) the costs are proportional or (ii) $a_{ij}^2 = 1$ and $a_{ij}^1 = 0$ or 1 for all $\{i, j\} \in E$ [Or91].

MLND Composite Heuristic

The following composite heuristic [BaMaMi94a] for $L = 2$ takes the better of two heuristic values to develop a worst-case performance bound for the heuristic solution value relative to the optimal solution value.

Step 1. Minimum Spanning Tree (Forward) heuristic: Treat the level-1 nodes as level-2 nodes, and find the minimum spanning tree in $G(N, E)$ using costs a_{ij}^2. Set $u_{i,j}^2 = 1$ on all edges of this tree to get a feasible solution.

Step 2. Steiner Overlay (Backward) heuristic: Find the Minimum Spanning Tree T_{MLND} spanning all nodes using costs a_{ij}^1. Using the incremental costs $a_{ij}^2 - a_{ij}^1$ for all edges in the minimum spanning tree, and a_{ij}^2 otherwise, solve a Steiner tree problem with N_2 as the terminal nodes and N_1 as the Steiner nodes. Install level-2 facilities on the edges of the Steiner tree and level-1 facilities on the remaining edges of T_{MLND} if they do not create a cycle.

COMPUTATIONAL NOTE: If $\rho < 2$ denotes the worst-case performance ratio of the Steiner tree solution method used in *Step 2*, then the worst-case performance ratio of the MLND composite heuristic is $\frac{4}{4-\rho}$ for proportional costs and $\rho + 1$ for general costs.

COMPUTATIONAL NOTE: Although, for simplicity, we have described the heuristic for $L = 2$, it generalizes to an exponential run-time recursive heuristic for the general MLND problem (the composite heuristic iterates $O(2^L)$ times). For the case when $L = 3, \rho = 1$ and costs are proportional, this generalized recursive heuristic provides a worst-case performance ratio of 1.52241 [Mi96].

REMARKS

R9: As with the uncapacitated network design problem, we can "directize" the edges of the MLND model (i.e., replace each undirected edge by two oppositely directed arcs). A solution approach based on the dual of this directized formulation solves large, randomly generated test cases producing solutions that are within 0.9 percent of the optimal solution for $L = 2$ and within 6 percent of the optimal solution for $L = 5$.

R10: The MLND model can be generalized by imposing more restrictive connectivity requirements. For example, we may require that the subgraph defined by the l-level facilities be k_l-connected for a pre-specified integer $k_l, \forall l$. The extant literature has not addressed this generalization even when $L = 2$, $k_1 = 1$, and $k_2 = 2$.

11.4.3 Survivable Network Design (SND)

Communication networks designed purely from a cost minimization perspective to satisfy commodity demand tend to be sparse, and an edge (representing a transmission facility) or a node (representing a communication switch) failure can lead to interruptions in communication service. Therefore, network designers build in redundancy by providing alternate communication paths in the network, so that the network can continue to satisfy communication demands even after a failure. The degree of network redundancy depends on the trade-off between network cost and the importance of maintaining the connection between each pair of nodes. Since the probability of the simultaneous failure of two or more elements (edges or nodes) is very small, it is generally assumed for network planning purposes that only one failure occurs at a time.

Uncapacitated Survivable Network Design [SNDUnc]

The mathematical program given below is based on the concept of an *edge-cut*, a set of edges of whose deletion disconnects the network. Edge-cuts and their relation to connectivity and internally-disjoint paths (Menger's theorem) are discussed in §4.1 and §4.7. The connection between edge-cuts and the algebraic structure of a graph is presented in §6.4, and their role in finding maximum flows in networks is discussed in §10.1.

DEFINITIONS

D15: A network is said to be ***survivable*** if it can continue to satisfy demand even when one of its edges or nodes fails.

D16: The ***connectivity requirement*** between nodes $i, j \in N, i \neq j$, denoted by a nonnegative integer r_{ij}, is the minimum number of edge-disjoint paths needed between i and j.

D17: When the maximum connectivity requirement is no more than two, the problem is said to be a ***low connectivity survivable [LCS]*** network design problem.

D18: Let $G = (N, E)$ be a network and let S be a proper nonempty subset of the node-set N. The ***(edge-)cut*** defined by S, denoted $\langle S, N \setminus S \rangle$, is the set of edges defined by

$$\langle S, N \setminus S \rangle = \{\{i, j\} \in E \mid i \in S \text{ and } j \in N \setminus S\}$$

NOTATION: For node subsets A and B, let $[A, B]$ denote the set of node-pairs given by

$$[A, B] \doteq \{i, j \mid i \in A \text{ and } j \in B\}$$

REMARKS

R11: A network with the connectivity requirements as in Definition 16 will continue to have at least one path between every pair of nodes i and j for which $r_{ij} \geq 2$, even when any node in $N \setminus \{i, j\}$ fails.

R12: Since the graph is undirected, we can assume that the connectivity requirements are symmetric, that is, $r_{ij} = r_{ji}$.

R13: When both i and j are backbone nodes, r_{ij} is at least two, and when either of them is a local access node, r_{ij} is typically one. In addition, $r_{ij} = 0$ if either i or j is an optional (Steiner) node that the network can, but is not required to, use.

Cut-Based Formulation of SND [SND-CUT]

If $S \subset N$, the basic cut formulation of the survivable network design (SND) problem is given below.

$$\text{Minimize} \sum_{\{i,j\} \in E} a_{ij} u_{ij}$$

subject to:

$$\sum_{\{i,j\} \in \langle S, N \setminus S \rangle} u_{ij} \geq \max_{i,j \in [S, N \setminus S]} \{r_{ij}\} \quad \text{for all proper nonempty } S \subset N$$

$$u_{ij} = 0 \text{ or } 1 \quad \text{for all } \{i, j\} \in E$$

FACTS

F11: If there exist ***node levels*** $r_i, i \in N$ such that $r_{ij} = \min(r_i, r_j)$, then an optimal solution to SND has at most one 2-connected component.

F12: For a slightly more general connectivity requirement function, using a *primal-dual* solution approach, [WiGoMiVa95] develop a worst-case performance bound of $2R$, where R denotes the maximum connectivity level; [GoGoPlShTaWi94] improve this bound to $2(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{R})$.

