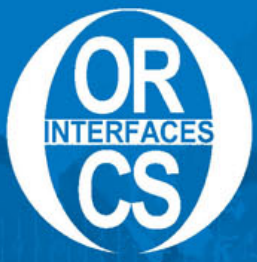
OPERATIONS RESEARCH I COMPUTER SCIENCE INTERFACES

Vasileios Zeimpekis
Christos D. Tarantilis
George M. Giaglis
Ioannis Minis
Editors



Dynamic Fleet Management

Concepts, Systems, Algorithms & Case Studies

Springer

DYNAMIC FLEET MANAGEMENT
Concepts, Systems, Algorithms & Case Studies

OPERATIONS RESEARCH/COMPUTER SCIENCE INTERFACES SERIES

Professor Ramesh Sharda
Oklahoma State University

Prof. Dr. Stefan Voß
Universität Hamburg

Greenberg /*A Computer-Assisted Analysis System for Mathematical Programming Models and Solutions: A User's Guide for ANALYZE*
Greenberg / *Modeling by Object-Driven Linear Elemental Relations: A Users Guide for MODLER*
Brown & Scherer / *Intelligent Scheduling Systems*
Nash & Sofer / *The Impact of Emerging Technologies on Computer Science & Operations Research*
Barth / *Logic-Based 0-1 Constraint Programming*
Jones / *Visualization and Optimization*
Barr, Helgason & Kennington / *Interfaces in Computer Science & Operations Research: Advances in Metaheuristics, Optimization, & Stochastic Modeling Technologies*
Ellacott, Mason & Anderson / *Mathematics of Neural Networks: Models, Algorithms & Applications*
Woodruff / *Advances in Computational & Stochastic Optimization, Logic Programming, and Heuristic Search*
Klein / *Scheduling of Resource-Constrained Projects*
Bierwirth / *Adaptive Search and the Management of Logistics Systems*
Laguna & González-Velarde / *Computing Tools for Modeling, Optimization and Simulation*
Stilman / *Linguistic Geometry: From Search to Construction*
Sakawa / *Genetic Algorithms and Fuzzy Multiobjective Optimization*
Ribeiro & Hansen / *Essays and Surveys in Metaheuristics*
Holsapple, Jacob & Rao / *Business Modelling: Multidisciplinary Approaches — Economics, Operational and Information Systems Perspectives*
Sleezer, Wentling & Cude/*Human Resource Development And Information Technology: Making Global Connections*
Voß & Woodruff / *Optimization Software Class Libraries*
Upadhyaya et al / *Mobile Computing: Implementing Pervasive Information and Communications Technologies*
Reeves & Rowe / *Genetic Algorithms—Principles and Perspectives: A Guide to GA Theory*
Bhargava & Ye / *Computational Modeling And Problem Solving In The Networked World: Interfaces in Computer Science & Operations Research*
Woodruff / *Network Interdiction And Stochastic Integer Programming*
Anandalingam & Raghavan / *Telecommunications Network Design And Management*
Laguna & Martí / *Scatter Search: Methodology And Implementations In C*
Gosavi/ *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*
Koutsoukis & Mitra / *Decision Modelling And Information Systems: The Information Value Chain*
Milano / *Constraint And Integer Programming: Toward a Unified Methodology*
Wilson & Nuzzolo / *Schedule-Based Dynamic Transit Modeling: Theory and Applications*
Golden, Raghavan & Wasil / *The Next Wave in Computing, Optimization, And Decision Technologies*
Rego & Alidaee/ *Metaheuristics Optimization via Memory and Evolution: Tabu Search and Scatter Search*
Kitamura & Kuwahara / *Simulation Approaches in Transportation Analysis: Recent Advances and Challenges*
Ibaraki, Nonobe & Yagiura / *Metaheuristics: Progress as Real Problem Solvers*
Golumbic & Hartman / *Graph Theory, Combinatorics, and Algorithms: Interdisciplinary Applications*
Raghavan & Anandalingam / *Telecommunications Planning: Innovations in Pricing, Network Design and Management*
Mattfeld / *The Management of Transshipment Terminals: Decision Support for Terminal Operations in Finished Vehicle Supply Chains*
Alba & Martí/ *Metaheuristic Procedures for Training Neural Networks*
Alt, Fu & Golden/ *Perspectives in Operations Research: Papers in honor of Saul Gass' 80th Birthday*
Baker et al/ *Extending the Horizons: Adv. In Computing, Optimization, and Dec. Technologies*

DYNAMIC FLEET MANAGEMENT

Concepts, Systems, Algorithms & Case Studies

Edited by

Vasileios Zeimpekis
Christos D. Tarantilis
George M. Giaglis
Ioannis Minis

Springer

Vasileios Zeimpekis
Athens University of Economics & Business
Athens, Greece

Christos Tarantilis
Athens University of Economics & Business
Athens, Greece

George M. Giaglis
Athens University of Economics & Business
Athens, Greece

Ioannis Minis
University of Aegean
Chios, Greece

Library of Congress Control Number: 2007924349

ISBN-13: 978-0-387-71721-0 e-ISBN-13: 978-0-387-71722-7

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

TABLE OF CONTENTS

PREFACE

The challenges of contemporary fleet management are moving beyond cost efficiency towards superior customer service, agility, and responsiveness to requirements that vary at a time scale unthinkable even a decade ago. Over the last forty years classical methods of fleet management have addressed extensively the issue of cost efficiency by developing *a priori* routing plans in a wide spectrum of practical problems. However, the use of an initial plan, although necessary, is by no means sufficient to address events that are likely to occur during plan execution and significantly affect system performance. Typical examples of such evens are customer orders that arrive in real time and should be served by vehicles already on route, as well as disturbances intrinsic to urban environments, such as traffic delays, parking unavailability, and breakdowns. The ability to deal with such cases in a satisfactory manner is increasingly important to the competitiveness of logistics and transport related operations.

Dynamic fleet management refers to environments in which information is dynamically revealed to the decision maker. This information may not be known at the initial planning stage, and/or may change after the construction of the initial fleet routes during plan execution. In addition, there are significant cases in which no routing exists and the system responds to requests that arrive dynamically.

Methods that address the critical issues of dynamic fleet management may be implemented in practical systems by taking advantage of recent advances in satellite and mobile communication technologies. Specifically, satellite location identification systems that use the Global Positioning System (GPS) and terrestrial mobile communication systems, such as the General Packet Radio Service (GPRS) or Terrestrial Trunked Radio (TETRA), enable fleet operators to monitor the execution of a plan and to manage operations in real time, thus improving fleet performance.

This edited volume aims to highlight important advances in the emerging field of Dynamic Fleet Management. The fundamental problem of real time vehicle routing is defined and solution methods are presented and classified. Emphasis is also given to algorithmic approaches that are able to process dynamic information and produce solutions of acceptable quality for significant dynamic fleet management problems in almost real time. Finally, the volume includes case studies that address actual dynamic problems by combining systemic and algorithmic approaches.

The first three chapters survey important aspects of the dynamic vehicle routing problem.

Chapter 1 reviews and classifies solution methodologies that address real-time vehicle routing problems where customer requests are dynamically revealed over time. Each service request either has a combined pick-up and delivery location or only a single pick-up (or delivery) location. Different algorithmic methodologies are presented that handle the occurrence of new requests through the construction of the part of the route that has not yet been executed by the vehicle (i.e. planned route). To construct the planned routes, adaptations of methods originally employed to solve the static problem are presented. Issues of diverting a vehicle away from its current planned destination to serve a new request that has just occurred are then discussed. Solution methodologies that anticipate future requests to efficiently satisfy future demands are also reviewed. Chapter 1 concludes by proposing future directions for research, such as the development of solutions approaches for handling the occurrence of vehicle breakdowns and unexpected congestion, formal modeling frameworks that integrate the uncertainty associated with future requests, as well as other theoretical and practical issues of research.

Chapter 2 discusses important characteristics and properties of the dynamic vehicle routing problem from a temporal point of view. Differences between the static and the dynamic vehicle routing problems are also demonstrated by analyzing critical issues from previous published papers. The importance of measuring the performance of a dynamic vehicle routing system is then highlighted, and measures for dynamism in systems with and without time windows are discussed. Methods for evaluating the performance of on-line routing algorithms are presented and important issues to include in the system objective are reported. A three-echelon classification of dynamic vehicle routing systems is also proposed based on a) their degree of dynamism and b) the objective of the system. Finally, Chapter 2 emphasizes the significance of considering the volume and the temporal composition of immediate requests along with the system objective when developing an algorithmic methodology for a dynamic vehicle routing system.

Chapter 3 discusses the experience gained in practical issues in stochastic and dynamic routing in the context of developing a VRP solver at a Norwegian research institute. First, a review of the literature on dynamic and stochastic vehicle routing problem (DSVRP) is presented. To illustrate the need for dynamic and stochastic models in real world applications, two examples involving transportation of goods and persons, respectively, are demonstrated. Modelling and formal description of DSVRPs is also proposed to create the platform upon which new computational methods will be tested and evaluated. Chapter 3 also considers the context in which a VRP solver operates and proposes solution approaches based on scenario generation. The way to exploit dynamic events to produce more robust plans to the VRP is discussed, as well as the role of generating statistical knowledge of events

automatically from past experience. Chapter 3 concludes with suggestions for further research in these areas.

Chapters 4 to 6 provide algorithmic approaches of significant applicability to practical dynamic fleet management problems.

Chapter 4 proposes a dynamic programming-based approach to address a general fleet dispatching problem. In this problem the vehicles are dispatched to serve load requests, which arise randomly during each time period of a finite time horizon at different locations in a transportation network. The fleet comprises vehicles of multiple types. An additional critical (and practical) complication is that the travel times between the network nodes are random. The approach presented in this chapter uses modelling and methodological concepts from the deterministic travel time case to present a novel approach for random travel times. Major contributions of this chapter include the formulation of the problem as a dynamic program; the way of approximating the value function of the resulting subproblem for each time period by separable piecewise linear concave functions; the proof that the approximate subproblem is a min-cost network flow problem; the further decomposition of the latter to multiple problem instances by location, which can be solved in parallel. Furthermore, an updating method is employed to improve the value function approximations, and a comprehensive algorithm is proposed to obtain solutions of superior quality, as evidenced by the experimental results of three classes of problems included in this work. Chapter 4 concludes by proposing challenging new opportunities for research in the dynamic fleet management area, such as the introduction of load pick up and delivery windows when the load requests arrive randomly, as well as other unresolved practical issues.

Chapter 5 proposes a column generation-based approach to plan the operations of an on-demand air transport system. The problem consists of determining the fleet assignment, aircraft routing and crew pairing in an integrated fashion for a system that provides point-to-point service at customer request. The dynamic elements of the problem are twofold: i) The demand for service is not known in advance, and is dynamically received. ii) There are unscheduled maintenance requirements that are also raised dynamically. The proposed model is based on a three-day planning period within a rolling horizon setting. It uses a crew duty network and a fleet-station time line in order to embed the crew and aircraft information in the fleet assignment problem, while keeping the crew and aircraft separate during planning. In addition to the model, the major contributions of this chapter include: The use of column generation and identification of good pairings by solving special shortest path problems; the dynamic adjustment of the plan when new requests for service or unscheduled maintenance are revealed without relying on demand forecasts; the comparison between cases with fixed (immovable) requests, and cases in which (limited) freedom is

given in satisfying a request. Experimental results from practical cases indicate the ability of the model and the proposed approach to deal effectively and on-time with the dynamic nature of demand requests in a realistic setting.

Chapter 6 addresses a transportation problem with time varying parameters. Specifically, it models and solves the problem of determining optimal paths in a transportation network with time dependent link costs and travel times. In addition, multiple modes of transport are considered, and the related transfer delays and costs are also time dependent and fully accounted for. The problem is solved to optimality by a minimum cost path algorithm that computes optimum path trees from all network nodes and feasible discrete departure times. The algorithm has been applied to intermodal routing in the case of hazardous materials transport. In this case the risks associated with mode-link combinations and transhipments are also time dependent, and the problem has been formulated in a way amenable to the proposed algorithm. It should be noted that due to its computational efficiency, the algorithm could be applied to dynamic problems that account for real time system changes.

Finally, Chapters 7 to 10 discuss real-life applications and case studies of dynamic vehicle routing and fleet management, demonstrating the applicability and practical significance of research in the area.

Chapter 7 introduces the need for real-time fleet management in emergency response situations. The authors propose an integrated emergency response fleet deployment system that embeds an optimization approach to assist dispatchers in assigning emergency vehicles to emergency calls, while having the capability to look ahead for future demands. The proposed system is tested and validated by means of a simulation model and a case study application in the area of Washington, DC. Moreover, a mathematical model for real time vehicle dispatching is presented and it is shown that its exact solution, minimizing the expected total wait over a large network, can be obtained with a short computation time.

Chapter 8 addresses the important area of City Logistics and reports on a DSS-based modelling framework aiming at supporting the design and evaluation of city logistics applications prior to their implementation. The decision support system draws on an underlying dynamic traffic simulation model that feeds a dynamic router and scheduler, which can then determine which vehicle to assign to new services as well as the new route for the selected vehicle. Further to the presentation of the proposed DSS, the chapter also discusses two case studies performed in the Italian cities of Lucca and Piacenza to illustrate how the system works in practice.

Chapter 9 also focuses on city logistics and discusses the design and implementation of a real-time fleet management system capable of rerouting

vehicles in real time when unforseen events, such as breakdowns or delayed vehicles that cannot meet future customer time windows, occur during urban freight distribution. The vehicle-mounted wireless communication sub-system monitors each vehicle through GPS-based positioning that is reported to the dispatch centre via GPRS. The dispatch centre utilizes this information to monitor the fleet, detect deviations from the initial distribution plan, and adjust the schedule accordingly by suggesting effective rerouting interventions. The chapter discusses the application of the system in one case study of a Greek 3PL operator, demonstrating the degree of customer service improvement that can be achieved through real time vehicle monitoring and rerouting.

Chapter 10 describes a real-time fleet management system designed and implemented for eCourier Ltd at London, UK. The chapter reports the overall system architecture, the main algorithms, the travel time forecasting procedure, and the job allocation heuristic used. The system is capable of monitoring courier location information and vehicle type, among other variables, in real-time. This information is fed to a set of algorithms that allocate each job to the most appropriate courier on the basis of road congestion and current fleet status, as well as individual courier efficiency. Courier location information is provided by GPS devices embedded into palmtop computers which are also used to provide directions to couriers. Results of system operation in real-life demonstrate its ability to reduce the requirements for human fleet management supervisors, to improve service and to increase courier efficiency.

ACKNOWLEDGMENTS

Preparing an edited volume is an exciting task based on collaboration and support by esteemed colleagues and co-workers. We owe gratitude to all authors who contributed their work on state-of-the-art methods and results related to dynamic fleet management.We would also like to express our appreciation to the chapter referees for their invaluable help in ensuring the quality standards of this volume. Specifically, we would like to thank: N. Altay, N. Ampazis, J. Barceló, E. Benavent, I. Benyahia, E. Chang, H.K. Chen, A. Corberan, L. Coslovich, G. Dounias, T. Fahle, G. Ghiani, A. Haghani, G. Hasle, J. Herrmann, S. Ichoua, G. Ioannou, B. Kallehauge, J. Q. Li, E. Manni, M. Montemanni, E. Mota, R. Nagi, R. Pesenti, J.-Y. Potvin, D. Pisinger, H. Psaraftis, M. Reimann, E. Taniguchi, P. Tsilingiris, B.W. Thomas, H. Topaloglu, S. Yang, Y. Yao, A. Ziliaskopoulos, and P. Zito. This volume would not be possible without the input, guidance and support of Gary Folven, editor-in-chief of the Operations Research stream in Springer Verlag and Carolyn Ford, editor assistant. Finally, many thanks are due to L. Amygdalou, G. Ninikas and T. Athanasopoulos for their support in editing and preparing the overall manuscript.

V. Zeimpekis
C.D. Tarantilis
G.M. Giaglis
I. Minis

Athens and Chios, March 2007

Chapter 1

PLANNED ROUTE OPTIMIZATION FOR REAL-TIME VEHICLE ROUTING

Soumia Ichoua[1], Michel Gendreau[2] and Jean-Yves Potvin[2]
[1]Département d'opérations et systèmes de décisions and Centre de recherche sur les technologies de l'organisation réseau, Pavillon Palasis-Prince, Université Laval, Québec, Canada, G1K 7P4; [2]Département d'informatique et de recherche opérationnelle and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succ. Centre-Ville, Montréal, Canada, H3C 3J7

Abstract: This paper reviews and classifies the work done in the field of dynamic vehicle routing. We focus, in particular, on problems where the uncertainty comes from the occurrence of new requests. Problem-solving approaches are investigated in contexts where consolidation of multiple requests onto the same vehicle is allowed and addressed through the design of planned routes. Starting with pure myopic approaches, we then review in later sections the issues of diversion and anticipation of future requests.

Keywords: real-time, vehicle routing, planned routes, diversion, anticipation.

1.1 INTRODUCTION

The field of real-time fleet management has steadily grown over the past few years. This increased interest comes from recent economical and technological developments, where modern economy markets tend to become increasingly open and competitive. Companies now need to focus on timeliness to insure not only their competitiveness, but also their survival. A key element in achieving this goal is the elaboration of efficient, just-in-time, distribution systems where goods are delivered at the right place, in the right quantity and exactly when needed. The availability of real-time information (e.g., vehicle position, traffic conditions, etc.) is thus critical. Fortunately, the rapid growth in communication and information technologies now provide opportunities for obtaining real-time information at lower costs.

As these inputs also need to be processed under stringent time limitations, a challenging issue is the elaboration of efficient solution approaches that integrate real-time information, while satisfying the time limitations that are inherent to continuously evolving environments.

In this paper, we are interested in dynamic vehicle routing and dispatching problems, which can be broadly stated as follows. We have a fleet of vehicles in movement to service customer requests that are dynamically revealed over time. The service is realized under various operational constraints such as time windows and limited vehicle capacity. Apart from new customer requests, other types of dynamic events can also occur like dynamic travel times, service cancellations, vehicle breakdowns, etc. Hence, decisions must be made in a changing environment. This is to be opposed to the static case where all data are known in advance and do not change afterward.

Solution quality typically relates to operations costs and revenues, like the total number of served requests or the total distance traveled by the vehicles, as well as service quality, like the total lateness at customer locations. Numerous applications for these real-time problems can be found in practice, for example dial-a-ride systems for transportation-on-demand, courier services, emergency services, pick-up and delivery of goods, and many others. General considerations, special journal issues and surveys about these problems can be found in Cordeau *et al*. (2004), Desrosiers *et al*. (1995), Gendreau and Potvin (1998, 2004), Ghiani *et al*. (2003), Ichoua (2001), Powell *et al*. (1995), Psaraftis (1988, 1995) and Séguin *et al*. (1997).

A first distinction can be made among these problems based on their degree of dynamism. The latter can be defined along two dimensions:

- *frequency of changes*. The degree of dynamism is higher when new service requests are more frequent and/or their attributes (e.g., demand, time windows) are prone to more frequent changes over time.
- *urgency*. The latter depends on the response time, which can be defined as the delay between the request arrival time and the time of beginning of service.

Examples of problems with a low degree of dynamism include transportation-on-demand for the elderly or the disabled, where most requests are static and where a few additional dynamic requests are known a fairly long time before their actual service. On the other hand, courier services in urban areas and emergency services are highly dynamic.

Another distinction can be made between problems with consolidation (e.g., dial-a-ride, less-than-truckload trucking) or without consolidation (e.g., truckload trucking). In the first case, many customer requests can be consolidated onto the same vehicle. Thus, routing issues arise and the need to adequately sequence the requests within planned routes becomes crucial.

The latter can be defined as the sequence of requests that have already been received and assigned to a vehicle, but that have not been serviced yet. The sequence can be based, for example, on the request arrival times. Figure 1-1 illustrates a vehicle route in a dynamic setting. This route is divided into three parts at any instant *t*:

- *completed movements* that correspond to the part of the route that has already been executed. This part cannot be modified anymore;
- *current movement* to reach the next destination;
- *planned movements* which correspond to the part of the route that has not yet been executed by the vehicle (planned route).

In Figure 1-1, the black square stands for the central depot and the little white circles are customer requests. The completed movements correspond to the arcs with broken lines, the current movement corresponds to the thick arc and the planned movements to regular arcs. Thus, customers 1 and 2 have already been served, customer 3 is the vehicle's current destination (the vehicle is moving between customers 2 and 3) and customers 4 and 5 are on the planned route. A planned route can be used, for example, to decide about the next destination of a vehicle or to decide about the acceptance or rejection of a new request.