EXAMPLE

E4: Figure 11.4.2 shows an example of a survivable network with $R = 3$ and with two 2-connected components.

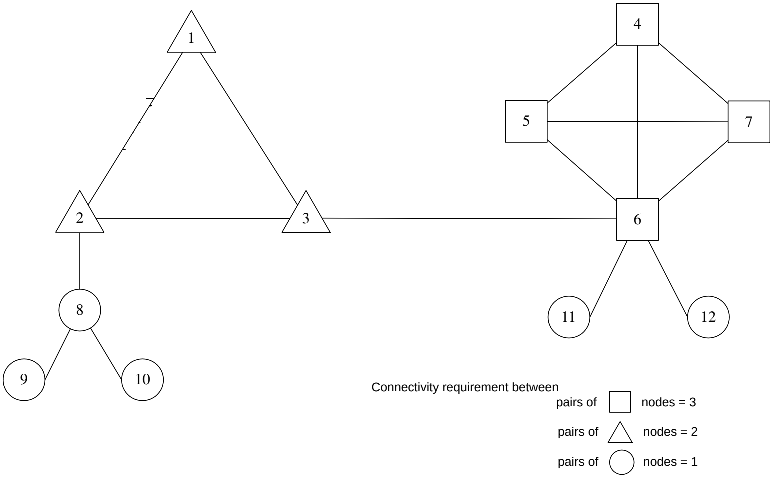


Figure 11.4.2 **Solution to an SND problem.**

SND Iterative Rounding Heuristic

[Ja01] develops a linear programming based iterative rounding heuristic, presented below, that has a (tight) worst-case performance bound of two relative to the linear programming relaxation value of formulation [SND-CUT]. The two-step strategy repeats until the heuristic finds a feasible integer SND solution.

Step 1. Find an optimal basic solution $u_{ij}, \{i,j\} \in E$ to the linear programming relaxation of Formulation [SND-CUT]. (Comment: Except for the first iteration, some of the u_{ij} variables would have been set to one in Step 2.)

Step 2. For all edges $\{i,j\}$ with $u_{ij} \geq \frac{1}{2}$, fix u_{ij} to one.

REMARK

R14: Using insightful arguments, Jain shows that in any optimal basic solution to the formulation [SND-CUT], at least one u_{ij} variable is at least $\frac{1}{2}$. Rounding this edge to one at most doubles its contribution to the solution value. Jain departs from traditional rounding heuristics that solve one linear program and then simultaneously round all fractional solutions suitably. By setting only those variables with value at least a half to one and resolving the linear program, he finds the "valuable" edges in each iteration, and provides the first known SND heuristic with a constant worst-case performance bound.

NOTATION: Let $G = (N, E)$ be a network with requirements r_{ij} as defined above, and let $A, B \subset N$. Let $[A, B]_{r \geq 2}$ denote the set of node-pairs i, j whose connectivity requirement is at least 2. That is, $[A, B]_{r \geq 2} = \{i, j \mid i \in A, j \in B \text{ and } r_{ij} \geq 2\}$.

FACTS

F13: The following constraint models the redundancy requirement that the network contain at least two node-disjoint paths between every pair of nodes i and j for which $r_{ij} \geq 2$.

$$\sum_{\{i,j\} \in E \;:\; i,j \in [S,\ (N \setminus \{S \cup \{z\}\})]} u_{ij} \geq 1$$

$$\forall z \in N, \quad \text{for all nonempty } S \not\subseteq N \setminus \{z\} \;:\; [S,\ (N \setminus \{S \cup \{z\})]_{r \geq 2} \neq \emptyset$$

F14: [GrMoSt90] and [GrMoSt92] identify facets, develop preprocessing routines, and develop a successful cutting plane approach for an LCS network design problem with both edge and node failures permitted.

Flow-Based Formulation of SND [SND-FLOW]

By using flow variables, we can model the SND problem as a special case of the general network design model.

NOTATION: For every pair of nodes with $r_{ij} > 0$, define a commodity k such that $d_k = r_{ij}$, $O(k) = i$, $D(k) = j$, and set $L = 1$.

$$\text{Minimize} \sum_{\{i,j\} \in E} a_{ij} u_{ij}$$

subject to:

Flow Conservation Constraints (FC)

$$\begin{aligned}
&f_{ij}^{k} \leq u_{ij} \quad \text{and} \\
&f_{ji}^{k} \leq u_{ij} \qquad \forall\ \{i,j\} \in E,\ \forall k \in \mathcal{K} \\
&f_{ij}^{k} \geq 0 \qquad \forall\ \{i,j\} \in E,\ \forall k \in \mathcal{K} \\
&u_{ij} = 0 \text{ or } 1 \quad \forall\ \{i,j\} \in E
\end{aligned}$$

REMARK

R15: [BaMi02] identifies several valid inequalities to strengthen the flow-based formulation and test their computational effectiveness for the case when the SND solution has at most one 2-connected component.

Survivable Network Design: Bounded Cycles

A limitation of both the SND formulations discussed above is that they allow long cycles. Thus, in an extreme case, the solution to an LCS network design problem might be a minimum cost Hamiltonian cycle through all the nodes. In such a solution, any edge failure requires the rerouting of the affected demands using long alternate paths. To prevent long cycles, we can impose the condition that every chosen edge belong to at least one cycle with length bounded by a specified constant.

NOTATION: We use the following notation to impose this requirement:

- For each edge $\{i,j\} \in E$, Y_{ij} denotes the set of cycles that contain $\{i,j\}$ and satisfy the length bound.
- For edge $\{i,j\} \in E$ and cycle $C \in Y_{ij}$, y_{ij}^C denotes a binary variable that is one if cycle C is included in the solution and zero otherwise. The following (exponentially-sized) set of constraints impose the bounded cycles condition.

$$\sum_{C \in Y_{ij}} y_{ij}^C \geq u_{ij} \quad \forall \{i,j\} \in E$$

$$\sum_{C \in Y_{ij}, \{i',j'\} \in C} y_{ij}^C \leq u_{i'j'} \quad \forall \{i,j\} \in E,\ \{i',j'\} \in E \setminus \{\{i,j\}\}$$

$$y_{ij}^C = 0 \text{ or } 1 \quad \forall \{i,j\} \in E, \quad C \in Y_{ij}$$

REMARKS

R16: Fortz et al. [FoLaMa00] design a *branch-and-cut* approach for the design of minimum-cost bounded-cycles networks that contain two node-disjoint paths between every pair of nodes.

R17: Another way of making a network robust to failures is to limit the number of edges (also called the number of *hops*) in the path used to satisfy the demand for each commodity (e.g., ([BaAl92], [Go98]).

11.4.4 Capacitated Network Design

In our discussion so far, we have not considered the capacity constraints appearing in the general models of §11.4.1. Many practical applications require activation of these constraints; such capacitated situations result in very hard optimization problems even when there are no node capacities and the edge capacity levels may be chosen from a continuous range (see, for example, [Mi81]). We describe three models below in each of which the capacities are available at discrete levels.

Network Loading Problem [NLP]

In telecommunication settings, we often encounter situations where we have L different types of facilities available such that their capacities are *modular*, that is, the capacity of a level-$(l+1)$ facility is a multiple of the capacity of a level-(l) facility, $l = 1, 2, \ldots, L-1$. Moreover, we can install (load) any number of facilities on an edge. We refer to this problem as the ***Network Loading Problem (NLP)***. The number of different types of facilities is often small (less than five) in practical situations. For example, in some telecommunications settings, *T1* and *T3* facilities may be the available transmission facilities. A T3 facility has 28 times the capacity of a T1 line, but costs less than 28 times the cost of a T1 line. This results in economies of scale in the fixed costs structure for any edge. Figure 11.4.3 below depicts an illustrative cost structure when we have two facilities and the capacity of the higher level facility is 12 times the capacity of the lower level facility. Note that, in this example, the break-even point is eight.

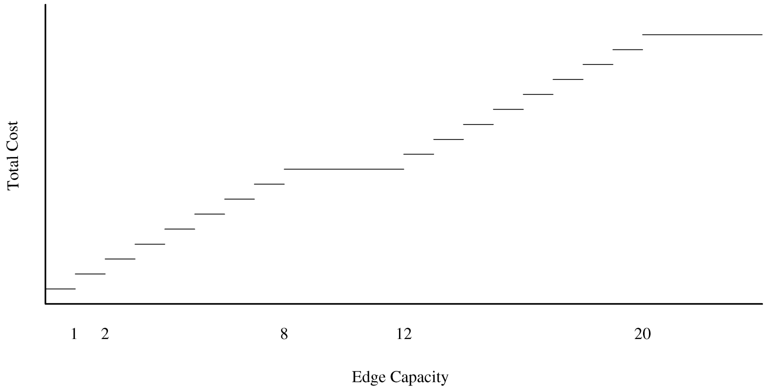


Figure 11.4.3 **Cost Structure for a Network Loading Problem.**

Network Loading Model [NLP]

The following formulation models the Network Loading Problem when $L = 2$ with $C_1 = 1$ and $C_2 = C$.

$$\text{Minimize } \left\{ \sum_{\{i,j\} \in E} (a_{ij}^1 u_{ij}^1 + a_{ij}^2 u_{ij}^2) + \sum_{\{i,j\} \in E,\ k \in \mathcal{K}} b_{ij}^k f_{ij}^k \right\}$$

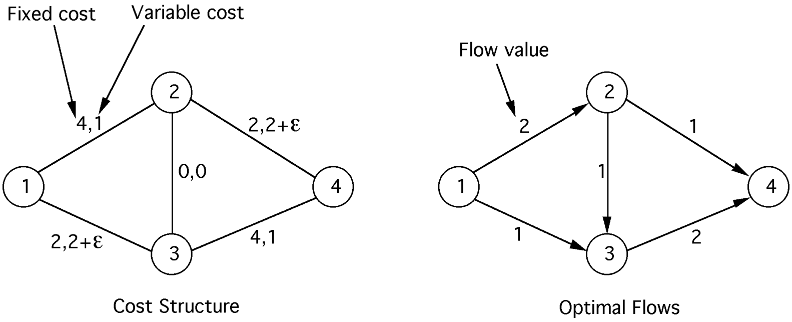
subject to:

Flow Conservation Constraints (FC)

$$\begin{aligned} \sum_{k \in \mathcal{K}} f_{ij}^k &\leq u_{ij}^1 + C u_{ij}^2 \quad \text{and} \\ \sum_{k \in \mathcal{K}} f_{ji}^k &\leq u_{ij}^1 + C u_{ij}^2 \quad \forall \{i,j\} \in E \\ f_{ij}^k &\geq 0 \qquad\qquad \forall \{i,j\} \in E,\ \forall k \in \mathcal{K} \\ u_{ij}^1, u_{ij}^2 &= 0 \text{ or } 1 \qquad \forall \{i,j\} \in E \end{aligned}$$

EXAMPLE

E5: Consider the single-commodity case (i.e., $K = 1$) with $L = 1, C_1 = C$. If d denotes the commodity demand, intuition suggests that the solution will use at most two paths: one carrying $(\lceil d/C \rceil - 1)C$ units of flow, and the other carrying $d - (\lceil d/C \rceil - 1)C$ units of flow. The following example ([Mi89]) (shown in Figure 11.4.4 below) shows that this intuition is not correct. In this example, the commodity origin is node 1, the commodity destination is node 4, $d = 3$ and $C = 2$.



Cost Structure

Optimal Flows

Figure 11.4.4 **An optimal solution with three flow paths.**

FACTS

F15: [MaMiVa95] and [BiGü95] develop and implement a polyhedral approach for solving the network loading problem, and [MaMiVa92] provide a complete polyhedral description of two of its core subproblems.

F16: The single-commodity network loading problem with one facility type and flow costs, or with two facility types and zero flow costs is NP-hard ([ChGiSa98]).

Capacitated Concentrator Location [CCL]

The design of local access networks (recall that local access networks are frequently trees) requires connecting the end-user nodes (terminals) on the local access tree to consolidation points, called *concentrators* on the tree. These concentrators compress the data signal and transmit it directly to the backbone network hub to which the local access tree is connected.