In problems without consolidation, a vehicle is dispatched to serve a single customer. These problems often arise in situations where the travel time between two service locations is large (e.g., wide area truckload trucking) or where the degree of dynamism is high (e.g., emergency services). In these cases, there is no need for planned routes and the problem is rather of the assignment type. Then, repositioning an idle vehicle after service completion, in anticipation of future requests, becomes a challenging issue.

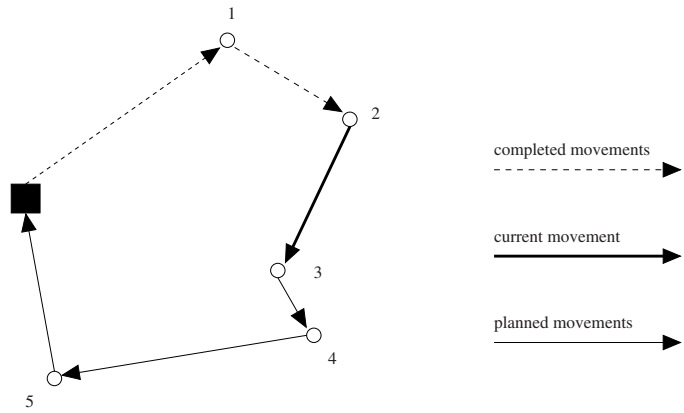


Figure 1-1. A vehicle route in a dynamic setting.

In this paper, we are interested in solution approaches for dynamic vehicle routing problems where the uncertainty comes from the occurrence of new service requests and where consolidation is allowed and handled through planned routes. The remainder of the paper is as follows. Section 1.2 addresses the construction of planned routes through adaptations of algorithms originally developed for the static case. Different algorithmic approaches reported in the literature, with a particular emphasis on meta-heuristics, will be reviewed. Section 1.3 discusses the issue of diverting a vehicle away from its current destination to serve a new request. Simple insertion strategies and reoptimization approaches will be reviewed. Section 1.4 examines the issue of anticipating future requests in the routing plans to allow future demands to be met more efficiently. Finally, section 1.5 concludes and proposes future avenues of research.

1.2 ADAPTATION OF STATIC ALGORITHMS

One common approach for constructing planned routes in a dynamic context is to exploit algorithms that have already been developed for the static case. Basically, the algorithm is applied on the static problem, as defined by known requests, each time an input update occurs. This optimization is usually realized over a rolling time horizon, where long-term events, which are likely to lead to useless calculations, are postponed (Psaraftis *et al.*, 1985). In fact, as the length of the rolling horizon increases, the problem becomes richer and contains more requests, but can be difficult to handle unless a fast and powerful optimization procedure is available.

In the following, we distinguish problems of the many-to-many and many-to-one (one-to-many) type. In the first case, each request has both a pick-up and a delivery location while, in the second case, each request has only a single pick-up (delivery) location. Problems of the many-to-many type are more challenging because the pick-up and delivery locations must be served by the same vehicle and the pick-up must be performed before the delivery. These characteristics typically lead to distinct algorithmic solutions.

Adaptations of static algorithms reported in this paper can be divided into three major classes: fast local update procedures, reoptimization procedures and hybrid approaches. Typically, some static algorithm is first applied over requests that are known at the start of the day, if any, to construct an initial set of routes. Then, as the working day unfolds, a fast local update procedure can be used to integrate newly occurring requests into these routes. These local procedures are also useful when a dispatcher must quickly tell a customer if his request can be accommodated or not. Although simple and easy to implement, they are also inherently myopic and do not fully exploit

the capabilities of modern computers. Reoptimization procedures are more computationally intensive because they reconsider all known requests within some rolling time horizon and resequence them. Their computational requirements might prevent their use in highly dynamic environment, although some control is possible by varying the length of the rolling horizon. To take advantage of the speed of local update procedures and the power (in terms of solution quality) of reoptimization procedures, many researchers have combined them to produce hybrid schemes. Typically, fast heuristics based on simple insertions are first used to quickly add a new request into the current solution. Then, a more sophisticated procedure is run, until the occurrence of the next event, to improve this initial solution. It is also possible to accumulate many requests over a given time interval, to insert them all at once into the current solution and to run the reoptimization procedure over the next time interval. These approaches are reviewed in the following.

1.2.1 Many-to-one (One-to-many) Problems

Given that a single location is associated with each customer request, the problems in this class are of the following types:

- variants of the traveling salesman problem (TSP), like the dynamic traveling repairman problem (DTRP); the latter typically involves a utility firm (electricity, gas, water and sewer) that responds to customer requests for repair or maintenance of its facilities at the customer premise,
- courier service problems where parcels and mail are collected at various locations and brought back at a central depot for further processing,
- delivery systems for various types of products and goods,
- feeder systems where, for example, people are taken by mini-buses and transported to a train station.

1.2.1.1 Local update procedures

In the academic formulation of the DTRP, one or more vehicles are used to serve a set of independently and uniformly distributed customers that occur over time according to a Poisson process. In this problem, the service time at each customer is a random variable and is typically an important part of the total route time. In Bertsimas and van Ryzin (1991), simplifying assumptions allow a tractable mathematical analysis of various routing policies for the DTRP aimed at minimizing the expected time spent in the system by each customer. This type of work is strongly inspired by queuing theory where a vehicle is viewed as a mobile server. Accordingly, no explicit

planned routes are constructed (although the routes can be traced back a posteriori). Rather, routing policies are defined to identify the customer to be served next, among a queue of pending requests. Bertsimas and van Ryzin (1991) analyze simple policies like First-Come First-Served and Nearest Neighbor in light and heavy traffic conditions. Later, in Bertsimas and van Ryzin (1993), the same authors extend their findings to the case of a fleet of homogeneous vehicles. With many vehicles, the authors first partition the service area into sub-regions and then apply the results of their first paper to each sub-region. Larsen *et al*. (2002) address another variant of DTRP that involves both advanced (static) and immediate (dynamic) requests. The authors empirically assess the performance of some policies reported in Bertsimas and van Ryzin (1991) under varying degrees of dynamism. Their results show that the route length increases linearly with the degree of dynamism and that Nearest Neighbor performs better on average than the other tested policies.

As a thorough mathematical analysis is seldom possible for real-world applications, ad hoc rules are often used in practice to tell the driver what his next destination should be. This can also be achieved through the construction of planned routes that allow decisions to be made with regard to all other known requests. If the environment is not highly dynamic, the planned routes are not likely to change much over time and can also be used for later decisions involving the same vehicle or other vehicles. This approach is used, for example, in Madsen *et al*. (1995b) for the repair of gas installations, where the authors propose a simple insertion heuristic. When a new request is received, a subset of routes is first selected, based on a proximity measure, and the feasible insertion position that minimizes the detour over that subset of routes is chosen for the new request. The real-time requirement comes from the need to quickly tell the customer if his request can be accepted and to specify a time window, chosen among a prespecified set of time slices, within which the service crew will arrive at the customer premise.

1.2.1.2 Reoptimization procedures

With regard to more powerful reoptimization procedures, the first implementations date back to the early 80's for commercial delivery systems. Bell *et al*. (1983) address the problem of routing and scheduling a fleet of vehicles delivering bulk products stored at a central depot. Many constraints were considered such as time windows at customers, vehicle capacity and compatibility constraints between products and vehicles. A static algorithm previously reported in Fisher *et al*. (1982) was applied once a day to determine the schedules for the next two to five days. The static

algorithm was based on a mixed integer programming model. The latter was solved through Lagrangian relaxation combined with a multiplier adjustment method.

Brown *et al*. (1987) consider the problem of dispatching petroleum tank trucks to satisfy customer requests under various constraints. The authors repeatedly apply an assignment and routing heuristic on known requests within a rolling horizon. The heuristic first assigns the loads to available vehicles and then solves a traveling salesman problem to optimize each route. A similar problem is addressed in Bausch *et al*. (1995). The reported heuristic first generates clusters of customers for each vehicle type. The total distance traveled is then optimized within each cluster using either a heuristic or an exact algorithm, depending on the problem size.

A dynamic vehicle routing problem (with no time windows) is considered in Montemanni *et al*. (2005) and Gambardella *et al*. (2003). Here, static VRPs are solved over time with an ant colony system algorithm. One interesting feature is that useful information about the solutions produced is transferred from one problem to the next through a pheromone conservation mechanism. Also, the horizon is divided into fixed time slots, as in Kilby *et al*. (1998), and new orders received during the current time slot are considered only at the end of that time slot. As the optimization algorithm runs on the current static problem for the duration of a time slot, it is easy to control the computation time allocated to each static problem.

Gendreau *et al*. (1999) use a hybrid approach to solve a dynamic vehicle routing problem with time windows motivated from the local operations of long-distance courier companies. First, an insertion heuristic is used to insert any newly occurring request. Then, a tabu search with an adaptive memory (Rochat and Taillard, 1995) is applied to this initial solution to improve it until the occurrence of the next event. The neighborhood structure is based on CROSS exchanges. Basically, two segments of variable length are taken from two different routes and swapped. The adaptive memory in this algorithm is used as a repository of elite solutions. New starting solutions for the tabu search can then be obtained by combining routes taken from different solutions in this memory. A new starting solution or set of routes is first partitioned into subsets of routes (sub-problems) through a sweep procedure. Then, a different tabu search heuristic is applied to each sub-problem, to provide a form of intensification. To reduce the wall-clock time, a two-level parallel scheme is also reported. First, different tabu search threads run in parallel. Second, within each search thread, many tabu searches are run independently on the sub-problems obtained through the partition procedure. Two types of events lead to the interruption of the tabu searches, namely: the occurrence of a new service request, as the latter needs

to be integrated into the current routes, and service completion at a customer location, as the driver needs to be told about his next destination.

1.2.2 Many-to-many Problems

In these problems, a pick-up and a delivery location are associated with each customer request. These problems are mostly found in the following applications:

- local express mail services in urban areas,
- dial-a-ride systems for transportation-on-demand services,
- less-than-truckload applications where different kinds of goods and products are picked up and delivered.

1.2.2.1 Local update procedures

Early work in this area was motivated from real-world applications and was based on simple insertion heuristics to quickly integrate new requests into the planned routes. The insertion mechanism was modified to handle both a pick-up and a delivery location, see Rousseau and Roy (1988) for express mail services and Madsen *et al*. (1995a), Roy *et al*. (1984), Wilson and Colvin (1977), for dial-a-ride systems. Also, in Swihart and Papastavrou (1999), the authors have extended the routing policies of Bertsimas and van Ryzin (1991) to dispatch requests with a pick-up and a delivery in a DTRP context.

1.2.2.2 Reoptimization procedures

A dynamic single-vehicle dial-a-ride problem was first addressed by Psaraftis (1980) with an exact algorithm. Based on a finite time horizon, a series of static problems were solved through a dynamic programming algorithm. An optimal solution was obtained in reasonable computation time due to the small size of each static problem. In the dynamic programming algorithm, the state vector $(L, k_1, k_2, \ldots)$ is defined as follows:

- L is the current vehicle location ($L=0$ at the depot, $L=i$ at the pick-up location of customer i and $L=i+n$ at the delivery location of customer i).
- k_i is the status of customer i ($k_i=3$ if i has not been picked-up yet, $k_i=2$ if i has been already picked-up but has not been delivered yet and $k_i=1$ if i has already been delivered)

Later, Psaraftis (1983) generalized his approach to account for time window constraints. A similar approach is used in Caramia *et al*. (2002) for a multi-cab metropolitan transportation system. Basically, a new request is

first inserted in one of the planned routes and this route is then sequenced optimally using a dynamic programming algorithm. Here also, this approach is possible due to the limited capacity of each cab which leads to small routes.

In Krumke *et al*. (2002), the authors address the problem faced by a German automotive club that maintains a fleet of vehicles to help people with a car breakdown. The problem corresponds to a multi-depot vehicle routing problem with time windows which needs to be solved under strict time restrictions, as new help requests continuously occur over time. The authors solve a sequence of static problems over known requests with a custom-made exact column generation method. The latter can be stopped before completion and shows good performance within only a few seconds of computation time. In Savelsbergh and Sol (1998), both approximation and incomplete exact optimization techniques are considered to obtain a good trade-off between response time and solution quality. The motivation for this work comes from the activities of a large company providing nation-wide transportation services for various types of packages.

In the work of Rivard (1981) simple insertions and reoptimization procedures are both used to solve a dial-a-ride problem motivated from transportation-on-demand services for the disabled. At fixed time periods, vehicle routes are reoptimized through the following procedure, where M is a parameter.

For each vehicle route R and for each request $i \in R$:

1. Remove i from R and update the pick-up and delivery times of the other requests in R.
2. Select the M best routes according to the following, least-disruptive, criterion: smallest total deviation of pick-up and delivery times along the route after the insertion of i.
3. Among these M routes, insert request i in the best route according to a least-cost criterion.

In Gendreau *et al*. (1998), the tabu search heuristic with adaptive memory reported in Gendreau *et al*. (1999) is extended to solve a local express courier service problem. Basically, the neighborhood structure is modified to handle requests with both a pick-up and a delivery location. This neighborhood is based on ejection chains (Glover, 1996), where a request is moved from one route to another and, in the process, might eject a request from that route (due to constraint violations, for example). The ejected request is then moved to yet another route, etc. The chain ends when the insertion of a request in a route does not lead to any ejection. A chain might be of any length and might be cyclic or not. A constrained shortest path problem is defined and solved to find the best possible ejection chain in the neighborhood.

A similar approach for a dynamic dial-a-ride problem is used in Attanasio *et al*. (2004). Here, the authors use a parallel implementation of a tabu search heuristic previously reported in Cordeau and Laporte (2003) for the static version of the problem. Whenever a new service request occurs, an insertion heuristic is first applied to know if the request can be accepted or not. Then, the tabu search is applied using a neighborhood structure where requests are moved from one route to another. In Angelelli *et al*. (2004), the authors address a courier application where the service area is divided into geographical zones, with a central hub in each zone. When the destination zone of a request is different from its origin, the transshipment from one hub to another is made overnight. A variable neighborhood search based on the extraction and insertion of a request is used to solve the problem.

1.2.3 Multiple Plan Approach (MPA)

To provide more robustness to the solution procedure, Bent and van Hentenryck (2004) propose a problem-solving framework, called MPA, where multiple routing plans that are consistent with the current state of information are maintained. A routing plan is defined as a set of vehicle routes that serve known requests. At each iteration, all plans in the pool are updated to be consistent with the distinguished plan that is followed until the next event. The latter is not necessarily the minimum cost plan, but rather the plan that is most similar to the others in the pool (to limit the amount of disruption). This problem-solving framework generalizes and abstracts from a particular search methodology the procedure reported in Gendreau *et al*. (1999) where an adaptive memory containing many solutions is used to feed a tabu search. In both papers, the experimental results demonstrate the benefits of a multiple plan approach over single-plan approaches.

It is well known that skilled human dispatchers use their experience and some knowledge about their customers to make decisions. In the following sections, we address two issues related to the practice of expert dispatchers: diversion and anticipation of future requests.

1.3 DIVERSION

Diversion is an interesting avenue that has seldom been addressed in the literature, except in Ichoua *et al*. (2000) and Regan *et al*. (1995). It consists of diverting a vehicle away from its current planned destination to serve a request that has just occurred in its vicinity. Exploiting diversion opportunities is now possible due to recent advances in communication and information technologies (e.g., cellular phones, global positioning systems, etc.). However, integrating diversion into a solution approach raises a

number of issues that need to be carefully addressed. That is why most problem-solving methods reported in the literature assume that the next destination of a vehicle is fixed. To the best of our knowledge, Regan *et al*. (1995) were the first to explicitly address diversion, although in a truckload context where no consolidation takes place. The authors empirically assessed the benefits of diversion under various demand arrival patterns, load acceptance and dispatching rules.

The work of Ichoua *et al*. (2000) propose a broader view of diversion in the context of a long-distance courier service where parcels are collected in a local area and brought back to a central office for further processing. Addressing diversion in this context is more challenging, as consolidating and sequencing the requests becomes an important issue. Furthermore, the response time requirements are more stringent (in contrast with truckload carrier applications which take place over wide geographical areas and where the time horizon is longer). Ichoua *et al*. (2000) propose a more general strategy where allowing diversion might lead to redirecting one or more vehicles. To assess its effectiveness, the new strategy was integrated into the parallel tabu search heuristic of Gendreau *et al*. (1999). Whenever a new request occurs, it is first inserted at its best feasible insertion place in the current set of routes, including the point between the current vehicle position and its planned destination (which corresponds to classical diversion). Then, the tabu search improves this solution and, in the process, is free to move any request between the current location of a vehicle and its planned destination. In other words, new opportunities are offered by considering the current vehicle position, instead of its next destination, as the starting point of the planned route. Thus, at the end of the optimization procedure, the destination of one or more vehicles might have changed but not necessarily to serve the new request. In fact, the latter might appear anywhere in a planned route and classical diversion is only one possible outcome. Although a tabu search is proposed in this work, other kinds of optimization methodologies could have been used as well.

Since diversion is applied in a highly dynamic context (vehicles are moving fast and diversion opportunities can be quickly lost), a limited amount of time Δt is allocated to the optimization procedure. In particular, when a new request is received at instant t, the optimization is performed on the solution obtained by projecting the current routing plan at instant $t+\Delta t$, when the results of the optimization procedure will be known and can be applied. Different rules are proposed to set Δt to obtain a good trade-off between computation time and solution quality. Basically, if Δt is too large, diversion opportunities can be lost; conversely, if Δt is too small, solution quality might suffer.

1.4 ANTICIPATION OF FUTURE REQUESTS

Human dispatchers typically have some valuable knowledge about demand patterns in space and time (e.g., "peak" time periods and intense geographical areas). This knowledge allows them to better manage the current resources in anticipation of forecasted needs. This practice has motivated a new research line aimed at developing solution approaches that better reproduce the dispatcher's behavior. Probability distributions about the occurrence of new service requests, as derived from historical data, are often exploited for this purpose. In other cases, the problem-solving method accounts for future demands without explicitly exploiting probability distributions. These are discussed below.

1.4.1 Double Horizon

This approach has been introduced in Mitrović-Minić *et al*. (2004) for a pick-up and delivery problem with time windows, where the number of vehicles is a free variable. The proposed double horizon is a generalization of the classical short-term rolling horizon approach (where only requests with a time window that is sufficiently close to the current time are considered). Here, both a short-term and a long-term planning horizon are considered. The latter is introduced to alleviate the adverse long-term effects of apparently good short-term decisions. Basically, a different objective is associated with each horizon type. The objective for the short term horizon corresponds to the true objective, like the total distance traveled, while the objective associated with the long-term horizon favors large slack times in the routes to better accommodate future requests. The optimization is done in both cases with a simplified version of the tabu search heuristic of Gendreau *et al*. (1998). The computational results on instances generated from data collected in two courier companies operating in Vancouver, Canada, demonstrate the benefits of the double horizon approach when compared with the classical single horizon approach.

1.4.2 Waiting Strategies

In the case of problems with time windows, a vehicle should wait if it arrives at its next destination before the corresponding time window. In a dynamic setting, however, it would be better for the vehicle to wait at the previous customer location in order to reach its next destination at the time window's lower bound (earliest departure policy). It would also be possible to wait more, as long as the vehicle does not arrive after the time window's upper bound (latest departure policy). With this kind of least-commitment strategy,

the next destination can be reconsidered if new requests occur in the mean time. More sophisticated strategies for introducing waiting times at strategic places along a planned route can also be devised.

For example, Mitrović-Minić and Laporte (2004) analyze this issue for a pick-up and delivery problem with time windows. They show that a mixed waiting strategy that combines earliest and latest departure policies provides the best results with regard to the number of vehicles and total traveled distance. The best approach is based on a dynamic partition of a planned route into segments made of close locations. Within a segment, a vehicle always departs as soon as possible from its current location; but when it is time to cross a boundary between two segments to travel further, the vehicle waits at its current location for a fraction of the time available up to the latest possible departure time.