DEFINITION

D19: In the ***capacitated concentrator location*** problem (***CCL***), we are given the local access tree $G(N, E)$ with node-set N and edge-set E, the demands d_i at the terminals $i \in N$, a set $M \subseteq N$ of possible sites for locating concentrators, the cost t_j associated with installing a concentrator of capacity T at node $j \in M$, and the cost a_{ij} of connecting terminal i to concentrator j. We want to determine the minimum cost location of the concentrators and assign each terminal to exactly one concentrator. The assignment should not violate the concentrator capacity constraint, nor the *contiguity* constraint: if terminal i is assigned to concentrator j, then all terminals on the unique path in G, P_{ij}, from node i to concentrator j are also assigned to the concentrator at node j.

NOTATION: Let v_j equal one if a concentrator is located at node $j \in M$ and 0 otherwise, and let u_{ij} equal one if terminal $i \in N$ is served by a concentrator at site $j \in M$.

Capacitated Concentrator Location Model [CCL]

The (CCL) problem can be formulated as the following 0-1 integer linear program:

$$\text{Minimize} \left\{ \sum_{i \in N} \sum_{j \in M} a_{ij} u_{ij} + \sum_{j \in M} t_j v_j \right\}$$

$$\sum_{j \in M} u_{ij} = 1 \qquad \forall i \in N \tag{CCL1}$$

$$\sum_{i \in N} d_i x_{ij} \le T v_j \qquad \forall j \in M \tag{CCL2}$$

$$u_{ij} \le v_j \qquad \forall i \in N,\ \forall j \in M \tag{CCL3}$$

$$u_{i'j} \ge u_{ij} \qquad \forall i' \in P_{ij} \tag{CCL4}$$

$$u_{ij} = 0 \text{ or } 1 \qquad \forall i \in N, \forall j \in M \tag{CCL5}$$

$$v_j = 0 \text{ or } 1 \qquad \forall j \in M \tag{CCL6}$$

REMARKS

R18: In formulation [CCL], the first set of constraints, CCL1, ensures that each terminal is connected to exactly one concentrator, and the CCL2 models the concentrator capacity constraint. The CCL3 constraints ensure that terminal i can be assigned to a concentrator only if it has been installed. The CCL4 constraints model the contiguity requirement.

R19: This problem can be considered to be a generalization of the bin-packing problem but with an additional cost incurred for making the item and bin assignments and the contiguity constraint on the items.

R20: Another problem that arises in local access network is the *capacitated minimum spanning tree* problem ([AlGa88], [Ga91]). In this problem, there are $n-1$ commodities and one of the nodes of $G(N, E)$, say node n, is the destination node for all commodities. The origin node for commodity k is k, $k = 1, 2, \ldots, n-1$ and its demand is d_k. The cost of using edge $\{i, j\}$ is a_{ij} and all flow costs equal zero. We need to select the least cost subset of E such that (i) the subset edges form a spanning tree of $G(N, E)$, and (ii) the sum of the demands of the nodes included in each subtree formed by deleting all the chosen edges incident to node n does not exceed a pre-specified capacity limit. A number of heuristic and optimization based approaches have been developed for this problem.

Survivable Network Design (Capacitated)

When we are designing capacitated networks, one way of improving survivability is to have two edge-disjoint paths, a working path and a backup path from $O(k)$ to $D(k)$, each with a dedicated capacity of d_k for meeting the demand for commodity k. While this approach, called ***1+1 Diverse Path Protection***, provides the necessary survivability instantaneously, it does so at a high cost because it more than doubles the capacity (the backup path is typically longer than the working path) in the network. Therefore, network designers have devised a number of other ways of imposing the survivability condition in capacitated networks. We discuss two of these approaches. The first uses *self-healing rings* (SHRs) (which may still provide dedicated protection capacity against a failure) and the second one limits the amount of disrupted flow by using a *diversification* and *reservation* strategy.

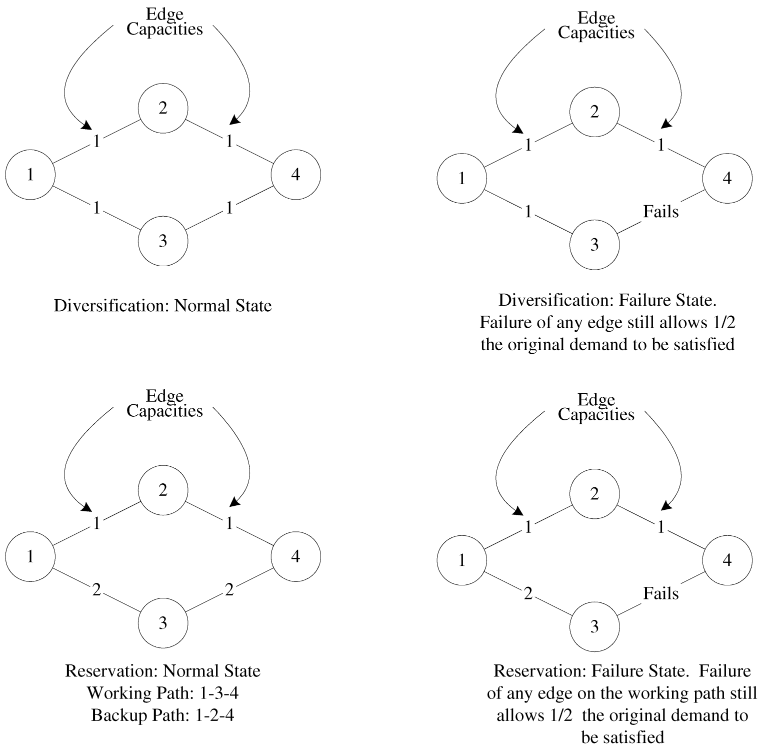
DEFINITIONS

D20: *Self-Healing Rings Approach.* ***Self-healing rings*** (SHRs) are cycles in the network formed by groups of nodes. Different SHRs may share edges (and thus be connected to each other); together the SHRs cover all the demand nodes. A switching device (called Add-Drop Multiplexer) is placed at the nodes that connect two SHRs and allows the signal to be transferred between the SHRs. Each edge in an SHR permits the signal to flow in both directions. Hence, each pair of nodes in an SHR is connected by two edge and two node disjoint paths. Therefore, any signal flowing through a ring is protected against a single edge or a single node failure on that ring. A number of different design problems arise when SHRs are used (see, for example, [SoWySéLaGeFo98]).

D21: *Diversification and Reservation Approach.* ***Diversification*** splits the flow of commodity k such that no more than a fraction δ_k flows through any edge or node (except $O(k)$ or $D(k)$). ***Reservation*** reserves enough spare capacity in the network such that it can reroute at least a fraction of ρ_k of commodity k if an edge or a node fails.

EXAMPLE

E6: [PaJoAlGrWe96] Figure 11.4.5 illustrates the difference between diversification and reservation. For this example, $K = 1, O(1) = 1, D(1) = 4$, and $d_1 = 2$.



Diversification: Normal State

Diversification: Failure State. Failure of any edge still allows 1/2 the original demand to be satisfied

Reservation: Normal State
Working Path: 1-3-4
Backup Path: 1-2-4

Reservation: Failure State. Failure of any edge on the working path still allows 1/2 the original demand to be satisfied

Figure 11.4.5 **Two ways of enhancing network survivability.**

Diversification and Reservation Model [DR]

In the model given below, all flow costs are zero.

NOTATION: The model uses the following notation:

- $\theta \in \Theta$ denotes the operating state of the network, where $\theta = 0$ denotes the normal operating state (when all edges and nodes are operational).
- The state when node i, $i \in N$, breaks down is denoted $\theta = i$, and $\theta = \{i,j\}$, $\{i,j\} \in E$, denotes the state when edge $\{i,j\}$ breaks down.
- $G(\theta) = (N(\theta), E(\theta))$, where $N(\theta)$ and $E(\theta)$ are the sets of nodes and edges, respectively, that are still operating under state θ.
- $P_k(\theta)$ denotes the set of feasible paths from $O(k)$ to $D(k)$ under operating state θ.

$$\text{Minimize} \quad \sum_{\{i,j\}\in E} \sum_{l=1}^{L} a_{ij}^l u_{ij}^l$$

subject to:

$$\sum_{k\in\mathcal{K}} \sum_{p\in P_k(\theta):\{i,j\}\in p} g_p^k(\theta) \;\le\; \sum_{l\in\mathcal{L}} C_l u_{ij}^l \qquad \forall\ \{i,j\} \in E,\ \forall \theta \in \Theta \tag{DR1}$$

$$\sum_{p\in P_k(0)} g_p^k(0) \;=\; d_k \qquad \forall k \in \mathcal{K} \tag{DR2}$$

$$\sum_{p\in P_k(\theta)} g_p^k(\theta) \;=\; \rho_k d_k \qquad \forall k \in \mathcal{K},\ \forall \theta \in \Theta \setminus \{0\} \tag{DR3}$$

$$\sum_{p\in P_k(0):i\in p} g_p^k(0) \;\le\; \delta_k d_k \qquad \forall k \in \mathcal{K},\ \forall i \in N \setminus \{O(k), D(k)\} \tag{DR4}$$

$$g_p^k(0) \;\le\; \delta_k d_k \qquad \forall k \in \mathcal{K}, p = \{O(k), D(k)\} \tag{DR5}$$

$$g_p^k(\theta) \;\ge\; 0 \qquad \forall\ \{i,j\} \in E, k \in \mathcal{K},\ \forall \theta \in \Theta \tag{DR6}$$

$$u_{ij}^l \;\ge\; 0 \text{ and integer} \qquad \forall\ \{i,j\} \in E, l \in \mathcal{L} \tag{DR7}$$

REMARK

R21: In the formulation above, (DR1) are the capacity constraints. Constraints (DR2) ensure that the full demand of each commodity is routed under normal operating conditions (no failures), and constraints (DR3) ensure that at least a fraction ρ_k of commodity k is routed under all other operating states. The next set of constraints (DR4) ensures that no node (and hence no edge other than the direct edge $\{O(k), D(k)\}$) carries a flow of more than δ_k. Constraints (DR5) ensure this diversification for direct edges. The remaining constraints are nonnegativity and integrality constraints. See [StDa94] and [AlGrJoPaWe98] for a discussion of cutting plane approaches for solving such models.

References

[AlGa88] K. Altinkemer and B. Gavish, Heuristics with Constant Error Guarantees for Topological Design of Local Access Tree Networks, *Management Science*, 34 (1988), 331–341.

[AlGrJoPaWe98] D. Alevras, M. Grötschel, P. Jonas, U. Paul and R. Wessäly, Survivable Mobile Phone Network Architectures: Models and Solution Methods, *IEEE Communications Magazine*, 36 (1998), 88–93.

[AlGrWe98] D. Alevras, M. Grötschel, R. Wessäly, Cost-Efficient Network Synthesis from Leased Lines, *Annals of Operations Research*, 76 (1998),1–20.

[ArHaMa90] G. J. R. Araque, L. Hall and T. Magnanti, Capacitated Trees, Capacitated Routing and Associated Polyhedra, *C.O.R.E. Discussion Paper*, 1990.

[BaAl92] A. Balakrishnan, and K. Altinkemer, Using a Hop-constrained Model to Generate Alternative Communication Network Designs, *INFORMS Journal on Computing*, 4, (1992), 192–205.

[BaMaMi94a] A. Balakrishnan, T. L. Magnanti and P. Mirchandani, A Dual-based Algorithm for Multi-level Network Design, *Management Science*, 40 (1994), 567–581.

[BaMaMi94b] A. Balakrishnan, T. L. Magnanti and P. Mirchandani, Modeling and Heuristic Worst-case Performance Analysis of the Two-Level Network Design Problem, *Management Science*, 40 (1994), 846–867.

[BaMaMi97] A. Balakrishnan, T. L. Magnanti and P. Mirchandani, Network Design, In *Annotated Bibliographies in Combinatorial Optimization*, M. Dell'Amico, F. Maffioli, and S. Martello (editors), John Wiley and Sons, New York (1997) 311–334.

[BaMaMi98] A. Balakrishnan, T. L. Magnanti and P. Mirchandani, Designing Hierarchical Survivable Networks, *Operations Research*, 46 (1998), 116–136.

[BaMaWo89] A. Balakrishnan, T. L. Magnanti and R. T. Wong, A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design, *Operations Research*, 37 (1989), 716–740.