In Ichoua *et al*. (2001), a vehicle that has completed its service at a customer location is forced to wait for some amount of time, if its next destination is far away and the probability of a new request arrival in its vicinity in the near future is high enough. Thus, explicit distribution probabilities are exploited in the waiting rule. If a new request occurs in the mean time, all waiting vehicles in the neighborhood are considered and a least-cost insertion is performed. Otherwise, the vehicle departs for its next planned destination. The proposed approach is assessed within the tabu search heuristic of Gendreau *et al*. (1999). Experimental results show that this strategy is effective, especially on harder problems (i.e., small fleet of vehicles and high demand rates). Branke *et al*. (2005) study different waiting strategies for a vehicle routing problem with no time windows, but with a time deadline. Given a set of planned routes and a single new request (with a location that is uniformly distributed within the service region and a time of occurrence that is either known or unknown), the authors study waiting strategies at customer locations aimed at maximizing the probability of being able to serve the new request. They prove that in the case of a single vehicle the optimal strategy is not to wait. They also derive an optimal waiting strategy for two vehicles. Then, different heuristic waiting strategies are compared for an arbitrary number of vehicles (i.e., different ways to distribute the slack time among the customers). The best one is derived from the optimal waiting strategy for two vehicles. An evolutionary algorithm that searches the space of waiting time values is also proposed. Basically, each chromosome is a real-valued vector that contains the waiting time associated with each customer. The paper empirically demonstrates that, when compared with the "no wait" strategy, distributing the slack time among the customers based on the best waiting heuristic leads to substantial improvements both with regard to the probability of serving the new request (up by about 10%) and the detour incurred to serve it (down by about 35%).

In Bent and van Hentenryck (2003), a waiting strategy that includes both known and sampled (future) customer requests is proposed to improve solutions obtained within the MSA framework (as described below). Basically, the vehicle departure at a given customer is delayed as long as there are sampled customers between that customer and the next known customer in the planned route.

1.4.3 Fruitful Regions

In the work of van Hemert and La Poutré (2004), an anticipated move toward a fruitful region is performed if this move does not induce any constraint violation for an actual request. A region is said to be fruitful when the potential of occurrence of a new service request, based on probability distributions, is high. This strategy is integrated within an evolutionary algorithm developed for a dynamic pick-up and delivery problem, where parent solutions in the current population exchange loads to generate new offspring solutions.

1.4.4 Multiple Scenario Approach (MSA)

Future requests are integrated within the MPA framework by sampling their probability distribution to produce plans that include both actual and forecasted requests (Bent and van Hentenryck, 2004). The intent is to leave room in the routing plan to accommodate future requests. The real plans are then obtained by projection over actual requests only. Experiments were conducted on simulated problems motivated from long-distance courier mail services, where both customer locations and their service time were stochastic variables.

1.5 CONCLUSION

Recent advances in communication and information technologies, as well as an increased interest in just-in-time distribution systems, have recently led researchers to focus on dynamic vehicle routing problems. Many important challenges remain for researchers working in this field. Among others:

- As new sources of real-time data become available, there is a need to filter out this information to focus on meaningful patterns and relationships. Methodologies aimed at analyzing data (e.g., data mining techniques) should thus be a focus of attention. Also, decentralized information processing associated with parallel system architectures should be another area where intense developments will be observed in the future.

- Some effort should be spent on developing a taxonomy of real-time fleet management problems, similar to the ones available for static problems. This will stimulate the development of problem-solving methodologies that are well adapted to the specific characteristics of the problems to be solved.
- Time pressure is a major impediment to optimization methods based on adaptation of algorithms originally developed for the static case. Recent progress in computer science, especially with regard to parallel computing techniques, offer powerful tools to alleviate this problem (although they cannot replace careful algorithmic development). It should be noted that the literature on parallel optimization algorithms for real-time fleet management problems is still very scarce.
- Diversion is another important issue that has been relatively neglected. Decisions are taken very quickly in this context, because vehicles are moving fast. Thus, a number of issues arise with regard to the trade off that should be achieved between solution quality and computation time.
- There is an increased interest in solution approaches that anticipate future demands. However, this research line has not yet reached maturity and major contributions are still lying ahead. For example, formal modeling frameworks that integrate the uncertainty associated with future service requests must still be developed.
- Most papers address the uncertainty associated with new customer requests (as reviewed in this paper) or variability in travel times. On the other hand, there is still a lack of solution approaches for less predictable events, like service cancellations, unexpected congestion due to an accident or vehicle breakdowns. Innovative ways to alleviate their impact on the overall system performance, in particular through appropriate recourse strategies based on resequencing and reassignment decisions, need to be devised.
- As pointed out in Larsen (2000), a highly dynamic environment is characterized by scarce and low-quality a priori information, as well as frequent changes in input data. Furthermore, most requests are urgent and a very short reaction time is required. Specific algorithmic developments must be developed for these challenging environments.
- On a more theoretical ground, the analysis of worst-case performance of dynamic vehicle routing algorithms is of interest. In particular, a fundamental question is by how much a particular algorithm can deviate from the optimum, due to unknown information.

ACKNOWLEDGEMENTS

Financial support for this work was provided by the Canadian Natural Sciences and Engineering Research Council (NSERC) and by the Fonds Québécois de Recherche sur la Nature et les Technologies (FQRNT). This support is gratefully acknowledged.

REFERENCES

Angelelli, E., Mansini, R., and Speranza, M. G., 2004, A real-time vehicle routing model for a courier service problem, in: *Distribution Logistics: Advanced Solutions to Practical Problems*, Lecture Notes in Economics and Mathematical Systems 544, B. Fleischmann and A. Klose, eds., Springer, Berlin, pp. 87-104.

Attanasio, A., Cordeau, J.-F., Ghiani, G., and Laporte, G., 2004, Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem, *Parallel Computing* **30**:377-387.

Bausch, D. O., Brown, G., and Ronen, D., 1995, Consolidating and dispatching truck shipments of heavy petroleum products, *Interfaces* **25**:1-17.

Bell, W., Dalberto, L., Fisher, M., Greenfield, A., Jaikumar, R., Kedia, P., Mack, R., and Prutzman, P., 1983, Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer, *Interfaces* **13**:4--23.

Bent, R., and Van Hentenryck, P., 2004, Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research* **52**:977-987.

Bent, R., and Van Hentenryck, P., 2003, Dynamic vehicle routing with stochastic requests, Technical Report CS-03-10, Department of Computer Science, Brown University, Providence, U.S.A.

Bertsimas, D. J., and Van Ryzin, G., 1991, A stochastic and dynamic vehicle routing problem in the Euclidean plane, *Operations Research* **39**:601-615.

Bertsimas, D. J., and Van Ryzin, G., 1993, Stochastic and dynamic vehicle routing problem in the Euclidean plane with multiple capacitated vehicles, *Operations Research* **41**:60-76.

Branke, J., Middendorf, M., Noeth, G., and Dessouky, M., 2005, Waiting strategies for dynamic vehicle routing, *Transportation Science* **39**:298-312.

Brown G., Ellis, C., Graves, G., and Ronen, D., 1987, Real time wide area dispatch of Mobil tank trucks", *Interfaces* **17**:107-120.

Caramia, M., Italiano, G. F., Oriolo, G., Pacifici, A., and Perugia, A., 2002, Routing a fleet of vehicles for dynamic combined pick-up and delivery services, in: *Operations Research Proceedings 2001*, P. Chamoni, R. Leisten, A. Martin, J. Minnemann and H. Stadtler, eds., Springer, Berlin, pp. 3-8.

Cordeau, J.-F., and Laporte, G., 2003, A tabu search for the static multi-vehicle dial-a-ride problem, *Transportation Research B* **37**:579-594.

Cordeau, J.-F., Laporte, G., Potvin, J.-Y., and Savelsbergh, M. W. P., 2004, Transportation on demand, Technical Report CRT-2004-25, Centre de recherche sur les transports, Montreal, Canada, 2004 (forthcoming in: *Transportation*, Handbooks in Operations Research and Management Science, C. Barnhart and G. Laporte, eds., North-Holland, Amsterdam).

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F., 1995, Time constrained routing and scheduling, in: *Network Routing*, Handbooks in Operations Research and Management Science 8, M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser, eds., North-Holland, Amsterdam, pp. 35-140.

Fisher, M. L., Greenfield, A. J., Jaikumar R., and Kedia, P., 1982, Real-time scheduling of a bulk-delivery fleet: Practical application of a Lagrangian relaxation, Report 82-10-11, Decision Sciences Department, University of Pennsylvania, Philadelphia, U.S.A.

Gambardella, L. M., Rizzoli, A. E., Oliverio, F., Casagrande, N., Donati, A. V., Montemanni, R., and Lucibello, E., 2003, Ant colony optimization for vehicle routing in advanced logistic systems, in: *International Workshop on Modelling and Applied Simulation*, Bergeggi, Italy, pp. 3-9.

Gendreau, M., Guertin, F., Potvin, J.-Y., and Séguin, R., 1998, Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries, Technical Report CRT-98-10, Centre de recherche sur les transports, Montreal, Canada (forthcoming in *Transportation Research C*).

Gendreau, M., Guertin, F., Potvin, J.-Y., and Taillard, É. D., 1999, Parallel tabu search for real-time vehicle routing and dispatching, *Transportation Science* **33**:381-390.

Gendreau, M., and Potvin, J.-Y., 1998, Dynamic vehicle routing and dispatching, in: *Fleet Management and Logistics*, T. G. Crainic and G. Laporte, eds., Kluwer, Boston, pp. 115-126.

Gendreau, M., and Potvin, J.-Y., eds., 2004, *Transportation Science* **38:**397-487 (special issue on real-time fleet management).

Ghiani, G., Guerriero, F., Laporte, G., and Musmanno, R., 2003, Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies, *European Journal of Operational Research* **151**:1-11.

Glover, F., 1996, Ejection chains, reference structures and alternating path methods for traveling salesman problems, *Discrete Applied Mathematics* **65**:223-253.

Ichoua, S., Gendreau, M., and Potvin, J.-Y., 2000, Diversion issues in real-time vehicle dispatching, *Transportation Science* **34**:426-438.

Ichoua, S., 2001, Problèmes de gestion de flottes de véhicules en temps réel, Ph.D. Thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Montreal, Canada.

Ichoua, S., Gendreau, M., and Potvin, J.-Y., 2006, Exploiting knowledge about future demands for real-time vehicle dispatching, Transportation Science **40:** 211-225.

Kilby, P., Prosser, P., and Shaw, P., 1998, Dynamic VRPs: A study of scenarios, Technical Report APES-06-1998, University of Strathclyde, Glasgow, UK.

Krumke, S. O., Rambau, J., and Torres, L. M., 2002, Real-time dispatching of guided and unguided automobile service units with soft time windows, in: *Proceedings of the 10th Annual European Symposium on Algorithms*, Lecture Notes in Computer Science 2461, pp. 637-648.

Larsen, A., 2000, The dynamic vehicle routing problem, Ph.D. Thesis, Report IMM-PHD-2000-73, Department of Mathematical Modeling, Technical University of Denmark, Lyngby, Denmark.

Larsen, A., Madsen, O. B. G., and Solomon, M. M., 2002, Partially dynamic vehicle routing – Models and algorithms, *Journal of the Operational Research Society* **53**:637-646.

Madsen, O. B .G., Ravn, H. F., and Rygaard, J. M., 1995a, A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives, *Annals of Operations Research* **60**:193-208.

Madsen, O. B. G., Tosti, K., and Vaelds, J., 1995b, A heuristic method for dispatching repair men, *Annals of Operations Research* **61**:213-226.

Mitrović-Minić, S., and Laporte, G., 2004, Waiting strategies for the dynamic pickup and delivery problem with time windows, *Transportation Research B* **38**:635-655.

Mitrović-Minić, S., Krishnamurti, R., and Laporte, G., 2004, Double-horizon based heuristics for the dynamic pickup and delivery Problem with time windows, *Transportation Research B* **38**:669-685.

Montemanni, R., Gambardella, L. M., Rizzoli, A. E, and Donati, A. V., 2005, Ant colony system for a dynamic vehicle routing problem, *Journal of Combinatorial Optimization* **10**:327-343.

Powell, W. B., Jaillet, P. and Odoni, A. R., 1995, Stochastic and dynamic networks and routing, in: *Network Routing*, Handbooks in Operations Research and Management Science 8, M. O. Ball, T. L. Magnanti, C. L. Monma and G. L. Nemhauser, eds., Elsevier, Amsterdam, pp. 141-295.

Psaraftis, H. N., 1995, Dynamic vehicle routing: Status and prospects, *Annals of Operations Research* **61**:143-164.

Psaraftis, H. N., 1988, Dynamic vehicle routing problems, in: *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad, eds., North-Holland, Amsterdam, pp. 223-248.

Psaraftis, H. N., 1983, An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows, *Transportation Science* **17**:351-357.

Psaraftis, H. N., 1980, A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem, *Transportation Science* **14**:130-154.

Psaraftis, H. N., Orlin, J. B, Bienstock, D., and Thompson, P. M., 1985, Analysis and solution algorithms of sealift routing and scheduling problems: Final report, Working Paper 1700-85, Sloan School of Management, MIT, Cambridge, U.S.A.

Regan, A. C., Mahmassani, H. S., and Jaillet, P., 1995, Improving efficiency of commercial vehicle operations using real-time information: Potential uses and assignment strategies, *Transportation Research Record* **1493**:188-197.

Rivard, R., 1981, Construction des parcours des véhicules et des horaires des chauffeurs pour le transport des personnes handicapées, Technical Report CRT-240, Centre de recherche sur les transports, Montreal, Canada.

Rochat, Y., and Taillard, É. D., 1995, Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics* **1**:147-167.

Rousseau, J.-M., and Roy, S., 1988, RAO - Répartition assistée par ordinateur: La description du prototype, Technical Report CRT-564, Centre de recherche sur les transports, Montreal, Canada.

Roy, S., Rousseau, J.-M., Lapalme, G., and Ferland, J. A., 1984, Routing and scheduling for the transportation of disabled persons: The algorithm, Technical Report CRT-412, Centre de recherche sur les transports, Montreal, Canada.

Savelsbergh, M., and Sol, M., 1998, DRIVE: Dynamic routing of independent vehicles, *Operations Research* **46**:474-490.

Séguin, R., Potvin, J.-Y., Gendreau, M., Crainic, T. G., and Marcotte, P., 1997, Real-time decision problems: An operational research perspective, *Journal of the Operational Research Society* **48**:162-174.

Swihart, M. R., and Papastavrou, J. D., 1999, A stochastic and dynamic model for the single-vehicle pick-up and delivery Problem, *European Journal of Operational Research* **114**:447-464.

Van Hemert, J. I., and La Poutré, J. A., 2004, Dynamic routing problems with fruitful regions: Models and evolutionary computation, in: *Parallel Problem Solving from Nature VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, eds., Springer, Berlin, pp. 690-699, 2004.

Wilson, N. H. M., and Colvin, N. H., 1977, Computer control of the Rochester dial-a-ride system, Technical Report R77-30, Department of Civil Engineering, MIT, Cambridge, U.S.A.

Chapter 2

CLASSIFICATION OF DYNAMIC VEHICLE ROUTING SYSTEMS

Allan Larsen[1], Oli B.G. Madsen[1] and Marius M. Solomon[2]
[1]Centre for Traffic and Transport, Technical University of Denmark, Bygningstorvet, DK-2800 Kongens Lyngby, Denmark; [2]Department of Management Sciences, College of Business Administration, Northeastern University, 314 Hayden Hall, 360 Huntington Avenue, Boston, Massachusetts 02115

Abstract: This chapter discusses important characteristics seen within dynamic vehicle routing problems. We discuss the differences between the traditional static vehicle routing problems and its dynamic counterparts. We give an in-depth introduction to the degree of dynamism measure which can be used to classify dynamic vehicle routing systems. Methods for evaluation of the performance of algorithms that solve on-line routing problems are discussed and we list some of the most important issues to include in the system objective. Finally, we provide a three-echelon classification of dynamic vehicle routing systems based on their degree of dynamism and the system objective.

Keywords: degree of dynamism, dynamic vehicle routing, competitive analysis

2.1 INTRODUCTION

The vehicle routing problem (VRP) has received an immense attention from the scientific community during the last three to four decades as it often play a vital role in the design of distribution systems. Basically, the VRP consists of designing routes for a set of capacitated vehicles that are to service a set of geographically dispersed customers at the least cost. In real-life contexts restrictions such as time windows for when the service can commence make up important side-constraints to the problem. The basic VRP deals with customers who are known in advance to the planning process. Furthermore, all other information such as the driving time between the customers and the service times at the customers are used to be known prior to the planning.

This provides that perfect set-up for applying advanced mathematical based optimization methods such as set partitioning. However, when dealing with real-life applications the information often tends to be uncertain or even unknown at the time of the planning. The traditional VRP can be said to be static as well as deterministic. In contrast to this, the dynamic vehicle routing problem (DVRP) considers a VRP in which a subset (or the full set) of customers arrive after the day of operation has begun. The DVRP will have to be able to consider how to include the new requests into the already designed routes.

The major technological advances during the recent years mean that the majority of new vehicles are equipped with advanced GPS/GIS systems. Hence, the distribution companies are now able to monitor the vehicles' position and status at any given time. Furthermore, the development and implementation of Enterprise Resource Planning (ERP) systems now means that the distribution companies also are able to link the customer data with inventory information etc. Until recently advanced distribution planning systems were usually only seen in big enterprises. However, the before mentioned technological achievements implies that also medium sized distribution companies now implement advanced distribution planning systems. The next step will be to move the implementation of advanced systems based on DVRP's into the small enterprises. This development will probably be accelerated during the coming years as it seems inevitable that the demand for logistics based on the just-in-time (JIT) concept will keep on growing year by year. An example of this could be the transportation of the elderly and handicapped. Until now, most services required the passengers to book their transport the day before the travel was to take place. However, with the increased access to the internet these services will experience a growing demand for on-the-day booking. This means that the service provider will have to implement a routing system which is able to insert the requests for service which is received during the day of operation into the planned routes.

In this chapter we will examine the dynamic vehicle routing problem from a temporal perspective. In section 2.2 we discuss how the set of dynamic vehicle routing problems can be defined. In section 2.3 we discuss some of the differences between the DVRP and the traditional static VRP. In section 2.4 we discuss how instances of the DVRP's can be classified according to their degree of dynamism. In section 2.5 we give a discussion on how performance of algorithms for the DVRP can be measured which elements are relevant to consider in the system objective. In section 2.6 a three-echelon framework for classifying DVRP's based on the degree of dynamism measure and the objective is presented. Finally, in section 2.7 we provide a short summary of the discussion of the characteristics of the DVRP.

2.2 THE DYNAMIC VEHICLE ROUTING PROBLEM

In order to give a definition of the Dynamic Vehicle Routing Problem we take a look at the work by Psaraftis, 1988, who was among the very first to consider the dynamic extension of the traditional static VRP. Psaraftis uses the following classification of the static routing problem;

- *"if the output of a certain formulation is a set of preplanned routes that are not re-optimized and are computed from inputs that do not evolve in real-time".*

While he refers to a problem as being dynamic;

- *"if the output is not a set of routes, but rather a policy that prescribes how the routes should evolve as a function of those inputs that evolve in real-time".*

In the above definition by Psaraftis the temporal dimension plays a vital role for the categorizing of a vehicle routing problem. In this chapter we will demonstrate that the time of when relevant information is made known to the planner distinguishes dynamic from static vehicle routing problems.

In the definition given below we verbally define what we mean when we talk about a static vehicle routing problem.

The Static Vehicle Routing Problem

- All information relevant to the planning of the routes is assumed to be known by the planner before the routing process begins.
- Information relevant to the routing does not change after the routes have been constructed.

The information which is assumed to be relevant includes all attributes of the customers such as the geographical location of the customers, the on-site service time and the demand of each customer. Furthermore, system information as for example the travel times of the vehicle between the customers must be known by the planner.

The dynamic counterpart of the static vehicle routing problem as defined in the above definition could then be formulated as:

The Dynamic Vehicle Routing Problem

- Not all information relevant to the planning of the routes is known by the planner when the routing process begins.
- Information can change after the initial routes have been constructed.

In Figure 2-1 a simple example of a dynamic vehicle routing situation is shown. In the example, two un-capacitated vehicles service two types of requests:

1. *Advance requests*, which can also be referred to as static customers as these requests for service has been received before the routing process was begun.
2. *Immediate requests*, which can also be referred to as dynamic customers as these will appear in real-time during the execution of the routes.

In the example in Figure 2-1 time windows are not considered. The advance request customers are represented by black nodes, while those that are immediate requests are depicted by white nodes. The solid lines represent the two routes the dispatcher has planned prior to the vehicles leaving the depot. The two thick arcs indicate the vehicle positions at the time the immediate requests are received. Ideally, the new customers should be inserted into the already planned routes without the order of the non-visited customers being changed and with minimal delay. This is the case depicted on the right hand side route. However, in practice, the insertion of new customers will usually be a much more complicated task and will imply either partial or full re-planning of the non-visited part of the route. This is illustrated by the left hand side route where servicing the new customer creates a large detour.