[BiChGüTs96] D. Bienstock, S. Chopra, O. Günlük and C. Tsai, Minimum Cost Capacity Installation for Multi-Commodity Network Flows, *Mathematical Programming*, 81 (1996), 177–199.

[BiDeSi94] D. Bienstock, Q. Deng and D. Simchi-Levi, A Branch-and-cut Algorithm for the Capacitated Minimum Spanning Tree Problem, 1994.

[BiGü95] D. Bienstock and O. Günlük, Computational Experience with a Difficult Mixed-integer Multicommodity Flow Problem, *Mathematical Programming*, 68 (1995), 213–237.

[ChGiSa98] S. Chopra, I. Gilboa and T. Sastry, Source Sink Flows with Capacity Installation in Batches, *Discrete Applied Mathematics*, 85 (1998), 165–192.

[EsWi66] L. R. Esau and K. C. Williams, On Teleprocessing System Design, *IBM Systems Journal*, 5 (1966), 142–147.

[FoLaMa00] B. Fortz, M. Labbé and F. Maffioli, Solving the Two-Connected Network with Bounded Meshes Problem, *Operations Research*, 48 (2000), 866–877.

[Ga85] B. Gavish, Augmented Lagrangian based Algorithms for Centralized Network Design, *IEEE Trans. Commun.*, COM-33 (1985), 1247–1257.

[Ga91] B. Gavish, Topological Design of Telecommunication Networks — Local Access Design Methods, *Annals of Operations Research*, 33 (1991), 17–71.

[Go98] L. Gouveia, Using Variable Redefinition for Computing Lower Bounds for Minimum Spanning and Steiner Trees with Hop Constraints, *INFORMS Journal on Computing*, 10 (1998), 180–188.

[GoBe93] M. X. Goemans and D. J. Bertsimas, Survivable Networks, Linear Programming Relaxations and the Parsimonious Property, *Mathematical Programming*, 60 (1993), 145–166.

[GoGoPlShTaWi94] M. X. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos, and D. P. Williamson, Approximation Algorithms for Network Design Problems, *SODA*, (1994), 223–232.

[GrMoSt90] M. Grötschel, C. L. Monma and M. Stoer, Integer Polyhedra Arising from Certain Design Problems with Connectivity Constraints, *SIAM Journal on Discrete Mathematics*, 3 (1990), 502–523.

[GrMoSt92] M. Grötschel, C. L. Monma and M. Stoer, Computational Results with a Cutting Plane Algorithm for Designing Communication Networks with Low-connectivity Constraints, *Operations Research*, 40 (1992), 309–330.

[GrMoSt96] M. Grötschel, C. L. Monma and M. Stoer, Design of Survivable Networks. In Volume *Network Models* of *Handbooks in Operations Research and Management Science*, M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser (editors), Elsevier Science, The Netherlands, (1995), 617–672.

[Ja01] K. Jain, A Factor-2 Approximation Algorithm for the Generalized Steiner Network Problem, *Combinatorica*, 21 (2001), 39–60.

[MaMiVa92] T. L. Magnanti, P. Mirchandani and R. Vachani, The Convex Hull of Two Core Capacitated Network Design Problems, *Mathematical Programming*, 60 (19952), 233–250.

[MaMiVa95] T. L. Magnanti, P. Mirchandani and R. Vachani, Modeling and Solving the Two-facility Capacitated Network Loading Problem, *Operations Research*, 43 (1995), 142–157.

[Mi81] M. Minoux, Optimum Synthesis of a Network with Non-simultaneous Multicommodity Flow Requirements. In *Studies of Graphs and Discrete Programming*, P. Hansen (editor), North-Holland, 269–277, (1981), 269–277.

[Mi00] P. Mirchandani, Projections of the Network Loading Problem, *European Journal of Operational Research*, 122 (2000), 534–560.

[Mi89] P. Mirchandani, Polyhedral Structure of a Capacitated Network Design Problem with an Application to the Telecommunication Industry, Unpublished PhD Dissertation, MIT, Cambridge, MA, 1989.

[Mi96] P. Mirchandani, The Multi-Tier Tree Problem, *INFORMS Journal on Computing*, 8 (1996), 202–218.

[NeWo88] G. L. Nemhauser and L. A. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1988.

[Or91] J. Orlin. Personal Communication, 1991.

[PaJoAlGrWe96] U. Paul, P. Jonas, D. Alveras, M. Grötschel, R. Wessäly, Survivable Mobile Phone Network Architectures: Models and Solution Methods, Preprint SC 96-48, Konrad-Zuse-Centrum für Informationstechnik, Berlin, 1996.

[RaMa97] S. Raghavan and T. L. Magnanti, Network Connectivity, In *Annotated Bibliographies in Combinatorial Optimization*, M. Dell'Amico, F. Maffioli, and S. Martello (editors), John Wiley and Sons, New York (1997) 335–354.

[SoWySéLaGeFo98] P. Soriano, C. Wynants. R. Séguin, M. Labbé, M. Gendreau and B. Fortz, Designing and Dimensioning of Survivable SDH/SONET Networks, In *Telecommunications Network Planning*, B. Sansò and P. Soriano (editors), Kluwer Academic Publishers, The Netherlands, (1998), 147–168.

[StDa94] M. Stoer and G. Dahl, A Polyhedral Approach to Multicommodity Survivable Network Design, *Numerische Mathematik*, 68 (1994),149–167.

[WiGoMiVa95] D. P. Williamson, M. X. Goemans, M. Mihail, and V. V. Vazirani, A Primal-Dual Approximation Algorithm for Generalized Steiner Network Problems, *Combinatorica*, 15 (1995), 435–454.

[Wo84] R. T. Wong, A Dual Ascent Approach for Steiner Tree Problems on a Directed Graph, *Mathematical Programming*, 28 (1984), 271–287.

GLOSSARY FOR CHAPTER 11

alternating path – relative to a matching: a path whose edges alternate between free and matched.

augmenting path$_1$ – relative to a matching: an alternating path that starts at one free vertex and ends at another free vertex.

augmenting path$_2$ P – in a flow network: a directed path from s to t in the residual network.