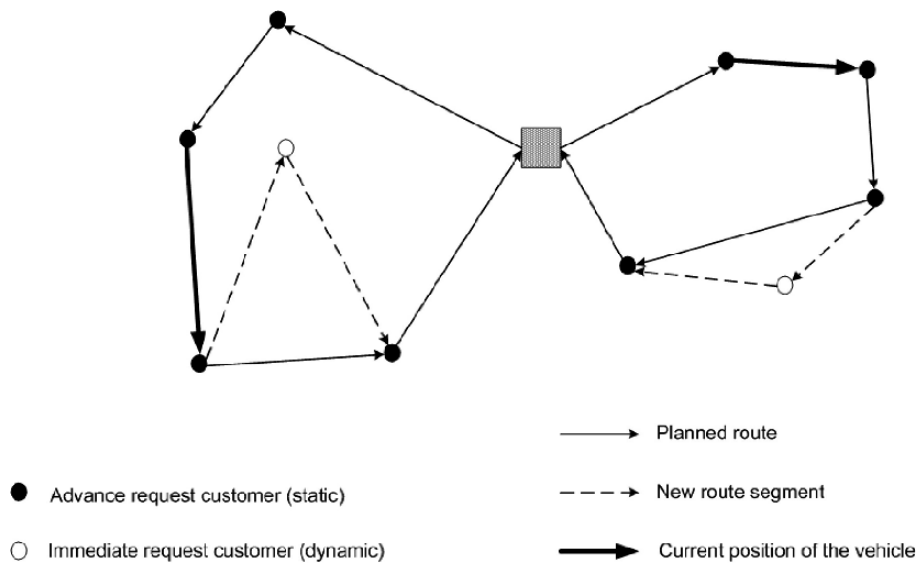


Figure 2-1. A dynamic vehicle routing scenario with 8 advance and 2 immediate request customers.

Generally, the more restricted and complex the routing problem is, the more complicated the insertion of new dynamic customers will be. For instance, the insertion of new customers in a time window constrained routing problem will usually be much more difficult than in a non-time constrained problem. Note that in an on-line routing system customers may even be denied service, if it is not possible to find a feasible spot to insert them. Often this policy of rejecting customers includes an offer to serve the customers the following day of operation. However, in some systems - as for instance the pick-up of long-distance courier mail - the service provider (distributor) will have to forward the customer to a competitor when they are not able to serve them.

2.3 STATIC VERSUS DYNAMIC VEHICLE ROUTING

In this section the differences between the conventional static and the dynamic vehicle routing problem as described in the section above will be discussed.

Psaraftis, 1988, Psaraftis, 1995, lists 12 issues on which the dynamic vehicle routing problem differs from the conventional static routing problem. Below we give a brief summary of these issues as they are indeed very central to our discussion of static versus dynamic routing. The full discussion of the issues can be found in Psaraftis, 1988, Psaraftis, 1995.

1. *Time dimension is essential*.

In a static routing problem the time dimension may or may not be important. In the dynamic counterpart time is always essential. The dispatcher must as a minimum know the position of all vehicles at any given point in time and particularly when the request for service or other information is received by the dispatcher.

2. *The problem may be open-ended.*

The process is often temporally bounded in a static problem. The routes start and end at the depot. In a dynamic setting the process may very well be unbounded. Instead of routes one considers paths for the vehicles to follow.

3. *Future information may be imprecise or unknown*.

In a static problem all information is assumed to be known and of the same quality. In a real-life dynamic routing problem the future is almost never known with certainty. At best probabilistic information about the future may be known.

4. *Near-term events are more important.*

Due to the uniformity of the information quality and lack of input updates all events carry the same weight in a static routing problem. Whereas in a dynamic setting it would be unwise immediately to commit vehicle resources to long-term requirements. The focus of the dispatcher should therefore be on near-term events when dealing with a dynamic routing problem.

5. *Information update mechanisms are essential.*

Almost all inputs to a dynamic routing problem are subject to changes during the day of operation. It is therefore essential that information update mechanisms are integrated into the solution method. Naturally, information update mechanisms are not relevant within a static context.

6. Re-sequencing and reassigning decisions may be warranted.

In dynamic routing new input may imply that decisions taken by the dispatcher become suboptimal. This forces the dispatcher to reroute or even reassign vehicles in order to respond to the new situation.

7. *Faster computation times are necessary.*

In static settings the dispatcher may afford the luxury of waiting for a few hours in order to get a high quality solution, in some cases even an optimal one. In dynamic settings this is not possible, because the dispatcher wishes to know the solution to the current problem as soon as possible (preferably within minutes or seconds). The *running-time* constraint implies that rerouting and reassignments are often done by using local improvement heuristics like insertion and *k*-interchange.

8. *Indefinite deferment mechanisms are essential*.

Indefinite deferment means the eventuality that the service of a particular demand be postponed indefinitely because of that demands unfavorable geographical characteristics relative to the other demands. This problem could for instance be alleviated by using time window constraints or by using a nonlinear objective function penalizing excessive wait.

9. *Objective function may be different.*

Traditional static objectives such as minimization of the total distance traveled or the overall duration of the schedule might be meaningless in a dynamic setting because the process may be open-ended. If no information about the future inputs is available, it might be reasonable to optimize only over known inputs. Some systems also use nonlinear objective functions in order to avoid undesirable phenomena such as the above mentioned indefinite deferment.

10. *Time constraints may be different*.

Time constraints such as latest pickup times tend to be softer in a dynamic routing problem than in a static one. This is due to the fact that denying service to an immediate demand, if the time constraint is not met, is usually less attractive than violating the time constraint.

11. *Flexibility to vary vehicle fleet size is lower.*

In static settings the time gap between the execution of the algorithm and the execution of the routes usually allows adjustments of the vehicle fleet. However, within a dynamic setting the dispatcher may not have instant access to backup vehicles. Implications of this may mean that some customers receive lower quality of service.

12. *Queuing considerations may become important.*

If the rate of customer demand exceeds a certain threshold, the system will become congested and the algorithms are bound to produce meaningless results. Although vehicle routing and queuing theory are two very well-studied disciplines, the effort to combine these has been scant.

Psaraftis, 1995, also proposes a taxonomy used for characterizing attributes of the information forming the input for the vehicle routing problem. The taxonomy consists of the following concepts:

- **Evolution of information**. In static settings the information does not change, nor is the information updated. In dynamic settings the information will generally be revealed or updated as time goes on.
- **Quality of information.** Inputs could either; 1) be known with certainty (deterministic), 2) be known with uncertainty (forecasts) or 3) follow prescribed probability distributions (probabilistic). Usually, the quality of the information in a dynamic setting is good for near-term events and poorer for distant events.
- **Availability of information.** Information could either be local or global. One example of local information is when the driver learns of the precise amount of oil the current customer needs, while a globally based information system would be able to inform the dispatcher of the current status of all the customers' oil tanks. The rapid advances within information technologies increase the availability of information. This fast growth in the amount of information available raises the issue of when to reveal/make use of the information. For instance, the dispatcher may choose to reveal only the information that is needed by the drivers although she might have access to all information.

- **Processing of information.** In a *centralized* system all information is collected and processed by a central unit. In a *decentralized* system some of the information could for instance be processed by the driver of each truck.

Powell *et al*., 1995, distinguish between dynamism within a problem, a model and the application of a model. They argue that:

- A **problem** is dynamic if one or more of its parameters is a function of time. This includes models with dynamic data that change constantly as well as problems with time-dependent data which are known in advance.
- A **model** is dynamic if it explicitly incorporates the interaction of activities over time. Here one should distinguish between deterministic dynamic models and stochastic models.
- An **application** is dynamic if the underlying model is solved repeatedly as new information is received. Consequently, solving models within dynamic applications require huge computational resources.

In this section we have focused on the differences between the static and the dynamic versions of the vehicle routing problems. As mentioned in this section the temporal attributes are among the most central characteristics of a dynamic VRP. As we will see in the next sections the time of when the immediate requests are received and the number of these requests can be used to classify dynamic VRP in a general framework.

2.4 THE DEGREE OF DYNAMISM

Measuring the performance of a dynamic vehicle routing system is not a trivial assignment. In contrast to a deterministic and static vehicle routing problem the performance of the dynamic counterpart is assumed to be dependent not only on the number of customers and their spatial distribution, but also the number of dynamic events and the time when these events actually take place. Therefore, a single measure for describing the system's "dynamism" would be very valuable when one wants to examine the performance of a specific algorithm under varying conditions.

As we will see in the following, measures that might seem promising for describing dynamism for one system might turn out to be inadequate for describing the dynamism of other systems. Therefore this chapter is divided into two sections where the first section discusses measures for dynamism in systems without time windows while the second section examines systems with the presence of time windows.

2.4.1 Dynamism Without Time Windows

In this section we examine measures which try to describe the dynamism of a dynamic vehicle routing system without time windows. In a system without time windows only three parameters are relevant: The number of static customers, the number of dynamic customers and the arrival times of the dynamic customers.

2.4.1.1 The degree of dynamism

Usually, not all information is received in real-time during the execution of the routes. In most cases part of the information will be available before the process of creating the routes begins. The extent of the information received in real-time relative to the total system information provides insights in how dynamic the routing system really is. The most basic measure for this in a routing context is the number of immediate requests, n_{imm}, relative to the total number of requests, n_{tot}. Lund *et al*., 1996, were the first to define this ratio as *the degree of dynamism* of the system considered. We denote the ratio as *dod* which means that:

$$dod = \frac{n_{imm}}{n_{tot}}$$

In the example shown in Figure 2-1 the degree of dynamism is therefore 20% (2 out of 10 customers arrive while the system is on-line).

However, this basic measure of Lund *et al*. does not take the arrival times of the immediate requests into account. This means that two systems, one in which the immediate requests are received at the beginning of the planning horizon and the other in which they occur late during the day, are perceived as equivalent. However, in real-life routing situations these two scenarios are very different. Figure 2-2 illustrates two DVRP scenarios in which the times for receiving immediate requests differ considerably. In **Scenario A** all six immediate requests are received relatively early during the planning horizon. In **Scenario B** the requests are distributed almost evenly throughout the planning horizon. We suggest that the planner would prefer the first scenario to the second, since **Scenario A** provides him with time to react to the immediate requests as opposed to the situation sketched in **Scenario B** in which he may not have enough time to find a suitable reaction to the immediate requests received at the very end of the planning horizon.

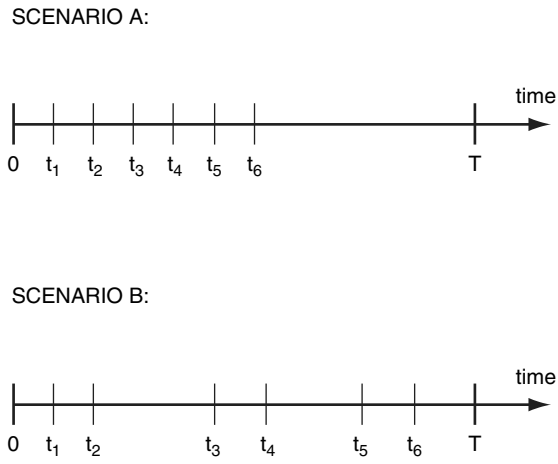


Figure 2-2. Arrival time of immediate requests.

Furthermore, from a performance point of view it is clear that having the highest number of requests in the pool of waiting requests improves the solution quality with respect to the objective of minimizing the total distance driven. Hence, in the systems illustrated by the two scenarios in Figure 2-2 the expected length of the route would be shorter in **Scenario A** than in **Scenario B** due to the fact that the planner in the former scenario from time t_6 has all information on the locations of the requests which means that he from that point in time could form an optimal TSP tour through the pool of waiting customers.

In section 2.4.1.2 we extend the above defined measure to include the times when the immediate requests are received.

Before turning to the extended degree of dynamism measure, a final comment on the scenarios illustrated in Figure 2-2 should be made. Assuming that a number of advance requests are already in the pool of waiting requests to be served, the system described in **Scenario A** is could be the most difficult to manage since if the planner is already busy taking care of the advance request customers during the early stages of the planning horizon, he would be likely to prefer to have to deal with the immediate requests later during the day after the advance requests have been serviced. Using this reasoning one might classify **Scenario A** as the more dynamic of the two systems illustrated. However, we chose to go with the first classification since this seems to be the most intuitively correct with respect to the performance of the system.

2.4.1.2 Effective Degree of Dynamism - EDOD

A natural extension of the basic measure defined above would be to include information on *when* the immediate requests are received by the dispatcher. The planning horizon is defined to start at time *0* and end at time *T*. All advance requests are received before the planning horizon starts or at time *0* at the latest. The request time of the *i*'th immediate request is denoted t_i, i.e. $0 < t_i \leq T$. The number of immediate requests received during the entire planning horizon is denoted n_{imm} and the total number of requests received during the planning horizon is denoted n_{tot}. We now define the following measure as the *effective degree of dynamism*, denoted *edod*:

$$edod = \frac{\sum_{i=1}^{n_{imm}} \frac{t_i}{T}}{n_{tot}}$$

The effective degree of dynamism then represents an average of how late the requests are received compared to the latest possible time the requests could be received. In other words, the effective degree of dynamism measure captures the temporal distribution of the customers. It can easily be seen that:

$$0 \leq edod \leq 1$$

In a pure dynamic system $edod = 0$ whereas $edod = 1$ in a pure static system in which all the requests are received at time *0* and time *T* respectively. It is also obvious that

$$\lim_{t_i \to T \forall i} edod = 1$$

2.4.2 Dynamism and Time Windows

The measures defined above do not allow for time windows to be taken into consideration. However, these measures can easily be refined in such a way that the time windows are also included in the measure. The time the *i*'th immediate request is received is denoted t_i and the earliest time that service can begin (i.e. the start of the time window) is denoted e_i while the latest possible time that service should begin is denoted l_i. In applications with time windows the *reaction time* is a very important issue. The reaction time is defined as the temporal distance between the time the request is received and the latest possible time at which the service of the

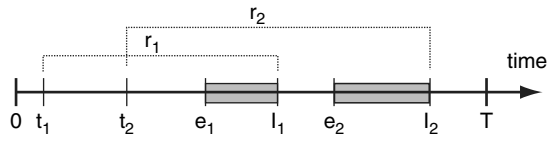


Figure 2-3. The reaction times of two dynamic customers in a DVRP with time windows.

requests should begin. In Figure 2-3 the reaction time of the *i*'th immediate request is denoted r_i, i.e. $r_i = l_i - t_i$.

2.4.3 Effective Degree of Dynamism – EDOD-TW

Consider the two scenarios sketched in Figure 2-4 - in both of these scenarios we have two immediate requests with time windows. In **Scenario A** the width of the time windows of the immediate requests is relatively wide compared to the width of the time windows in **Scenario B**. Furthermore, the reaction times in **Scenario A** are relatively long compared to the reaction times in **Scenario B**. This means that **Scenario A** would be preferred by the planner because this situation gives him much more room to insert the immediate requests into the routes.

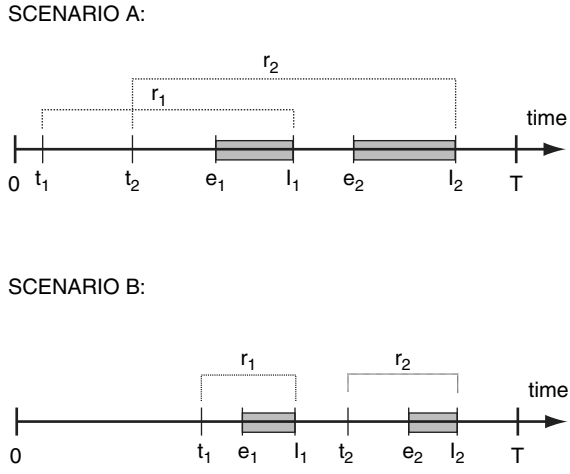


Figure 2-4. Two scenarios with two immediate requests.

In general the planner would prefer to have a relatively long reaction time for the immediate requests as this will increase the probability of finding a slot in which the new request can be inserted. The following measure therefore uses the relation between the reaction time and the remaining part of the planning horizon as the key component.

The effective degree of dynamism measure can then be extended to:

$$edod_{tw} = \frac{1}{n_{tot}} \sum_{i=1}^{n_{imm}} \left(\frac{T - (l_i - t_i)}{T} \right) = \frac{1}{n_{tot}} \sum_{i=1}^{n_{imm}} \left(1 - \frac{r_i}{T} \right)$$

As, for the *dod* and the *edod* measures it is straightforward to see that

$$0 \le edod_{tw} \le 1$$

as

$$l_i - t_i \le T, i = 1,2,...,n_{imm}$$

The degree of dynamism measure has been used by Bent, *et al*., 2004; Larsen, *et al*., 2004 and Larsen, *et al*., 2004, for studies of the performance of different versions of the dynamic routing problem. Both Bent, *et al*., 2004 and Larsen *et al*., 2004, examined the so-called Partially Dynamic Repairman Problem in which a subset of the requests is known in advance to the start of the planning horizon. In these works the performance of the algorithms examined was shown as a function of the degree of dynamism. Larsen *et al*., 2004, also used this approach to analyze a dynamic version of the TSPTW in which the dispatcher has access to a-priori information on the locations of the immediate requests. In the next section we discuss how the performance of on-line algorithms can be measured.

2.5 MEASURING THE PERFORMANCE OF DVRP'S

The objective of the DVRP often is a combination of multiple measures. For static VRP's the traditional objective has been to minimize the overall distribution costs. However, for DVRP's the level of service offered to the customers plays an important role in the overall performance of the system. In this section we will discuss some important issues that are relevant to consider when measuring the performance of a DVRP. First, we discuss the framework referred to as *competitive analysis*.

2.5.1 Competitive Analysis

The most accepted framework for measuring the performance of on-line algorithms is probably *competitive analysis*. Sleator, *et al*., 1985, were the first to formally introduce competitive analysis. The framework is often used during analyses of performance in production planning contexts. For a minimization problem the *competitive ratio*, cr_A, can be defined as:

$$cr_A = \sup_I \frac{z(A,I)}{z^*(I)}$$

where $z(I)$ is the cost of the solution found by algorithm A for instance I and z^* is the optimal cost found by an (ideal) offline algorithm which had access to all the instances beforehand. This way the competitive analysis framework offers a measure for evaluating the performance of a certain on-line routing policy based on the worst-case ratio between this policy and the optimal offline policy. This means that the loss of cost-efficiency which is due to the lack of full information can be quantified for each policy examined.

The competitive analysis framework provides a strong basis for studies of the performance of on-line algorithms which may produce interesting analytical results and insights. However, normally only very simple versions of the DVRP can be treated using this framework. Important real-life constraints such as time-windows have so far proved to be too complex to be dealt with. Bertsimas, *et al*., 1991; Bertsimas, *et al*., 1993 and Bertsimas, *et al*., 1993, derived a number of worst-case bounds in their early work on the Dynamic Traveling Repairman Problem (DTRP). These contributions were probably the first attempts to examine the class of DVRP's using competitive analysis. Ausiello *et al*., 2001, have studied the on-line version of the classical Traveling Salesman Problem (TSP) using competitive analysis. The authors examine two versions of the problem and provide lower bounds for the competitive ratio. Jaillet, *et al*., 2006, study online versions of the TSP and the Traveling Repairman Problem (TRP) and propose new online algorithms for these problems. Jaillet and Wagner quantify the value of the advanced information of when the requests are disclosed to the dispatcher by providing improved competitive ratios.

Complexity results and competitive analysis for Vehicle Routing Problems is the subject of the PhD-thesis by Paepe, 2002. Paepe gives a thorough analysis of the on-line version of the Dial-a-ride problem in which a single capacitated vehicle serves a set of customers that requests to be picked-up at some geographical location and to be transported to another location. The requests appear in real-time and Paepe derives the competitive

ratios of a number of routing policies. Angelelli *et al*., 2005, study a dynamic multi-period routing problem. Here, the orders arriving have to be completed either at that time period or the next. This means that the system will hold customers that are to be served right away as well as customers that will have to wait to be served. The authors introduce simple routing policies and analyze these by examining their competitive ratios. Even though competitive analysis provides a good basis for examining performance of on-line algorithms it should also be noted that competitive analysis often has been criticized as being too crude and unrealistic as in most real-life situations it is indeed possible to achieve an average performance which is considerably better than the one suggested by the competitive ratio.

As mentioned above the competitive analysis framework is most suited for simple version of the class of DVRP's. For more advanced versions of the problem the performance has to be evaluated through empirical studies. This is usually done by discrete-time simulation. Examples of this is the work on the dynamic version of the traveling salesman problem with time windows (DTSPTW) by Larsen *et al*., 2004 and the work on the dynamic vehicle routing problem with time windows by Gendreau *et al*., 1999.

This type of discrete-time simulation can also be extended so that the performance of a certain algorithm is evaluated by running the algorithm on both the original dynamic instances and on the instances in which the immediate requests are changed into static data. So for the example shown in Figure 2-1 the two immediate requests are turned into being advance requests making the problem a pure static VRP. This approach provides an estimate of the competitive ratio of the algorithm. Naturally, this empirical approach should not be mistaken for the real competitive analysis but it will be able to provide a high-quality estimate of performance of the algorithm provided that the appropriate data are used to perform the analysis.

2.5.2 Determining the Objectives

When measuring the performance of a DVRP system multiple objectives are often met. Sometimes, these objectives may even be conflicting. Naturally, the final objective of DVRP algorithms differs from one application to the next. However, some elements are almost always relevant to consider when defining the objective. Below we list the most important elements.

- **Distribution Costs**. Traditionally, distribution costs have been the main objective for static VRP's. For DVRP's the distribution costs should not be left out the main objective as these represent a true experienced cost to the distribution company.

- **Service Level**. The level of service offered to the customers may be in contrast to the objective of minimizing the distribution costs as a fast response to a new immediate request for service may imply that the vehicles will have to be routed in a sub-optimal manner according seen from the distance perspective.

- **Throughput Optimization**. The ability to serve as many customers as possible may for some DVRP's be the most important objective. Maximization of the expected number of requests serviced is for instance seen as the primary objective within the taxi cab business. However, in cases where no information about the future is available, it may be reasonable to optimize only over known input.

2.6 THREE-ECHELON FRAMEWORK FOR DVRP's

After having discussed various characteristics of the DVRP we will now use the degree of dynamism measure and the objectives to categorize a variety of DVRP's into a three echelon framework. The framework was initially introduced in Larsen *et al*., 2002 and distinguishes between weakly, moderately and strongly dynamic systems.

2.6.1 Echelon I – Weakly Dynamic Systems

The distribution of goods to a relatively high number of customers seldom tends to be subject to frequent changes. One example of this can be found in the distribution of heating oil or liquid gas to private households. The majority of the customers (at least 80%) are known in advance. These are often referred to as "automatic replenishment" customers as the oil company estimates their demand according to the "degree days" measure. However, requests may also be received in real-time from customers that are out of oil and therefore request immediate service. The reaction time is considerably longer in such a problem compared to that in a taxi dispatching system. Another example is the distribution of prepackaged bio-dynamic groceries such as vegetables and fruits to private households. The customers subscribe to a certain type of product and receive a delivery once a week. The distributor produces a set of fixed delivery routes once a month for the set of subscribers. However, in case some of the customers are away on holidays or are throwing a party the subscription may either be cancelled or increased on the day of operation. The fixed routes will have to be produced in such a way that they allow for extra orders. The number of immediate requests (or cancellations) usually is quite low (less than 5%) compared to the total number of customers on a daily route. The transportation of elderly and handicapped people is usually modeled as a

dynamic version of the dial-a-ride problem (see Paepe, 2002) and has until now also been subject to quite few immediate requests as the passengers tend to book their rides well before the day of the trip. Therefore, these routing systems also belong to the echelon of weakly dynamic systems.

These examples all have in common that relatively few immediate requests are received during the day of operation. This means that the distributors normally will be able to focus primarily on distribution costs and secondarily on minimizing the response time and hereby on maximizing the service level.

Solving weakly dynamic routing problems are usually based on one of the following two algorithmic approaches:

1. *Re-optimization* each time a new request is received by the dispatcher. This approach is obvious as only few immediate requests are received and therefore the re-optimization will only be relevant relatively few times. However, this approach only seem computational tractable in cases where the degree of dynamism is quite low. The re-optimization approaches seen in the literature are usually based on either meta-heuristics or in some cases on tailor-made heuristics. Weakly dynamic problems are well-suited for meta-heuristics as the computer can used the idle time between two immediate requests to improve the current solution. Examples of work based on re-optimization include Larsen, *et al*., 2004; Gendreau, *et al*., 1999, and Ichoua *et al*., 2006. The latter two work use parallel versions of the tabu search heuristic.
2. *Insertion* in the routes that were generated before the start of the day of operation. In many cases the dispatcher will be able to produce a set of routes with sufficient slack to accommodate the immediate requests while the routes are being carried out. Naturally, depending on the application it can be quite hard to estimate how much slack should be introduced in the routes. The insertion procedure is normally based on a simple savings-based heuristic that calculates the best spot to insert the new immediate request(s).

2.6.2 Echelon II – Moderately Dynamic Systems

The second echelon embraces routing applications in which the number of immediate requests accounts for a significant proportion of the total number of customers. On the other hand, for a moderately dynamic system, the full information on the advance request customers still constitutes a fair part of the total system data and should therefore not be under valuated in the design of the routing algorithm. Here, the objective can be hard to define as it has to reflect the subtle combination of minimization of the distribution costs and of the response time. Examples of moderately dynamic systems include the

pick-up and delivery of long-distance courier mail and the service and repair of for instance bank ATM terminals. For these applications the advance request customers often has a fixed contract for service at a specified time during the day. The requests that appear in real-time will on the other hand almost always require immediate action.

When dealing with a moderately dynamic routing system it is vital that the routing algorithm is computationally fast as the number of immediate requests is relatively high implying that the algorithm must be run over and over again. Therefore, simple insertion heuristics based on local-search approaches is an obvious way to solve these types of routing problems. Fleischmann *et al*., 2004, investigate various simple algorithmic approaches for solving a dynamic routing problem of a local area courier service. The datasets used for the empirical tests consist of 31% and 49% immediate requests respectively. The authors conclude that even very simple insertion methods perform surprisingly well.

Having a very fast solution method means that the decision on where to send the vehicle(s) next can be deferred until the latest possible moment. Ideally, this will improve the quality of the decisions made because the level of uncertainty decreases as the time elapses. The simple insertion approaches can be enhanced by implementing improvement heuristics that utilize the time between input updates to improve the current routes (similar to the tabu search approach as mentioned in section 2.6.1).

In case detailed a-priori information on future requests exists this should be integrated into the insertion solution approach. However, methods based on stochastic programming do not seem to be appropriate within a dynamic setting as these methods tend to become extremely cumbersome to solve due to their combinatorial structure.

2.6.3 Echelon III – Strongly Dynamic Systems

The most extreme type of strongly dynamic routing systems is without doubt emergency services, such as police, fire fighting and ambulances. Here, no requests a known in advance to the day of operation and these applications are characterized by the intense focus on the response time. The quality of an emergency system is often measured by the actual perceived response time. The emergency service providers and the public administration agree on a certain level of service which for instance defines that 90% of the calls should be served within 5 minutes whereas the remaining 10% of the calls should be served within 8 minutes. Another example of an application which also belongs to the third echelon is taxi cab services. Only a negligible number of the customers have ordered their ride in advance.

Due to the importance of the emergency service systems this application has received substantial attention from the scientific community since the early 1970s. The majority of these works focus on ways to decrease the system response time. Larson and Odoni deal with the subject in Larson, *et al*., 1980 and propose the hypercube queuing model which has been used in especially police patrol dispatch for more than two decades. However, the quality of a-priori information such as the potential locations of the next request is often quite poor. If, on the other hand, a-priori information on future requests are available these could potentially improve the solution quality. For example, this could involve moving an idle vehicle currently situated in a low demand area to a central location. This was considered in the interesting work by Gendreau, *et al*., 2001, who proposed a model for real-time relocation of ambulances. Brotcorne, *et al*., 2003, study the development of ambulance location and relocation models proposed during the past three decades. The study covers both deterministic and probabilistic models used at the planning stage as well as dynamic models that are able to relocate ambulances throughout the day.

For strongly dynamic systems queuing also plays an important role and the developed routing algorithms must be able to incorporate these issues. Examples of algorithms that are based on queuing theory include the work on the dynamic traveling repairman problem (DTRP) by Bertsimas, *et al*., 1991; Bertsimas, *et al*., 1993 and Bertsimas, *et al*., 1993. In the DTRP the objective is to minimize the expected waiting time and not the travel cost.

2.6.4 System Classification

In the previous section we have discussed how a dynamic vehicle routing application can be classified according to its degree of dynamism and the system objective and in this section we will provide a framework for classifying the routing applications according to these issues. The framework was first proposed by Larsen, *et al*., 2002. In Figure 2-5 the relationship between the degree of dynamism and the objective is illustrated for a number of the routing applications which have previously been discussed in this chapter.

The routing problems placed in the upper-right corner of the figure are characterized by a strong dynamism and an objective which seeks to minimize the response time of the system. The emergency service systems are the most pure examples of this but also taxi cab services possess similar properties. Oppositely, the problems that are placed in the lower-left corner are characterized by a primary objective that seeks to minimize the distribution costs. As it can be seen from the figure the routing applications are located near the diagonal that runs from the lower left corner to the upper

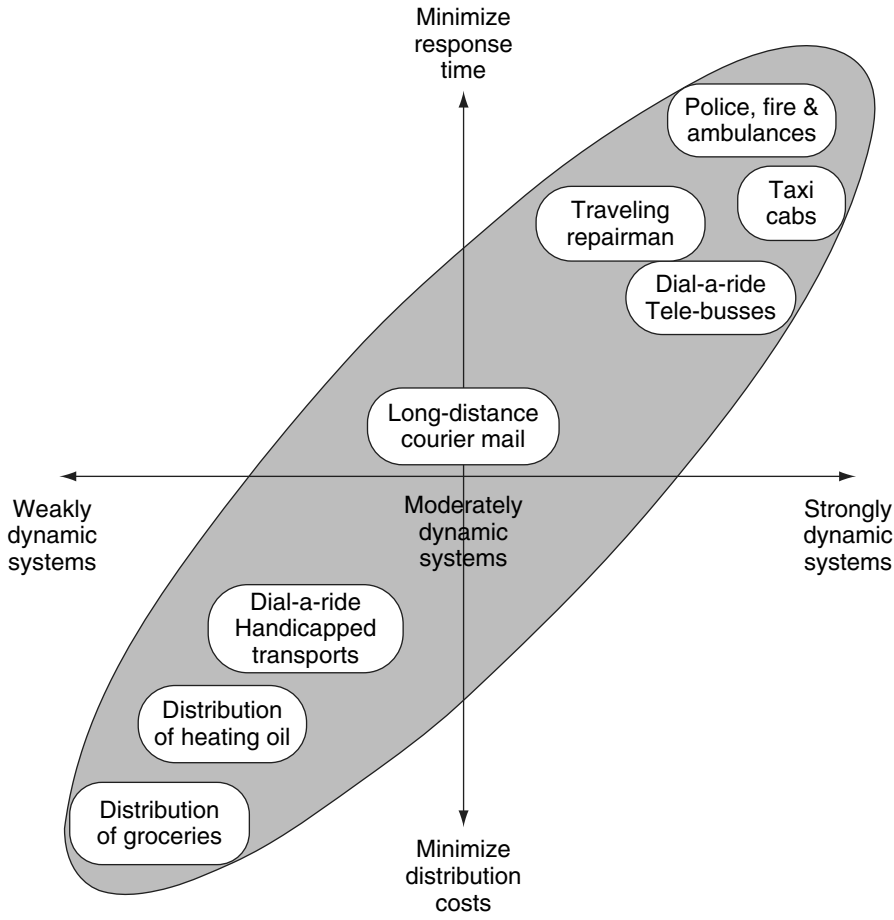


Figure 2-5. Framework for classifying dynamic routing problems by their degree of dynamism and their objective.

right corner of the Figure. This illustrates the changing trade-off between the minimizing the distribution costs and maximizing the level of service.

Naturally, a simple framework such as the one proposed here can only cover parts of the issues that may be relevant in real-life settings and the relations shown in Figure 2-5 should be handled with care. Some problem types may in fact turn out to have multiple versions. The dial-a-ride problem is one example of such an application as the so-called tele-bus version of the problem usually will be more dynamic than the transportation of the elderly and handicapped as quite different travel behaviors is seen for these groups of passengers. The tele-bus problem will in most real-life application be closer to the taxi-cab problem as both the percentage of immediate request calls and the response time will be considerably higher than seen in the case of transportation of the elderly and handicapped.

2.7 CONCLUDING REMARKS

This chapter has discussed important aspects of the dynamic vehicle routing problem. First, we provided a discussion on the wide range of special properties that dynamic vehicle routing systems possess. Then, the degree of dynamism measure for systems with and without time window constraints was discussed. Next, we discussed how the performance of an algorithm that must run in a real-time environment can be measured. Furthermore, we gave a brief discussion on some of the most important elements to consider when determining the system objective. The framework presented classifies dynamic vehicle routing systems according to their degree of dynamism and their objective. We believe that the degree of dynamism measure provides a simple way of describing the most important characteristics of a DVRP, namely the volume and the temporal composition of the immediate requests along with the system objective. These characteristics should always be considered carefully when designing a routing algorithm for a dynamic system.

Future research within this area should focus on extending the degree of dynamism measure to also cover other important problem characteristics such as the length of the service time and the demand size of the immediate requests. Naturally, finding a single measure that captures multiple characteristics will be very challenging. However, being able to capture also these characteristics within the degree of dynamism measure would potentially improve the descriptiveness of the measure.

REFERENCES

Angelelli, E., Savelsbergh, M.W.P. and Speranza, M.G. Competitive analysis for dynamic multi-period uncapacitated routing problems. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, 2005. Submitted to Networks.

Ausiello, G., Feurstein, E., Leonardi, S., Stougie, L. and Talamo, M. Algorithms for the on-line traveling salesman. Algorithmica, 29:560-581, 2001.

Bent, R. W. and Van Hentenryck, P. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research, 52(6):977-987, 2004.

Bertsimas, D. and Van Ryzin, G. A stochastic and dynamic vehicle routing problem in the Euclidean plane. Operations Research, 39:601-615, 1991.

Bertsimas, D. and Van Ryzin, G. Stochastic and dynamic vehicle routing problem in the Euclidean plane with multiple capacitated vehicles. Operations Research, 41:60-70, 1993.

Bertsimas, D. and Van Ryzin, G.. Stochastic and dynamic vehicle routing with general demand and inter travel time distributions. Appl. Probl., 25:947-978, 1993.

Brotcorne, L., Laporte, G. and Semet, F. Ambulance location and relocation models. European Journal of Operational Research, Vol. 147, p 451-463, 2003.

Fleischmann, B., Gnutzmann, S. and Sandvoss, E. Dynamic Vehicle Routing Based on Online Traffic Information. Transportation Science, 38(3):420-433, 2004.

Gendreau, M., Guertin, F., Potvin, J.-Y. and Taillard, E. Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science, 33:381-390, 1999.

Gendreau, M., Laporte, G. and Semet, F. A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. Parallel Computing, 27:1641-1653, 2001.

Ichoua, S., Gendreau, M. and Potvin, J.-Y. Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. Transportation Science, 40(2):211-225, 2006.

Jaillet, P. and Wagner, M. Online Routing Problems: Value of Advanced Information as Improved Competitive Ratios. Transportation Science, 40(2):200-210, 2006.

Larsen, A., Madsen, O.B.G. and Solomon, M. Partially dynamic vehicle routing – models and algorithms. Journal of Operational Research Society, 53:637-646, 2002.

Larsen, A., Madsen, O.B.G. and Solomon, M.The a-priori dynamic traveling salesman problem with time windows. Transportation Science, 38(4):459-472, 2004.

Larson, R.C. and Odoni, A.R. Urban Operations Research. Prentice Hall, Englewood, Cliffs, New Jersey, 1980.

Lund, K., Madsen, O.B.G. and Rygaard, J. M. Vehicle routing problems with varying degrees of dynamism. Technical report, IMM, The Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1996.

Paepe, W.E., Complexity Results and Competitive Analysis for Vehicle Routing Problems. PhD thesis, Technical University of Eindhoven, The Netherlands, 2002.

Powell, W.B., Jaillet, P. and Odoni, A. Stochastic and dynamic networks and routing. In M.O. Ball *et al*., editor, Network Routing, Handbooks in OR & MS, p. 141-295. Elsevier Science, Amsterdam, 1995.

Psaraftis, H.N. Dynamic vehicle routing. In B.L. Golden and A.A. Assad, editors, Vehicle routing: Methods and studies, pages 223-248. North-Holland, Amsterdam, 1988.

Psaraftis, H.N. Dynamic vehicle routing: Status and prospects. Annals of Operations Research, 61:143-164, 1995.

Sleator, D. and Tarjan, R.E. Amortized efficiency of list update and paging rules. *Communications of the ACM,* 28(2):202-208, 1985.

Chapter 3

DYNAMIC AND STOCHASTIC VEHICLE ROUTING IN PRACTICE

Truls Flatberg, Geir Hasle, Oddvar Kloster, Eivind J. Nilssen and Atle Riise
SINTEF Applied Mathematics, P.O. Box 124 Blindern, N-0314 Oslo, Norway, {Truls. Flatberg, Geir.Hasle, Oddvar.Kloster, Eivind.J.Nilssen, Atle.Riise}@sintef.no

Abstract: The VRP is a key to efficient transportation logistics. It is a computationally very hard problem. Whereas classical OR models are static and deterministic, these assumptions are rarely warranted in an industrial setting. Lately, there has been an increased focus on dynamic and stochastic vehicle routing in the research community. However, very few generic routing tools based on stochastic or dynamic models are available. We illustrate the need for dynamics and stochastic models in industrial routing, describe the Dynamic and Stochastic VRP, and how we have extended a generic VRP solver to cope with dynamics and uncertainty.

Keywords: Logistics; Transportation; Vehicle Routing; Dynamic; Stochastic; Optimization.

3.1 INTRODUCTION

In transportation, there is a huge potential for improvement of logistics performance through better co-ordination. Route design for a fleet of vehicles, and dynamic, real-time dispatching are both highly complex co-ordination tasks for operations of some size. Despite this fact, transportation management of today is predominantly performed by human planners and dispatchers, even in large companies. Commercial routing software with optimization functionality is implemented in industry at an increasing rate. However, such tools are predominantly used for the design of static routes. Very few cases of dynamic routing supported by advanced tools with optimization functionality are known. Moreover, the authors do not know of any commercial routing tools that are based on a model that includes the inherent stochastic nature of route planning.

In this chapter, we focus on practical approaches to stochastic and dynamic routing in the context of previous and ongoing RTD efforts at SINTEF - a Norwegian contract research institute. We describe how we have tackled the associated challenges in our own VRP solver, SPIDER, which is a component of several commercial routing tools, including SPIDER Designer from Spider Solutions AS. Most of the work presented was done through DOiT (Dynamic Optimization in Transportation), a 3-year (2004-2007) project supported by the Research Council of Norway and with the Norwegian Road Authorities as project owner.

The remainder of this chapter is organized as follows: In Section 3.2, we give a brief introduction to the dynamic and stochastic vehicle routing problem and present a focused survey of relevant literature. In Section 3.3, we describe two application examples based on cases from the DOiT consortium. Section 3.4 covers modeling and formal description of dynamic and stochastic VRPs, whereas Section 3.5 presents an architecture for dynamic and stochastic routing systems. Our suggestion for a generic and robust algorithmic approach follows in Section 3.6. Section 3.7 focuses on the task of learning statistical event models. Section 3.8 gives a summary and points to further research.

3.2 THE DYNAMIC AND STOCHASTIC VEHICLE ROUTING PROBLEM

At the core of fleet management and supply-chain coordination, there is a highly challenging optimization problem called the Vehicle Routing Problem (VRP). In broad terms, it deals with the optimal assignment of transportation orders to a fleet of vehicles, and the sequencing of stops for each vehicle that represents the formation of routes. The VRP has a large number of real-life applications and comes in many guises, depending on the type of operation, the time frame for decision making, the objective components, and the types of constraint that must be adhered to. It has been heavily studied in Operations Research (OR) since its definition (Dantzig *et al.*, 1959).

OR has been highly successful in studying idealized versions of the VRP. The classical Capacitated VRP (CVRP) is defined with a single depot, a homogeneous fleet of vehicles, and Euclidean distances. Vehicle capacity is the only constraint type. The objective reflects minimization of total transportation costs for the routing plan. It is either formulated as minimization of total distance, or a hierarchical objective where the primary goal is to minimize the number of vehicles needed, and the secondary is total distance. In this way one may optimize on a combination of fleet acquisition/depreciation costs and driving costs. For most real-life applications, the CVRP is not an adequate model, as there will be many

important additional constraints and objective components. The conventional OR approach has been to study extensions in a basically reductionistic way. A taxonomy of VRP with more or less general variants has emerged, and research has tended to focus on one of them. The most commonly studied constraint type extensions include time windows on orders.