___, **capacity of** – in a residual network $G_f = (V, E_f, s, t, u_f)$: denoted Δ_P and given by $\Delta_P = \min_{(v,w)\in E_f} r_f(v, w)$.

backward arc (v, w) – across a cut $\langle S, T\rangle$: when $v \in T$ and $w \in S$.

bandwidth – of a communication facility: capacity of a communications facility in bits per second (bps).

blossom: an odd length cycle formed by joining two even vertices of an alternating path, rooted at a free vertex.

circulation – in a cost-flow network $G = (V, A, cap, c, b)$: a flow for the supply vector $b \equiv \mathbf{0}$.

commodity k – in a communication network $G = (N, E)$: a communication *demand* of value d_k (measured in *bits per second (bps)*) between the *origin node* $O(k) \in N$ and *destination node* $D(k) \in N$.

communication facility (or simply *facility*): a transmission medium (e.g., a copper cable, a fiber-optic cable, or a cellular tower-to-satellite connection) installed on a network edge.

communication network: a graph $G = (N, E)$, where N is the set of nodes (or vertices) and E is the set of edges. The nodes denote geographical locations where communication demands originate or terminate, or locations where routing hardware and software is installed. The edges denote potential transmission links.

___, **backbone**: a two-connected network containing high capacity edges and nodes.

___, **local access**: a network, usually a tree, for transmitting the data signal from the backbone network to the end user.

___, **survivable**: a network in which traffic disrupted by a node or edge failure can be rerouted using spare capacity.

cost-flow network $G = (V, A, cap, c, b)$: a directed graph with vertex-set V, arc-set A, a nonnegative capacity function $cap : A \to N$, a linear cost function $c : A \to Z$, and an integral supply vector $b : V \to Z$ that satisfies $\sum_{w\in V} b(w) = 0$.

___, s-t: a flow network $G = (V, A, cap, c, b)$ that contains two distinguished vertices s and t such that $b(v) = 0$ for all $v \in V - \{s, t\}$ and $b(s) = -b(t) > 0$.

___, **extended** s-t $G' = (V', A', cap')$ of $G = (V, A, cap, c, b)$ is an s-t network with vertex-set $V' = V \cup \{s, t\}$, arc-set $A' = A \cup \{(s, v)|b(v) > 0\} \cup \{(w, t)|b(w) < 0\}$ and capacity function cap' defined by

$$cap'(v, w) = \begin{cases} cap(v, w), & \text{if } (v, w) \in A \\ b(v), & \text{if } v = s \\ -b(w), & \text{if } w = t \end{cases}$$

s-t **cut** $\langle S, T\rangle$ – corresponding to a partition (S, T) of V such that $s \in S$ and $t \in T$: the set of arcs that have one endpoint in one of the sets and the other endpoint in the other set; denoted

___, **capacity of**: the sum of the capacities of the arcs crossing the cut in the forward direction, i.e., $cap\langle S, T\rangle = \sum_{(v,w)\in E: v\in S, w\in T} cap(v, w)$.

___, **minimum**: a cut of minimum value, i.e., $\min\{cap\langle S, T\rangle : \langle S, T\rangle$ is an *s-t* cut$\}$.

dual – of a minimum-cost flow problem: see §11.2, Definition 14.

even vertex – relative to an alternating path: a vertex that is an even distance (number of edges) from the root of the path.

excess $e(v)$ – at vertex v: $e(v) = \sum_{(w,v)\in E} f(w, v) - \sum_{(v,w)\in E} f(v, w)$.

facility: a transmission medium (e.g., fibre-optic or wireless) installed on an edge for sending data, voice or video signals.

flow$_1$ f – in an *s-t* flow network $G = (V, E, s, t, cap)$: a function $f : E \to \mathbb{N}$ which obeys three types of constraints:

capacity constraints: $f(v, w) \leq cap(v, w)$, for each arc $(v, w) \in E$.
conservation constraints: $\sum_{(w,v)\in E} f(w, v) = \sum_{(v,w)\in E} f(v, w)$ for each vertex $v \in V - \{s, t\}$.

nonnegativity constraints: $f(v, w) \geq 0$, for each arc $(v, w) \in E$.

flow$_2$ f – in a cost-flow network $G = (V, A, cap, c, b)$: a function $f : A \to Z$ that satisfies

capacity constraints: $f(v, w) \leq cap(v, w)$ for all $(v, w) \in A$,

flow conservation constraints: $\sum\limits_{w} [f(v, w) - f(w, v)] = b(v)$ for each $v \in V$,

nonnegativity constraints: $f(v, w) \geq 0$ for all $(v, w) \in A$.

___, **minimum**: a flow f with minimum $c^T f$ value among all flows.

___, **value of**: the total flow into the sink, i.e., $val(f) = \sum_{(v,t)\in E} f(v, t)$.

flow across cut $\langle S, T\rangle$: the flow crossing the cut in the forward direction minus the flow crossing the cut in the backward direction, i.e., $f\langle S, T\rangle = \sum_{(v,w)\in E: v\in S, w\in T} f(v, w) - \sum_{(v,w)\in E: v\in T, w\in S} f(v, w)$.

s-t **flow network** $G = (V, E, s, t, cap)$: a directed graph with vertex set V and arc-set E, two distinguished vertices, a *source* s and a *sink* t and a nonnegative *capacity* function $cap : E \to \mathbb{N}$.

flow with gains – in a gain network $\bar{G} = (V, A, cap, \gamma, c, s, t)$: a function $f : A \to \mathbb{R}$ that satisfies:

capacity constraints: $f(v, w) \leq cap(v, w)$ for all $(v, w) \in A$,

nonnegativity constraints: $f(v, w) \geq 0$ for all $(v, w) \in A$,

flow conservation constraints:

$$\sum_{w\in V} [f(v, w) - f(w, v)\gamma(w, v)] = 0, \quad \text{for each } v \in V - \{s, t\}.$$

___, **maximum**: a flow with gains that maximizes the amount of flow reaching t given an unlimited supply at s.

___, **minimum-cost maximum-**: a maximum flow with gains that minimizes

$$\sum_{e\in A} c(e)g(e)$$

flow-over-time network: a flow network $G_\mathcal{T} = (V, A, cap, \tau, b)$, where each arc $(v, w) \in A$ has an associated *transit time* τ_{vw}. The transit time τ_{vw} represents the amount of time that elapses between when flow enters arc (v, w) at v and when the same flow arrives at w.