VRP research is regarded as one of the successes of OR. The methods that have been developed have been implemented for many real life applications. A software tool industry for routing decision-support tools has emerged. However, if one aims at a wide market, routing tools must be based on generalized, rich VRP models and accompanying, robust optimization algorithms. In the past few years, research focus has shifted towards more general VRP variants. A separate sub-field of rich VRPs has emerged (Bräysy, O., *et al*., 2005a, 2005b). This can partly be explained by external forces from end users and the tool industry, as routing tools are being implemented in industry at an increasing rate. In addition, the VRP research community has taken on more challenging VRP variants, as the classical variants are now more or less regarded as being solved from a pragmatic perspective. The tremendous improvement of our ability to provide high quality results to VRPs since 1959 must be attributed to a combination of better methods and faster computers.

In its classical form, the VRP is a static and deterministic problem in the OR literature. All information is known with full precision, and it is available before problem solving starts. Even with these simplifications, the VRP is a highly demanding discrete optimization problem due to its computational complexity. The most idealized and least general variants belong to the class of NP-hard problems. For the classical VRP with only capacity constraints, methods of today can only consistently solve to optimality instances up to some 70 orders in reasonable computing time. There is little hope of substantial extensions of this limit. In general, approximation methods constitute the only viable general approach for larger-size instances and extended variants. For an excellent survey on the VRP, we refer to the book (Toth, P., *et al*., 2002).

It is clear that the classical assumptions may be drastic. By nature, the VRP is a stochastic optimization problem. It deals with future events in an environment that typically includes significant sources of uncertainty. Examples of such sources are traffic conditions that may severely influence driving time, and missing or imprecise information on order volumes or whether the customer order will materialize at all. Service times at customers may be subject to large variation. In practice, the VRP is generally a dynamic problem, as significant, new information will typically emerge as the routing plan is being executed. New customer orders, drivers calling in sick, vehicle breakdowns, early arrival and precise volumes are examples of

information that may render a current routing plan infeasible or sub-optimal. Technological advances in the form of wireless communication and global positioning systems have enabled the realization of tools for dynamic routing.

Dynamic and stochastic VRP variants have been subject to scientific study for two decades. In the past years, applied research on these very important aspects has increased. There are three main classes of models. The first class deals with a priori route planning under uncertainty. The general approach is to generate an a priori solution that has the least expected cost. The second approach involves making decisions and observing outcomes on a continuous, rolling horizon basis, but without utilizing probabilistic knowledge. The third combines utilization of stochastic knowledge with strategies for dynamic decision making and planning.

The purely Stochastic VRP (SVRP) arises when some of the elements of the problem are stochastic, as alluded to above, and one has knowledge of or makes assumptions on the probability distribution of these elements, but optimization is only performed a priori. The optimization criterion often relates to minimizing the expected value of some definition of cost. SVRPs are usually modeled as mixed or pure integer stochastic programs, or as Markov decision processes. We refer to the survey papers of Dror *et al*. (1989), and Gendreau *et al*. (1996).

Psaraftis (1988) defines the Dynamic VRP (DVRP) as follows: *A VRP is dynamic if information on the problem is made known to the decision maker or is updated concurrently with the determination of the set of routes.* By contrast, if all inputs are received before the determination of the routes and do not change thereafter, the problem is termed static. The pure DVRP is focused on the dynamic revision of a routing plan as new information arrives, without utilizing knowledge on stochastic elements. For details on the DVRP, see the surveys by Psaraftis (1988, 1995), Lund *et al*. (1996), Powell *et al*. (1995) and the PhD thesis by Larsen (2000).

In the following, we define the dynamic and stochastic vehicle routing problem (DSVRP) as a VRP for which some parts of the problem definition are not available until plan execution. Thus, "Dynamic" means that new information becomes available during plan execution, and that planning therefore is done more or less continuously to handle these changes. This is sometimes referred to as "on-line" transportation planning. All dynamic problems are of course stochastic, in the sense that the problem updates are not known in advance. However, in the context of the DSVRP, the label "stochastic" signifies that the problem definition includes some information about the probability of occurrence and nature of some or all future problem updates. This information may be exploited by solution algorithms to anticipate expected problem updates.

The Dynamic and Stochastic VRP (DSVRP) has been studied in Ichoua *et al*. (2005), Bent and Van Hentenryck, (2003, 2004), and Hvattum *et al*., (2006). Current work by the authors, which is the backdrop for this chapter, belongs to this type of approach. For a focused literature survey, we refer to Flatberg *et al*. (2005).

3.3 APPLICATION EXAMPLES

To show how dynamic and stochastic aspects show up in the real world applications, we present two examples. One example involves the transport of goods, while the other deals with person transportation. In both examples, proper handling of dynamic events is essential for a successful operation.

Schenker Linjegods AS is a major distribution company that services all of Norway. Goods are collected at customers and brought to one of several terminals, where they are registered and measured. Long distance goods are then routed, mostly by rail, to other terminals, whence they are delivered to the destination. At each terminal, a fleet of vehicles is employed to collect and deliver goods in the surrounding area, and we can formulate a VRP for their operation.

We focus on the operations at Alnabru terminal, which services Oslo and the surrounding areas. This terminal employs around 100 vehicles that make 3500–3800 deliveries and 450–550 pickups in a day. Most vehicles start their day delivering goods that have arrived at the terminal during the night, while a few start collecting right away. Later in the day, the focus shifts to collecting goods. All goods have to be brought to the terminal before a set deadline for routing to other terminals during the night, and a fair amount must be brought in earlier to avoid congestion at the terminal close to the deadline.

Most of the customers are regular, which means that they will be visited every day for collecting goods. The rest of the customers must order each pickup by phone or e-mail. If the order is placed before noon, it will be collected the same day — otherwise it will be collected the next day. The amount of goods must be given when placing an order.

There are several aspects of this situation that require modeling as a dynamic and stochastic problem. Until noon, new orders may be placed that must be accommodated in today's plan. The amount of goods to be collected at regular customers may vary considerably from day to day, and the actual amount for a given day is often not known before arriving at the customer. Regular customers are encouraged to report in advance when the amount differs significantly from the usual amount, but may fail to do so. Some customers may be serviced only using small vehicles, while others may require a lift or other equipment on the vehicle. When arriving at a new

customer, it may be discovered that the present vehicle cannot be used, and re-planning is required. Travel times vary in ways that cannot be reliably predicted, e.g. depending on the level of rush traffic on a given day.

Nor-Link is a Norwegian owned software company that delivers solutions for *personalized on-demand transportation*. Typical uses of their system are for dial-a-ride and door-to-door transportation of elderly or disabled people, non-emergency medical patients or schoolchildren. To achieve an efficient utilization of the fleet, it is important to make plans that can combine orders on the same tour, reducing both the number of vehicles needed and the total distance traveled.

In a typical operation, a fleet of vehicles is to serve a set of orders, each order being either a departure type order or an arrival type order. For the departure type, the customer specifies a time for pickup along with the pickup and delivery location. For the arrival type, the customer specifies when he must reach the delivery location. For both types there are rules that limit how far from the given times the operator can serve the order. In addition there may be restrictions on the total time spent in the vehicle for each customer.

In a normal day-to-day operation some customers are regular and some are known days in advance, while others arrive during the day's operation. The problem does not have as many dynamic and stochastic aspects as the previous example, but there are similar aspects that require this problem to be handled as a dynamic and stochastic problem. New orders arrive continuously and must be accommodated in today's plan, within a response time of a few seconds. Customers may require changes to previously registered orders with regard to pickup and arrival times. Some customers have special needs concerning equipment and facilities, e.g., require a wheelchair lift. Upon arrival it may be discovered that the assigned vehicle is unable to serve the order, and re-planning is necessary. For operations in congested areas, it is important to have plans that are robust with regard to varying travel times.

To handle new requests fast, it is important to have plans that easily can accommodate new orders. Thus, plans based on the already known orders should incorporate enough slack to handle the expected number of new orders. As the number of customers is limited and each customer has a history with more or less regularity, it should be possible to make predictions of where the slack should be placed.

3.4 A FORMAL DESCRIPTION OF DYNAMIC AND STOCHASTIC VRPS

A formal description of a DSVRP is useful for testing algorithms and communicating benchmark problem instances, thus enabling different researches to work on the same problems and compare results. The model should be able to describe the full complexity of real world applications, yet be intuitive and easily exchanged with other researchers.

In our model, a DSVRP is defined by the following items:

1. The initial VRP, as known prior to all planning
2. The set of dynamic updates (events) that occur during planning and plan execution
3. Probabilistic knowledge about possible future problem updates
4. The commitment strategy

The *initial VRP* is the definition of the VRP instance as it is known before planning and execution begins. To cover a wide variety of real world applications, the VRP model must be a rich extension of the classical CVRP. For the purpose of benchmarking, the initial VRP must also define the topology that gives travel times and distances. Ideally, one should include a definition of a complete road network, including relevant physical aspects and time dependent travel times for each road segment. However, this is both impractical and involves legal issues relating to the ownership of such data. For the time being, therefore, only Euclidian and table topologies are supported, and all travel times are static.

The second item in a DSVRP is the set of dynamic updates. We use the term *event* to mean a dynamic change in the VRP that becomes known during planning and plan execution. Events may represent any kind of change in the problem definition. The types of events that are present in a DSVRP is determined by which aspects that are considered to by stochastic in that particular problem. The most common types, which are found in the examples in Section 3.3, are:

- the occurrence of new orders (with given properties)
- order updates, including changes in demand, time windows, order-vehicle compatibilities, service time, etc.
- travel time updates, in case of delays or on the basis of an updated prognosis

Our current method is currently used to handle the stochastic occurance of new orders. Stochastic order updates and travel time updates are also addressed, but due to reasons discussed in Section 3.6 these are handled in a

simplified manner. Figure 3-1 shows the information structure of an event. Each event has a *trigger*, which indicates the reason for the information in the event to become available. In general, we can divide events into two groups: Those triggered by the execution of the plan (such as arrival at a customer), and those that are independent of the plan (and simply triggered by the passing of time). Each event also has an action type, such as e.g. "New order", "Change order", or "Cancel order". In addition, the necessary updated information for the VRP element in question must be supplied, using the same format as in the initial VRP description.

As mentioned, a motivation for using a standardized formal description format is the ability to exchange problem instances with other researchers. There is, however, a point to be aware of. Events that are independent of the plan and its execution, such as the arrival of new requests, may easily be included in a formal description of a DSVRP. For events triggered by the execution of the plan, the situation is more complicated. For instance, changes in the demand or service time of a customer may be discovered as a vehicle serves the customer in question. Had the sequencing of customers in the plan been different, then this event would have arrived at another point in time. If the plan did not serve that particular customer, the event would never occur. This plan dependency introduces some complications when benchmarking solvers, since the events received by a solver at any time during planning will be dependent on the plan that this particular solver has produced at that time. The total problem definition as it is known at the end of the planning period will therefore depend on the solution method. It is important to keep this dependency in mind when performing empirical comparison between methods. The issue may also be addressed explicitly by the solution method itself, by using probabilistic information to minimize the costs of future stochastic events.

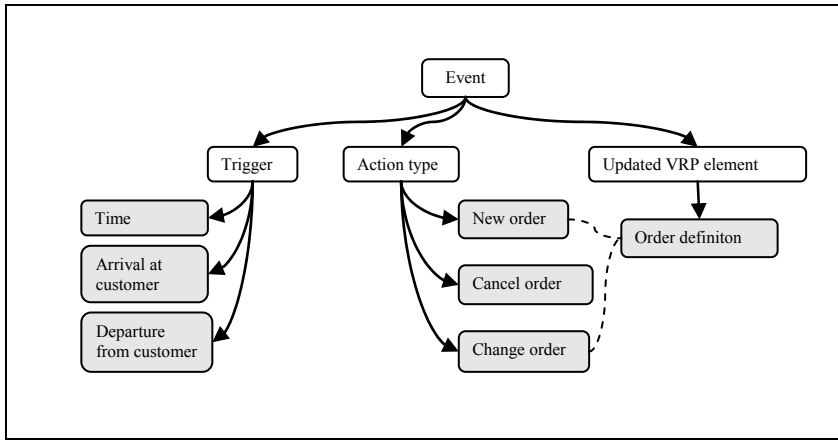


Figure 3-1. Elements and possible values for dynamic events. Shaded boxes represent values or data.

The third item in a DSVRP is knowledge about future events. This knowledge is typically based on statistics over past experience, represented as (conditional) probability distributions. The knowledge may be available to the solver as a software black box, from which it is possible to extract probabilistic knowledge about the occurrence and nature of future events.

The type of information that one would want to extract from such a black box depends on the solver's algorithm. This will influence the black box software interface, as well as the internal structures in which the information is stored. It is therefore difficult to find a unified and standard way of representing such knowledge as a part of a DSVRP benchmark problem instance. We therefore resort to a simpler solution. Each benchmark instance includes the full history from which the statistical knowledge may be extracted. That is, together with the actual DSVRP definition, we supply a collection of "historical background DSVRP cases". The designer of the individual solver may then build and exploit the statistical knowledge that will best serve his or her algorithm.

Finally, the DSVRP description contains a commitment strategy. The strategy determines when requests are irrevocably assigned to vehicles, and what parts of the plan may still be altered by the DSVRP solver. Commitments will typically mirror the actual dispatching of customers to vehicles during planning. A common commitment strategy says that a fixed number of the future requests served by each vehicle are committed, and may therefore not be reassigned by the solver.

For examples of full DSVRP descriptions, including historical background cases, see the VRP benchmark pages located at http://www.top.sintef.no.

3.5 A DYNAMIC AND STOCHASTIC VRP SOLVER

We now turn our attention to solving the DSVRP. First, we consider the context in which a solver will operate.

3.5.1 Overall Architecture

Other authors have suggested system architectures for DSVRP planning applications and other applications involving dynamic routing problems (see e.g. Regan *et al*., 1998; Fleischmann *et al*., 2004). In this section, however, we will limit our scope to the immediate context in which a DSVRP solver will operate. Our overall architecture is shown in Figure 3-2.

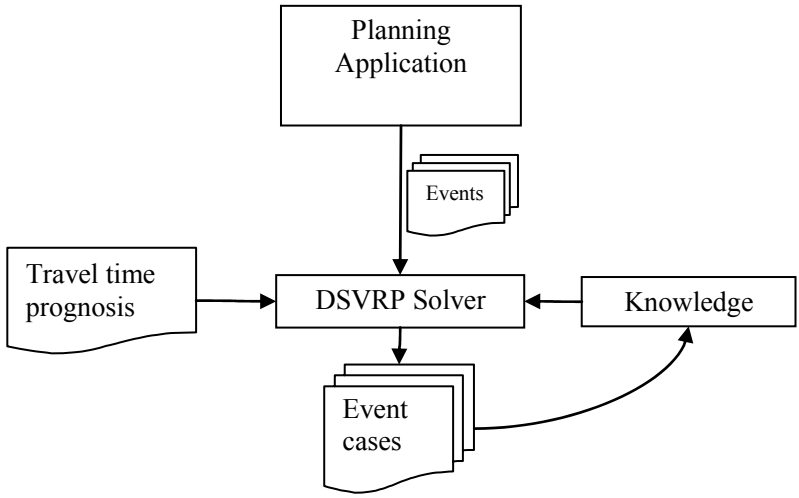


Figure 3-2. The context in which a commercial DSVRP solver may operate.

The DSVRP solver in the figure is a software component with a well defined API that can be exploited by a higher level Planning Application or fleet management system. The Planning Application is the tool that a transportation planner or dispatcher uses in daily dynamic transportation planning. During dynamic planning, the DSVRP solver receives the initial VRP definition and reacts to dynamic events from the Planning Application. Resulting and updated plans are returned to the Planning Application. For some applications, the solver will be an integral part of the Planning Application.

To enable the DSVRP solver to use statistical knowledge about the occurrence and properties of future events, a real world application must establish such knowledge from recorded history. A DSVRP solver may contribute to this by recording information about the actual DSVRPs that it solves, including both the initial VRP definitions and the dynamic updates in the form of events. The event case files in Figure 3-2 represent such storage, typically logging one day's problem definition to each file. Such files may then be used by a knowledge module to build or update statistical knowledge about the transporter's daily operations. Finally, the results of this learning process may be exploited by the solver during optimization.

A DSVRP solver may also use travel time prognosis from some external system. The travel time may typically be time dependent and valid for each segment in the road network. This information may be static or updated dynamically. Statistical information about the stochastic nature of travel times may or may not be available.

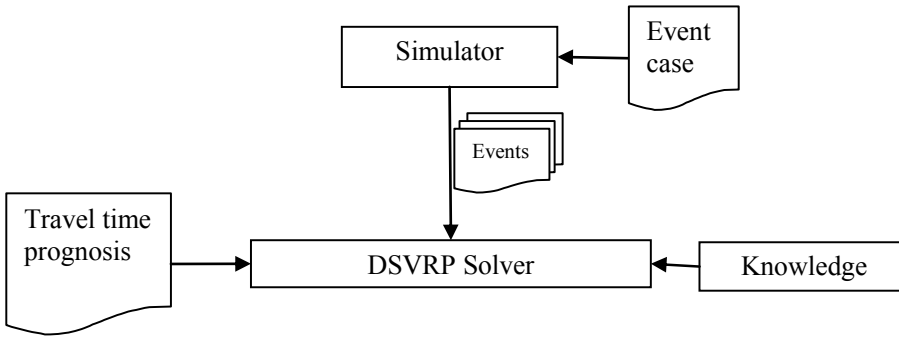


Figure 3-3. The DSVRP Solver in a simulation setting.

When developing and testing the optimization algorithms in a DSVRP solver, it is necessary to test the algorithms on a large number of cases. Using the Planning Application for this purpose is not practical. Instead, we may embed the solver in a simulation framework, as shown in Figure 3-3. Here, a discrete event Simulator takes on the role of the Dispatcher and his/her Planning Application by:

- Defining the initial VRP
- Running the DSVRP solver continuously to improve the currently accepted best plan
- Enforcing a commitment strategy by simulating vehicle dispatching
- Simulating vehicle activities
- Triggering events (new requests, updates to known requests, etc.)

The Simulator typically acts according to the information in an input event case file, identical in format to those that are logged by the DSVRP Solver during learning in real world transportation optimization. This makes it possible to test DSVRP solvers on problem instances that are extracted from real world operations. The Simulator may speed up simulation time, so that a case for a whole day, say, may be run in much less time, thus making testing more efficient. Of course, the response of the DSVRP solver will limit the speed at which the Simulator may run.

In the following, we will take a closer look at special requirements for the DSVRP Solver and the knowledge learning module.

3.5.2 Requirements

Commercial VRP solvers support a wide range of VRP variants. Even the classical CVRP is very hard to solve, and additional constraints and objectives usually add to the complexity. Such solvers therefore usually apply heuristic algorithms to yield usable solutions within the expected

response times. Still, commercial solvers in general solve only static vehicle routing problems. Taking the step to include dynamic, on-line optimization involves new challenges. On the main application level, the user interface must handle a more dynamic interactive dialogue than is the case in a static planning situation. As events occur, the dispatcher will need to change the current plan accordingly, usually aided by the underlying solver. The response requirements for this kind of recovery and re-planning are usually much tighter than is the case is for a typical static (off-line) planning situation. Also, in the periods between events, the DSVRP solver may work in the background to improve the current plan. The Planning Application may impose a desired frequency with which the dispatcher is ready to consider new and better plans from the underlying DSVRP solver, as well as requirements on how much better, and/or how different a new plan from the solver must be to be acceptable.

How these things are handled, both in the user interface and in communication with the underlying solver, depends on the commitment strategy and will vary from application to application. E.g., in a courier company, the situation will be very dynamic, and the dispatcher will need to make commitment decisions in a matter of seconds. This puts a lot of pressure on the underlying solver, which will have to respond based on a fast and simple decision method. On the other hand, couriers will typically be dispatched on a "least commitment" basis, so that the solver may quite freely change the future plan after, say, the next request that is planned for each courier. A more standard distribution company will have a different profile, in which the time between events is larger and the solver has more time to optimize the plan. This optimization may however be more restricted than in the courier case, if a larger part of the plan has already been dispatched to the drivers, who may not look kindly to frequent changes in their schedule. To enforce commitment, the solver must support locking parts of the current plan that should not be modified during further optimization. It may also be relevant for the solver to optimize future plans while striving to make as few changes as possible compared to the current plan.

The Knowledge module in Figures 3-2 and 3-3 must fulfill certain basic requirements. Firstly, it must learn from past experience based on event case log files or similar. Secondly, it must supply a DSVRP Solver with the appropriate information about its knowledge on request. To achieve these goals, the module must store the resulting knowledge in a way that facilitates later updating and access of the information. As the probabilistic information is often conditional, the chosen data structure should be effective with respect to building, maintaining and retrieving conditional probabilities. In Section 3.7 we describe in some detail how such a knowledge/learning module may be designed.

3.5.3 The SPIDER DSVRP Solver

The SPIDER library is a commercially available DSVRP Solver. SPIDER also serves as the technical platform for our VRP research, and is one of the few commercial solvers that are responsible for best known solutions to some of the standard VRP benchmark problems. Originally designed to handle rich variants of the static VRP, this software component has over the years been extended to handle a wide variety of transportation problems. This includes problems with pickup-and-delivery orders, service orders, heterogeneous fleet, work hour regulations, plan dependent service times, driver dependent service times, bulk transport orders with compartment constraints and adjustable order quantities, multiple time windows, alternative pickup/delivery locations, vehicle dependent road topologies, etc.

We first consider how static versions of these rich extensions to the VRP are solved. The inherent complexity of these problems has led us to choose a unified meta-heuristic approach in SPIDER. All problem instances are basically solved by the same algorithmic machinery, although the use and configuration of individual operators and other search mechanisms may take advantage of characteristics of the instance at hand.

A given static VRP is solved by first constructing an initial solution. Alternative solution constructors with different strengths are available. The initial solution is then improved by a Iterated Local Search (Lourenço *et al*., 2003) procedure, with phases of *intensification* and *diversification* (See Figure 3-4). Intensification is achieved through Variable Neighborhood Descent (Hansen and Mladenovic, 1999), using a selection of well known intra-tour and inter-tour operators, some of which have been extended or created to accommodate the richness of the SPIDER VRP model. The list of operators includes Insert, Relocate, 2-opt, 3-opt, Exchange, Cross, Change locations, Change Time Window, etc. When a local optimum has been found, several diversification mechanisms may be invoked to "jump" to an unexplored, promising part of the search space.

```
OPTIMIZEPLAN( p, stop)
Input: p = the plan to be optimized, stop = a stopping criterion
Return value: The improved plan
while ! stop.ISSATIFIED()
  p ← LOCALSEARCH(p)
  p* ← CHECKFORNEWBESTSOLUTION(p)
  p ← DIVERSIFY(p)
return p*
```

Figure 3-4. The overall search strategy for improving a given input plan.

Travel times, distances and costs are computed by SPIDER's topology module, based on a digital model of the road network. The module is able to model relevant aspects such as one way roads, speed limits, turning restrictions, height/weight restrictions and rush hours. SPIDER's model handles travel times that are time dependent (and not necessarily symmetric).

To solve the DSVRP, SPIDER uses a scenario based approach to sample the space of possible future events. This approach is presented in Section 3.6. For solving each scenario, the same search framework is used as for static VRPs (Figure 3-4). Dynamic problem updates are handled by incorporating the concept of events, as defined in Section 3.4, and interrupting the search each time such an external event occurs. To be able to learn and exploit statistic information about some problem, SPIDER incorporates a knowledge module that satisfies the requirements described above. SPIDER may record events to event case files, use the knowledge module to learn from the historical records, and then utilize the resulting statistical knowledge when generating new robust plans for the DSVRP. For more details about the knowledge module in SPIDER, see Section 3.7.

3.6 A ROBUST APPROACH TO DYNAMIC AND STOCHASTIC VRPS

For the task of producing optimized plans for DSVRP, authors have previously chosen to deal with dynamic problem updates in different ways. Some algorithms are purely reactive, such as simple assignment rules (Larsen *et al*., 2002; Yang *et al*., 2004), insertion heuristics (Madsen *et al*., 1995), vehicle positioning (Larsen *et al*., 2002), vehicle diversion (Ichoua *et al*., 2000), parallel methods (Attanasio *et al*., 2004), double-horizon approaches (Mitrovic-Minic *et al*., 2004), waiting strategies (Mitrovic-Minic and Laporte, 2004), problem decomposition (Giaglis *et al*., 2004), and more elaborate branch-and-price-algorithms (Savelsbergh and Sol, 1998). Others utilize probabilistic knowledge about future problem updates to produce more robust plans. This includes positioning of vehicles (Horn, 2000), Markov decision processes (Thomas and White III, 2004), waiting strategies (Ichoua *et al*., 2005), scenario-based methods (Bent and Van Hentenryck, 2003, 2004; Hvattum *et al*., 2006; Van Hentenryck *et al*., 2006), and stochastic programming (Powell and Topaloglu, 2003). Further details of the above methods are described in (Flatberg *et al*., 2005).

One robust and appealing method of approximating the best solution for a DSVRP is to base the algorithm on the repeated optimization of a set of scenarios, where each scenario includes possible future events that are drawn from a known probability model of stochastic events. The method is a natural extension of our method solving the static VRP, and different

stochastic aspects of the DSVRP may be incorporated in a unified manner. A DSVRP solution can be viewed as a series of order allocation decisions. Ideally, we would want to evaluate each decision at a given time step by solving all possible scenarios resulting from that decision. However, the available computation time makes this approach infeasible. Instead, we use a consensus approach to choose a plan that approximates the optimal expected objective value. Each plan that results from the optimization of a scenario is added to a pool of plans. The plan being returned to the user is the plan that most resembles the other plans in the pool. This strategy is more robust than e.g. selecting the plan from the pool that best minimizes the objective function (Bent and Van Hentenryck, 2004), since the latter will favor scenarios with "cheaper" plans that are not necessarily more probable than more "expensive" scenarios. For a comprehensive theoretical analysis of the consensus approach, see (Van Hentenryck *et al*., 2006).

Our optimization algorithm utilizing the consensus approach is presented in Figure 3-5. The input plan is typically the last plan that was returned to the user, but may be modified e.g. when the user has made some dispatching decision that locks parts of the plan. The input problem definition may also have changed as a result of outside events, such as new customer requests. The function UPDATEPOOL in line 1 repairs or deletes plans in the pool that are not consistent with the input plan and problem definition. This can happen if vehicles in the input plan are committed to certain requests, but a plan in the pool has assigned these vehicles to other requests, or if a customer's demand was increased, making some plans infeasible with respect to capacity constraints. Then the algorithm iterates over all plans in the pool (lines 2–10), as long as it is not interrupted by outside events. In line 5, a scenario is created, in this case by sampling a new set of requests and including these in the VRP to be optimized. Line 6 optimizes the plan of the scenario using the local search framework described in Figure 3-4. The time allowed for optimization, τ, may depend on the pool size and the time spent in the last iteration. Sampled requests are removed from the plan in line 7. The improved plan may be accepted in line 8 if its optimization process is not interrupted too early by some external event, or if the new plan is better or much different from the other plans in the plan pool. Accepted plans are added to the pool in line 9, without deleting the original plan. Thus, the pool of plans is continuously growing and shrinking. Line 10 calculates a new consensus plan and notifies the calling application if this plan is better than the input plan, *p*. Note that this may not depend only on the plan's objective value but may e.g. depend on the degree of statistical support this plan has in the current plan pool. Thus a plan may be reported which is not better than *p* in objective value, but which is more probable to be robust under future events.

IMPROVEPLAN(p, P, n, V, R, t)
Input: current plan p, plan pool $P = \{p_i\}$, consensus length n, vehicles V, requests R, current time t
1. UPDATEPOOL(P, p)
2. **while** (! STOPOPTIMIZER())
3. $m \leftarrow |P|$
4. **for** $i \leftarrow 1 \ldots m$
5. $R \leftarrow R \cup$ SAMPLENEWREQUESTS(t)
6. $p' \leftarrow$ OPTIMIZEPLAN(p_i, TIMEOUT (τ))
7. $R \leftarrow$ REMOVESAMPLEDEVENTS(R)
8. **if** (ACCEPT(p'))
9. $P \leftarrow P \cup p'$
10. REPORTANYNEWCONSENSUSPLAN(P, n, V, R)

Figure 3-5. Improvement of a plan for a DSVRP, using scenarios with sampled events to handle the dynamism and stochasticity of the problem.

The algorithm for choosing the consensus plan is based on plan similarity. A simple similarity score is calculated for efficiency reasons. First, a count matrix C is created, where each entry C_{vrk} is given by

$$C_{vrk} = \sum_{p=1}^{|P|} \delta_{vrk}^{(p)}$$

Here, $|P|$ is the size of the plan pool. $\delta_{vrk}^{(p)}$ equals 1 if vehicle v visits customer r as the k'th next visit following the locked parts of the tour for vehicle v in plan p. Otherwise $\delta_{vrk}^{(p)}$ equals 0. Thus, C_{vrk} is a count of how many plans in the pool have vehicle v going to customer r as the k'th next visit. Adding 1 to the count for every such plan can easily be extended to a more general formulation, where some function of the objective value of the plan is added as a reward instead. Let n be the consensus length, i.e. the number of visits to consider. Let $r_k^{(p,v)}$ be the k'th next visit for vehicle v in plan p. We may then assign a similarity score s_p to each plan $p \in P$ as follows:

$$s_p = \sum_{v=1}^{|V|} \sum_{k=1}^{n} C_{vr_k^{(p,v)}k}$$

The plan with the highest score is chosen as the consensus plan. Whether this plan should replace the plan to be executed may be decided based on the relative objective value and/or the relative consensus score of these two plans.

The presented algorithm is computationally intensive, but reasonably good plans can at any time be reported back to the user because the algorithm can run continuously in the background. It is also noted that a parallel implementation would be straightforward, using the available processors to distribute the task of optimizing the scenarios. The methodology of using scenarios and a consensus plan has proved to be a useful strategy for solving the DSVRP. It is robust with regard to the level of dynamism and stochasticity of the problem. It is also robust with regard to the probability models being used, in the sense that plans are acceptable also when the statistics are noisy due to e.g. errors in the underlying assumptions or errors in the recorded historical data used for learning the models (Bent and Van Hentenryck, 2004).

For the applications presented earlier, we gather statistics not only for the arrival rate and distribution of new requests, but also for modifications of requests from regular customers. This can include a change in the size of the goods to be picked up, a change in the time window for servicing the request, and so forth. Including samples of such events in the scenarios is theoretically possible, but a key difference to sampled new requests is that such modification events may result in problems that are easier rather than harder than the original problem (i.e., a reduced size of the goods results in more available vehicle capacity). A plan for such a scenario may be illegal when removing the sampled events and applying the plan to the original problem (i.e., the vehicle capacity is exceeded because too many orders were assigned to the vehicle, the orders being reduced by sampled modification events). In such cases we choose instead to apply our knowledge of modification events as a preprocessing step. By examining the statistical distribution of e.g. the request size of a regular customer, we fix the request size to the value s so that the probability of the actual size being smaller than s equals some predefined level p. A similar method is used to handle the stochastic nature of road network travel times.

3.7 LEARNING EVENT MODELS

As discussed in Section 3.4, events are dynamic updates to a VRP instance that become available during plan execution. The previous section explained how knowledge about such events can be exploited to produce more robust plans to the VRP, by sampling different scenarios. We now consider the representation of statistical knowledge of events, and how this knowledge can be learned automatically from past experience.

Probabilistic models of stochastic events are gathered from logged events, building upon both domain knowledge and historical data. The models are sampled to generate stochastic event cases with realistic

properties for test purposes, or utilized by the optimization algorithm to draw stochastic samples for scenarios. In addition, a client implementing our framework may want to study the learned models to discover e.g. suboptimal behavior among the regular customers. The probabilistic models represent attributes of events; arrival rates, the geographical distribution of new requests, goods sizes, time windows, modifications to existing requests, call-in times, and so forth.

3.7.1 Bayesian Networks

The probabilistic models used to represent stochastic events are based on the framework of *Bayesian networks* (Heckerman, 1995) in order to capture possible correlations among the variables. A Bayesian network encodes the joint probability distribution of a set of variables as a directed, acyclic graph with conditional independence assumptions. The nodes of the graph represent the stochastic variables, while edges represent local conditional dependencies; the set of nodes having a directed edge to a node *n* represent the set of parent nodes that *n* depends upon. A network is completely specified by its nodes, edges, and probability parameters associated with each edge (in case of discrete variables, each edge parameter is a conditional probability table listing the probability that the child node assumes a value given the parent node values). Both the network structure and all parameters may be set manually, learned from historical data, or found by a combination of the two.

Learning a Bayesian network from historical data may be divided into four categories, depending on the initial model uncertainty and the quality of the observed data. In increasing order of difficulty, we have 1) Known network structure, full observation, 2) known structure, partial observation with missing data, 3) unknown structure, full observation, and 4) unknown structure, partial observation. The easiest case amounts to simple counting of the observed states, often including some prior biases. If data is missing for some nodes in the observation, an iterative Expectation-Maximization algorithm is typically applied to find a locally optimal set of network parameter values. For the harder case of having an unknown network structure, the common approach is to define a network scoring metric which measures a tradeoff between network complexity and the ability to describe the observed data. Then a local search in the landscape of possible networks is performed using the metric and move operators well-known to the VRP community (relocate, cross, etc.).

Once a Bayesian network has been constructed, random samples may be drawn by topologically ordering the nodes and generating values so that all values of the parent nodes are determined before determining the value of a child node.

3.7.2 Modeling of Stochastic VRP Events

We use domain knowledge to set up the structure of Bayesian networks that model the dependencies of the stochastic variables of interest. In the following, we will address the application examples described in Section 3.3, most notably the first application involving both new requests and modifications to known requests from regular customers. All variables are discretized. The parameters are learned from a database of event cases, initially assuming known structure and full observation. The geographical area of interest is split up into regions, and for each region and each day of the week we define three models representing the arrival rate of new requests, the attributes of new requests, and the attributes of modifications to existing requests for regular customers. The models are shown in Figures 3-6 – 3-8.

The first model relates the number of new requests for the remaining part of the day (*N'*), the time of day (*T*), and the number of requests observed so far (*N*). The time is discretized to a few intervals in a day. If e.g. the past history shows that an unusually low number of requests by noon on Mondays indicate a low total number of requests that day, while a low number of requests on Fridays indicate a delayed but not lower demand from customers, then this model will capture such differences.

Request attributes are modeled by the network shown in Figure 3-7. Many variables are thought to correlate with the time window center T_c: The time window width T_w, the size of the goods S, the service time T_s, and the number of minutes the call-in time precedes the time window, T_b. Attributes required by the vehicle servicing the request, V_a, may correlate with the size of the goods (e.g., whether a van or truck is needed).

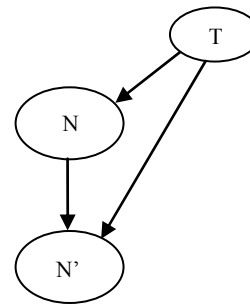


Figure 3-6. Bayesian network representing the arrival rate of new requests.

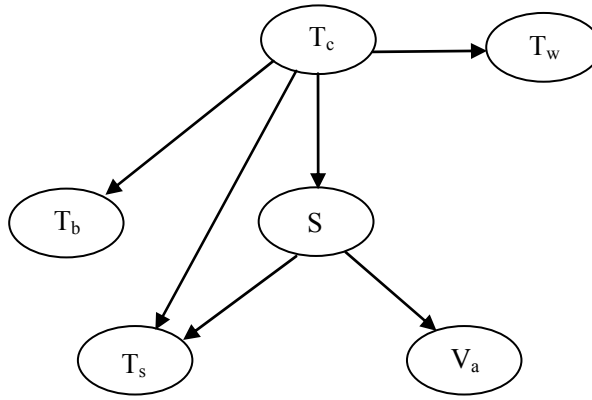


Figure 3-7. Bayesian network representing attributes of new requests.

Finally, given a known request from a regular customer, modifications to the request may be modeled by the network shown in Figure 3-8. Here, f_S is a discretized multiplication factor relating the requested size and the modified size (1 if no change, less than 1 if the updated request size is reduced, and 0 if the request is cancelled). V_a represents a modification to the vehicle attributes, and T_b represents the number of minutes before the time window center the new information is provided (discretized to a few intervals).

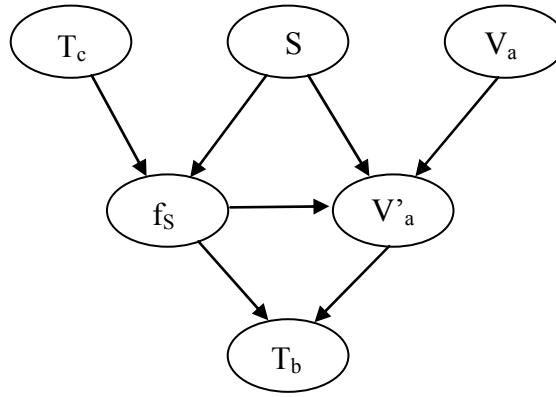


Figure 3-8. Bayesian network representing attributes of modifications to existing requests for regular customers.

3.8 CONCLUSIONS AND FURTHER RESEARCH

Huge economical and environmental effects may follow from implementation of optimization based route planning tools. A critical factor is the adequacy of the underlying VRP model. In this chapter, we have illustrated and exemplified the need for dynamic planning capabilities, as well as the need for an explicit representation of uncertainties that are inherent in many routing applications. We have described a rich model for dynamic and stochastic VRPs that is suitable for a generic, industrial routing tool. Furthermore, we have proposed an architecture for dynamic routing systems, and described a scenario-based, robust algorithmic approach for solving the DSVRP. Finally, we have focused on the important aspect of learning and modeling the uncertainties of real-life routing applications.

The classical static and deterministic VRP is a computationally hard problem. For the large-size instances that are typically found in industrial applications, heuristics is the only viable generic approach. It is clear that adding dynamics and uncertainty to the problem will typically compound its computational complexity.

We have concluded that an algorithmic approach based on scenario generation is robust, in the sense that it may accommodate uncertainty in most elements of the VRP. This approach effectively reduces the DSVRP to a number of static and deterministic VRPs. Major issues of further research are related with the overall computational efficiency of this approach, methods for representation and learning uncertainty, and the tradeoffs between response time, the number of scenarios, and the optimization time for each scenario. Finally, a highly important area of further research concerns the practical effects of using dynamic routing tools that are based on a stochastic VRP model.

REFERENCES

Attanasio, A., Cordeau, J. F., Ghiani, G. and Laporte, G., 2004, Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem, *Parallel Computing* **30**(3)**:** 377-387.

Bent, R. and Van Hentenryck, P.,2003, *Dynamic Vehicle Routing with Stochastic Requests*, Technical Report, Technical Report, Department of Computer Science, Brown University.

Bent, R. and Van Hentenryck, P., 2004, Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research* **52**(6)**:** 977-987.

Bräysy, O., Gendreau, M., Hasle, G. and Løkketangen, A.,2005a, *A Survey of Heuristics for the Vehicle Routing Problem, Part I: Basic Problems and Supply Side Extensions*, SINTEF Report, Oslo, Norway.

Bräysy, O., Gendreau, M., Hasle, G. and Løkketangen, A.,2005b, *A Survey of Heuristics for the Vehicle Routing Problem, Part II: Demand Side Extensions*, SINTEF Report, Oslo, Norway.

Dantzig, G. B. and Ramser, J. H., 1959, The truck dispatching problem, *Management Science* **6:** 80.

Dror, M., Laporte, G. and Trudeau, P., 1989, Vehicle routing with stochastic demands: properties and solution frameworks, *Transportation Science* **23:** 166-176.

Flatberg, T., Hasle, G., Kloster, O., Nilssen, E. J. and Riise, A.,2005, *Dynamic and Stochastic Aspects in Vehicle Routing - A Literature Survey*, SINTEF Technical Report STF90A05413.

Fleischmann, B., Gnutzmann, S. and Sandvoß, E., 2004, Dynamic Vehicle Routing Based on Online Traffic Information, *Transportation Science* **38**(4)**:** 420-433.

Gendreau, M., Laporte, G. and Sèguin, R., 1996, Stochastic vehicle routing, *European Journal of Operational Research* **88:** 3-12.

Giaglis, G. M., Minis, I., Tatarakis, A. and Zeimpekis, V., 2004, Minimizing logistics risk through real-time vehicle routing and mobile technologies: Research to date and future trends, *International Journal of Physical Distribution & Logistics Management* **34**(9)**:** 749-764.

Hansen, P. and Mladenovic, N.,1999, An Introduction to Variable Neighborhood Search, in: *Metaheuristics, Advances and Trends in Local Search Paradigms for Optimization*, S. Voss *et al*, eds., Dordrecht, Kluwer.

Heckerman, D.,1995, *A Tutorial on Learning Bayesian Networks*, Technical Report, Microsoft Research.

Horn, M. E. T., 2000, Fleet scheduling and dispatching for demand-responsive passenger services, *Transportation Research Part C* **10**.

Hvattum, L. M., Løkketangen, A. and Laporte, G.,2006, *A branch-and-prune heuristic for stochastic and dynamic vehicle routing*, Working paper.

Ichoua, S., Gendreau, M. and Potvin, J.-Y., 2000, Diversion Issues in real-time vehicle dispaching, *Transportation Science* **34**(4)**:** 426-438.

Ichoua, S., Gendreau, M. and Potvin, J.-Y., 2005, Exploiting knowledge about future demands for real-time vehicle dispatching, *Transportation Science* **Forthcoming**.