flow-over-time: see §11.2, Definition 26.

forward arc (v, w) – across a cut $\langle S, T\rangle$: when $v \in S$ and $w \in T$.

free edges – of a matching M: the edges of the graph not in M.

free vertices – of a graph G: the vertices of G not incident on a matched edge.

gain network $\bar{G} = (V, A, cap, \gamma, c, s, t)$: a network $G = (V, A, cap, c, b)$ with positive-valued *gain function* $\gamma : E \to \mathbb{R}^+$ and supply function $b_v = 0$ for all $v \in V - \{s, t\}$; the *gain factor* $\gamma(e) > 0$ for arc e enforces that for each unit of flow that enters the arc, $\gamma(e)$ units exit. For standard network flows, the gain factor of every arc is one.

increasable arc (v, w) – in network G for a given flow: $f(v, w) < cap(v, w)$.

matched edges – of a matching M: the edges of M.

matched vertices – of a graph G: the vertices of G incident on a matched edge.

matching – in a graph G: a set of pairwise nonadjacent edges.

___, **complete** – of a bipartite graph $G = (X \cup Y, E)$: a matching that meets each vertex of X; also called *X-saturating*.

___, **maximum-size**: a matching M having the largest size $|M|$.

___, **maximum-weight**: a matching M having the largest weight $wt(M)$.

___, **perfect** – of a graph G: a matching that meets each vertex of G exactly once.

___, **size of**: the number of edges in the matching.

___, **weight of**: the sum of the weights of edges in the matching.

maximum flow problem: given a flow network $G = (V, E, s, t, cap)$, find a flow of maximum value.

maximum multicommodity flow problem: for each commodity i, find a flow f_i of value (demand) $val(f_i)$ such that $\sum_{i=1}^{k} val(f_i)$ is maximized.

multicommodity flow – in a multicommodity flow network $G = (V, E, K, u)$: a set of $k = |K|$ functions $f_i : E \to \mathbb{R}^+$ satisfying the following conditions:

joint capacity constraints: $\sum_{i=1}^{k} f_i(v, w) \leq cap(v, w)$, for each arc $(v, w) \in E$.

conservation constraints: $\sum_{(w,v)\in E} f_i(w, v) = \sum_{(v,w)\in E} f_i(v, w)$ for each vertex $v \in V - \{s_i, t_i\}$ and $i = 1, \ldots, k$, and $\sum_{(w,t_i)\in E} f_i(w, t_i) - \sum_{(t_i,w)\in E} f_i(t_i, w) = d_i$ for each $i = 1, \ldots, k$.

nonnegativity constraints: $f_i(v, w) \geq 0$, for each arc $(v, w) \in E$ and each $i = 1, \ldots, k$.

multicommodity flow network $G = (V, E, K, u)$: a directed graph with vertex-set V and arc-set E, commodity set K, and a nonnegative *capacity* function $u : E \to N$. We adopt the convention that if arc $(v, w) \in E$ then the reverse arc $(w, v) \notin E$. The commodities are indexed by the integers $1, 2, \ldots, k$.

odd vertex – relative to an alternating path: a vertex that is an odd distance (number of edges) from the root of the path.

preflow: a relaxed version of a flow, a function $f : E \to Z^+$ which obeys three types of constraints:

capacity constraints: $f(v,w) \leq cap(v,w)$, for each arc $(v,w) \in E$.

relaxed conservation constraints:

$\sum_{(w,v)\in E} f(w,v) - \sum_{(v,w)\in E} f(v,w) \geq 0$ for each vertex $v \in V - \{s,t\}$.

nonnegativity constraints: $f(v,w) \geq 0$, for each arc $(v,w) \in E$.

reducible arc (v,w) in network G for a given flow: $f(v,w) > 0$.

residual capacity $r_f(v,w)$ – of arc (v,w) in residual network G_f for a given flow or preflow f: see §10.1, Definition 10.

residual network$_1$ – for a maximum-flow network: see §10.1, Definition 10.

residual network$_2$ – for a cost-flow network: see §10.2, Definition 9.

***s-t* cut** $\langle S,T\rangle$ – corresponding to a partition (S,T) of V such that $s \in S$ and $t \in T$: the set of arcs that have one endpoint in one of the sets and the other endpoint in the other set; denoted

___, **capacity of**: the sum of the capacities of the arcs crossing the cut in the forward direction, i.e., $cap\langle S,T\rangle = \sum_{(v,w)\in E: v\in S, w\in T} cap(v,w)$.

___, **minimum**: a cut of minimum value, i.e., $\min\{cap\langle S,T\rangle : \langle S,T\rangle$ is an s-t cut$\}$.

***s-t* flow network** $G = (V,E,s,t,cap)$: a directed graph with vertex set V and arc-set E, two distinguished vertices, a *source* s and a *sink* t and a nonnegative *capacity* function $cap : E \to \mathbb{N}$.

shrunked blossom: obtained by collapsing a blossom into a single vertex.

Steiner tree problem: given a weighted graph in which a subset of vertices are identified as terminals, find a minimum-weight connected subgraph that includes all the terminals.

switch – in a communication network: node equipment for routing and processing communication traffic.

transshipment network: a cost-flow network $G = (V,A,cap,c,b)$ in which all arcs have infinite capacity.

___, **associated**: the transshipment network obtained by replacing each arc $e = (v,w)$ by three arcs (v,x_e), (y_e,x_e), and (y_e,w) having infinite capacity and with costs c_e, 0, and 0, repectively; the supplies at the new nodes x_e and y_e are defined to be $b(x_e) = -cap_e$ and $b(y_e) = cap_e$.

___, **completion of**: the complete transshipment network obtained by adding all missing arcs and giving each of them infinite capacity and arc cost $M+1$, where $M = \sum_e c(e)cap(e)$.

unsplittable flow problem: multicommodity flow problem with the additional restriction that each commodity must be routed on one path.

vertex cover – of a graph G: a set of vertices incident on all edges of G.

weight of a path P – relative to a matching M: the sum of the weights of the free edges in P minus the sum of the weights of the matched edges in P, denoted $wt(P)$.