Larsen, A.,2000, *The Dynamic Vehicle Routing Problem*, PhD Thesis, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark.

Larsen, A., Madsen, O. and Solomon, M., 2002, Partially dynamic vehicle routing - models and algorithms, *Journal of the Operational Research Society* **53:** 637-646.

Lourenço, H. R., Martin, O. C. and T., S.,2003, Iterated Local Search, in: *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, eds., Kluwer, pp. 321-354.

Lund, K., Madsen, O. B. G. and Rygaard, J. M.,1996, *Vehicle routing problems with varying degree of dynamism*, Technical Report, IMM, Department of Mathematical Modelling, Technical University of Denmark.

Madsen, O., Ravn, H. F. and Rygaard, J. M., 1995, A heuristics algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives, *Annals of Operations Research* **60:** 193-208.

Mitrovic-Minic, S., Krishnamurti, R. and Laporte, G., 2004, Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B* **38:** 669-685.

Mitrovic-Minic, S. and Laporte, G., 2004, Waiting strategies for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B* **38:** 635-655.

Powell, W. B., Jaillet, P. and Odoni, A.,1995, Stochastic and dynamic networks and routing, in: *Handbooks in operations research and management science 8: Network Routing*, M. O. Ball *et al.,* eds., Amsterdam, Elsevier, pp. 141-295.

Powell, W. B. and Topaloglu, H.,2003, Stochastic Programming in Transportation and Logistics, in: *Handbooks in Operations Research and Management Science: Stochastic Programming*, A. Shapiro and A. Ruszczynski, eds., Amsterdam, Elsevier.

Psaraftis, H. N.,1988, Dynamic vehicle routing problems, in: *Vehicle routing: methods and studies*, B. Golden and A. Assad, eds., Amsterdam, Elsevier Science Publishers, pp. 223-248.

Psaraftis, H. N., 1995, Dynamic vehicle routing: status and prospects, *Annals of Operational Research* **61:** 143-164.

Regan, A. C., Mahmassani, H. S. and P., J., 1998, Evaluation of dynamic fleet management systems: Simulation Framework, *Transportation Research Record*(1645).

Savelsbergh, M. and Sol, M., 1998, DRIVE: Dynamic routing of independent vehicles, *Operations Research* **46:** 474-490.

Thomas, B. W. and White III, C. C., 2004, Anticipatory route selection, *Transportation Science* **38**(4)**:** 473-487.

Toth, P. and Vigo, D., eds., 2002, *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Van Hentenryck, P., Bent, R. and Upfal, E.,2006, *Online Stochastic Optimization Under Time Constraints*, Working Paper.

Yang, J., Jaillet, P. and Mahmassani, H., 2004, Real-time multivehicle truckload pickup and delivery problems, *Transportation Science* **38**(2)**:** 135-148.

Chapter 4

A PARALLELIZABLE AND APPROXIMATE DYNAMIC PROGRAMMING-BASED DYNAMIC FLEET MANAGEMENT MODEL WITH RANDOM TRAVEL TIMES AND MULTIPLE VEHICLE TYPES

Huseyin Topaloglu
School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853, USA

Abstract: This chapter presents an approximate dynamic programming-based dynamic fleet management model that can handle random load arrivals, random travel times and multiple vehicle types. Our model decomposes the fleet management problem into a sequence of time-indexed subproblems by formulating it as a dynamic program and uses approximations of the value function. To handle random travel times, the state variable of our dynamic program includes all individual decisions over a relevant portion of the history. We propose a sampling-based strategy to approximate the value function under this high-dimensional state variable in a tractable manner. Under our value function approximation strategy, the fleet management problem decomposes into a sequence of time-indexed min-cost network flow subproblems that naturally yield integer solutions. Moreover, the subproblem for each time period further decomposes by the locations, making our model suitable for parallel computing. Computational experiments show that our model yields high-quality solutions within reasonable runtimes.

Keywords: dynamic programming; approximate dynamic programming; fleet management.

4.1 INTRODUCTION AND RELEVANT LITERATURE

Although the majority of the dynamic fleet management models assume that the travel times are deterministic, there are a variety of applications where traffic jams, equipment failures and undesirable weather conditions create

substantial variability in the travel times. Furthermore, even if these events are rare, the travel times may appear to be random to the modeler, since they depend on factors outside the scope of the model, such as the skill level of the drivers and the schedules of the ferryboats that are used by the vehicles to cross waterways. This chapter presents an approximate dynamic programming-based model for the dynamic fleet management problem with random load arrivals, random travel times and multiple vehicle types.

The work we present in this chapter is motivated by the empty railcar allocation setting. In the car allocation business, the railroad company receives car requests from its clients on a daily basis. These requests are for a particular number of cars of a particular type, at a particular operating station and on a particular date. The company decides which cars should be used to satisfy the requests and tries to get these cars to the clients. After using the cars for a certain amount of time, the clients return the cars to the company. To serve the clients in a prompt manner and to offset the imbalances between where the requests originate and where the cars are returned, the company continuously repositions the empty cars. Due to limited train capacities and shifting local train schedules, the travel times can be highly variable.

The strategy that we propose in this chapter has ties with the previous research. Godfrey and Powell (2002a) and Godfrey and Powell (2002b) propose approximate dynamic programming-based models for fleet management problems with random load arrivals, deterministic travel times and a single vehicle type. Topaloglu and Powell (2006) extend this work to problems with multiple vehicle types. The idea in these models is to decompose the fleet management problem into time-indexed subproblems by formulating it as a dynamic program and to use approximations of the value function. We employ a similar strategy here, but we use a new dynamic programming formulation to handle random travel times and multiple vehicle types. The difficulty in handling random travel times arises from the fact that when a vehicle is dispatched from a particular origin to a particular destination, it is not known when the vehicle will reach its destination. Consequently, the state variable in our dynamic programming formulation keeps track of *all individual decisions over a relevant portion of the history*. This increases the number of dimensions of the state variable, but we show that one can approximate the value function in a tractable manner under this high-dimensional state variable.

In two recent companion papers (see Topaloglu, 2005 and Topaloglu and Powell, 2006), we address random load arrivals and random travel times in more restricted settings. One of our goals here is to extend these papers and other earlier work in the following four dimensions to build fleet management models that can simultaneously handle random load arrivals, random travel times and multiple vehicle types. 1) We devise a value function approximation strategy

under which the subproblems that need to be solved for each time period reduce to min-cost network flow problems that yield integer solutions naturally. If one naively attempts to generalize the earlier models to handle multiple vehicle types, then the subproblems that need to be solved for each time period reduce to min-cost integer multicommodity network flow problems, in which case obtaining integer solutions may be difficult. 2) We use separable approximations of the value function and there are other fleet management models that use such value function approximations. One can argue that separable approximations work well because the fleet management problem is "inherently separable" by the geographical locations due to the fact that the vehicles located at different locations can serve different sets of loads. However, it is difficult to claim that the fleet management problem is "inherently separable" by the vehicle types when there are multiple types of vehicles that can exist at the same location and compete to serve the same set of demands. In this case, the success of separable approximations is not as obvious. Our computational experiments indicate that separable approximations can work well even in the presence of multiple vehicle types. 3) Our model decomposes the fleet management problem by locations as well as by time periods. In particular, it solves one subproblem for each time period-location pair, and in a certain time period, the subproblems corresponding to different locations can be solved in parallel. When coupled with the fact that these subproblems are min-cost network flow problems, this parallelization opportunity gives our model a significant runtime advantage. Even for problems with deterministic load arrivals or deterministic travel times or a single vehicle type, our model may be preferable due to its runtime advantage. 4) As a byproduct of parallelization, making the decisions for different locations by solving independent subproblems accurately mimics the decision-making process in many applications. Freight carriers usually have multiple dispatchers responsible from managing the vehicles at different locations and each dispatcher pays little attention to the other dispatchers when making its vehicle allocation decisions. Our model allows each dispatcher to concentrate only on the location that it is responsible from and the dispatchers coordinate their decisions through the value function approximations.

Fleet management models have a long history and comprehensive reviews can be found in Dejax and Crainic (1987), Powell (1988), Powell *et al*. (1995) and Crainic and Laporte (1998). We restrict our review to the most relevant literature. Early fleet management models appear as the first applications of linear programming and min-cost network flow algorithms (see Dantzig and Fulkerson, 1954, Ferguson and Dantzig, 1955, White and Bomberault, 1969 and White, 1972). These models formulate the problem over a *state-time network*, where the nodes represent the supply of vehicles at different locations and at different time periods, and the arcs represent the vehicle movements. They assume that the load arrivals over the entire planning

horizon are known in advance or incorporate the uncertain future load arrivals through their expected values. In practice, solving these models often requires integer programming techniques because the network structure is quickly lost when one attempts to address multiple vehicle types or load pick up windows (see Abara, 1989 and Hane *et al*., 1995). Not too far from the state-time network models are the myopic assignment models that solve a simple assignment problem for each time period (see Powell, 1988 and Powell, 1996). These models do not incorporate the uncertain future load arrivals at all and only work with what is known with certainty, which is justified by the fact that the carriers receive the shipment requests far in advance. They are easy to implement, and especially for this reason, they are widely used in practice.

A second class of fleet management models attempt to address the randomness in the load arrivals explicitly. The earliest examples of these models assume that a constant fraction, say β_{ijt}, of the empty vehicles available at location i at time period t is repositioned to location j. In this case, the fleet management problem can be formulated as a nonlinear program to find the best values for these fractions (see Jordan and Turnquist, 1983 and Powell, 1986). Recent models address the randomness in the load arrivals by decomposing the problem into time-indexed subproblems and assessing the impact of the current decisions on the future through value functions. Due to the large number of decision variables and possible load realizations, classical dynamic programming techniques are not feasible for computing the value functions and most of the effort revolves around approximating the value functions in a tractable manner (see Frantzeskakis and Powell, 1990, Crainic and Gendreau, 1993, Carvalho and Powell, 2000, Godfrey and Powell, 2002a, Godfrey and Powell, 2002b and Adelman, 2004). The model we present in this chapter falls in this category.

Myopic assignment models remain applicable when the travel times are random. Other than these straightforward models, we are not aware of a fleet management model that can handle random travel times. Laporte *et al*. (1992) and Kenyon and Morton (2003) consider random travel times in the context of the vehicle routing problem. Their work does not apply to the fleet management problem because they focus on building fixed vehicle routes that yield the best average performance, whereas the fleet management setting requires continuous management of the vehicles. Parallelization and distributed computing have not seen much attention in the area of transportation. These concepts usually appear as a byproduct of an algorithmic strategy such as Lagrangian relaxation or Dantzig-Wolfe decomposition. For example, Chien *et al*. (1989) and Fumero and Vercellis (1999) propose decomposition strategies for inventory distribution problems motivated by Lagrangian relaxation. Bourbeau *et al*. (2000) parallelize branch-and-bound for a large-scale fleet management application.

4.2 PROBLEM DESCRIPTION

We have a heterogeneous fleet of vehicles to serve the loads that occur at different locations in a transportation network over a finite planning horizon. At every time period, a random number of loads enter the system, and we need to decide which loads we should carry and to which locations we should reposition the empty vehicles. We are interested in maximizing the total expected profit over the planning horizon. We define the following.

$\mathcal{T}$ = Set of time periods in the planning horizon, $\mathcal{T} = \{1,...,T\}$ for some finite *T*.

$\mathcal{V}$ = Set of vehicle types.

$\mathcal{I}$ = Set of locations in the transportation network.

$\mathcal{L}$ = Set of movement modes using which a vehicle can move from one location to another, $\mathcal{L} = \{0,...,L\}$ for some finite *L*. Movement mode 0 always corresponds to empty repositioning. The other modes correspond to carrying different types of loads. We elaborate on the concept of movement modes below.

x^v_{ijlt} = Number of vehicles of type *v* dispatched from location *i* to *j* at time period *t* under movement mode *l*.

c^v_{ijlt} = Profit from dispatching one vehicle of type *v* from location *i* to *j* at time period *t* under movement mode *l*.

D_{ijlt} = Random variable for the number of loads that need to be carried from location *i* to *j* at time period *t* and that correspond to movement mode *l*.

τ_{ij} = Random variable for the number of time periods required to move from location *i* to *j*. We assume that $1 \leq \tau_{ij} \leq \tau$ for some finite τ.

Whenever we have $l \in \{1,...,L\}$, the decision variable x^v_{ijlt} captures the number of vehicles of type *v* carrying a load of type *l* from location *i* to *j* at time period *t*, and the profit from each one of these loads is c^v_{ijlt}. If it is not feasible to use a vehicle of type *v* to carry a load of type *l*, then we assume that $c^v_{ijlt} = -\infty$ for all $i,j \in \mathcal{I}$, $t \in \mathcal{T}$. The decision variable x^v_{ij0t} captures the number of vehicles of type *v* moving empty from location *i* to *j* at time

period t, and the cost of each one of these movements is $-c^v_{ij0t}$. Since the empty movements are not bounded, we have $D_{ij0t} = \infty$ for all $i,j \in \mathcal{I}$, $t \in \mathcal{T}$.

One advantage of our notation is that it does not require making a distinction between empty and loaded movements. For example, we can succinctly write the profit at time period t as $\sum_{i,j \in \mathcal{I}} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c^v_{ijlt} \, x^v_{ijlt}$ and the number of vehicles of type v leaving location i at time period t as $\sum_{j \in \mathcal{I}} \sum_{l \in \mathcal{L}} x^v_{ijlt}$. Finally, we note that the decision variable x^v_{ii0t} captures the number of vehicles of type v held at location i at time period t. For notational uniformity, we let $\tau_{ii} = 1$ for all $i \in \mathcal{I}$, although the travel time from a location to itself is, of course, 0 in reality.

By suppressing some of the indices in the variables above, we denote a vector composed of the components ranging over the suppressed indices. For example, we have $x_t = \{x^v_{ijlt} : i, j \in \mathcal{I}, l \in \mathcal{L}, v \in \mathcal{V}\}$ and $D_t = \{D_{ijlt} : i, j \in \mathcal{I}, l \in \mathcal{L}\}$. We reserve the letters s, t and u to index the time periods, and if two or three of them are used in the same context, then the ordering $s \leq t \leq u$ holds.

4.3 MODEL FORMULATION

We begin by reviewing the fleet management model proposed by Topaloglu and Powell (2006). Although it is unable to handle random travel times, this model gives a good starting point.

4.3.1 Deterministic Travel Times

To capture the state of the vehicles, we define the following.

r^v_{iut} = Right before making the decisions at time period t, the number of vehicles of type v that are inbound to location i and that will reach location i at time period u.

Since $\tau_{ij} \leq \tau$ for all $i,j \in \mathcal{I}$, a vehicle dispatched to location i before time period t reaches its destination before time period $t + \tau$. Therefore, we have $r^v_{iut} = 0$ for all $i \in \mathcal{I}$, $v \in \mathcal{V}$, $u = t + \tau, \ldots, T$, and the vector $r_t = \{r^v_{iut} : i \in \mathcal{I}, v \in \mathcal{V}, u = t, \ldots, t + \tau - 1\}$ completely defines the state of the vehicles right before making the decisions at time period t. We note that $r_1 = \{r^v_{iu1} : i \in \mathcal{I}, v \in \mathcal{V}, u = 1, \ldots, \tau\}$ gives the initial position of the vehicles

and is a part of the problem data. Due to the decisions made before the beginning of the planning horizon of the problem, r^v_{iu1} can be greater than 0 for some $i \in I, v \in \mathcal{V}, u > 1$.

Since r^v_{itt} captures the number of vehicles of type v available at location i at time period t, the set of feasible decisions for any state vector r_t and load realizations D_t is given by

$$X(r_t, D_t) = \{x_t : \sum_{j \in I} \sum_{l \in \mathcal{L}} x^v_{ijlt} = r^v_{itt} \quad i \in I, v \in \mathcal{V} \tag{1}$$

$$\sum_{v \in \mathcal{V}} x^v_{ijlt} \le D_{ijlt} \quad i, j \in I, l \in \mathcal{L} \tag{2}$$

$$x^v_{ijlt} \in Z_+ \quad i, j \in I, l \in \mathcal{L}, v \in \mathcal{V}\}, \tag{3}$$

where the left side of (1) accounts for the total number of vehicles of type v leaving location i and the left side of (2) accounts for the total number of vehicles carrying a load of type l from location i to j. Given the decisions x_t and the state vector r_t at time period t, the state of the system at the beginning of the next time period is given by

$$r^v_{ju,t+1} = \sum_{i \in I} \sum_{l \in \mathcal{L}} 1_{\tau_{ij}}(u-t)\, x^v_{ijlt} + r^v_{jut} \quad j \in I, v \in \mathcal{V}, u = t+1, \ldots, t+\tau, \tag{4}$$

where we assume that τ_{ij} is deterministic, and $1_a(b)$ takes value 1 when $a = b$ and takes value 0 otherwise. We bring (1)-(4) together by

$$Y(r_t, D_t) = \{(x_t, r_{t+1}) : x_t \in X(r_t, D_t)$$

$$r^v_{ju,t+1} = \sum_{i \in I} \sum_{l \in L} 1_{\tau_{ij}}(u-t)\, x^v_{ijlt} + r^v_{jut}$$

$$j \in I, v \in \mathcal{V}, u = t+1, \ldots, t+\tau\},$$

whereby $(x_t, r_{t+1}) \in Y(r_t, Dt)$ means that the decisions x_t are feasible when the state of the system is r_t and the realization of the loads is D_t, and applying the decisions x_t on the state vector r_t generates the state vector r_{t+1} for the next time period. Using r_t as the state variable, the problem can be formulated as a dynamic program as

$$V_t(r_t) = E\left\{ \max_{(x_t, r_{t+1}) \in Y(r_t, D_t)} c_t \cdot x_t + V_{t+1}(r_{t+1}) \,|\, r_t \right\}, \tag{5}$$

where $V_t(.)$ is the value function at time period t (see Puterman, 1994). For any system state r_t and load realizations D_t, the optimal decisions for time period t can be found by solving the *subproblem*

$$V_t(r_t, D_t) = \max_{(x_t, r_{t+1}) \in Y(r_t, D_t)} c_t \cdot x_t + V_{t+1}(r_{t+1}). \quad (6)$$

Due to the so-called curse of dimensionality, solving problem (5) in order to compute $\{V_t(.) : t \in \mathcal{T}\}$ is usually intractable. Motivated by the fact that the value function is piecewise-linear concave, Topaloglu and Powell (2006) propose replacing the value function $V_t(.)$ with a separable piecewise-linear concave approximation $\hat{V}_t(.)$ of the form

$$\hat{V}_t(r_t) = \sum_{i \in \mathcal{I}} \sum_{v \in \mathcal{V}} \sum_{u=t}^{t+\tau-1} \hat{V}^v_{iut}(r^v_{iut}), \quad (7)$$

where each $\hat{V}^v_{iut}(.)$ is a single-dimensional piecewise-linear concave function. Consequently, they propose solving the *approximate subproblem*

$$\tilde{V}_t(r_t, D_t) = \max_{(x_t, r_{t+1}) \in Y(r_t, D_t)} c_t \cdot x_t + \hat{V}_{t+1}(r_{t+1}) \quad (8)$$

to make the decisions at time period t, where $\tilde{V}_t(r_t, D_t)$ is simply a place-holder for the optimal objective value. Their approach solves the problem above for different values of r_t and D_t, and iteratively improves the quality of the value function approximations. Our strategy closely parallels this approach, but we use a new state variable and a new method to improve the quality of the value function approximations. Closing this section, we note that, due to constraints (4), solving problem (8) requires prior knowledge of $1_{\tau_{ij}}(u-t)$ for all $i,j \in \mathcal{I}$, $u = t+1,...,t+\tau$. Therefore, this model cannot be used when the travel times are random.

4.3.2 Random Travel Times

We deal with the random travel times by keeping track of all individual decisions over a relevant portion of the history. To formalize, we define the following.

f^v_{ijst} = Right before observing the vehicle arrivals and making the decisions at time period t, the number of vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before time period t.

Since $\tau_{ij} \leq \tau$ for all $i,j \in I$, a vehicle dispatched from location i to j before time period $t-\tau$ will reach its destination before time period t. Therefore, we have $f^v_{ijst} = 0$ for all $i,j \in I$, $v \in \mathcal{V}$, $s = 1,...,t-\tau-1$, and the vector $f_t = \{f^v_{ijst} : i,j \in I, v \in \mathcal{V}, s = t-\tau,...,t-1\}$ gives all decisions over the portion of the history relevant to the decisions made at time period t. Thus, we use f_t as the state variable in our dynamic programming formulation. We note that $f_1 = \{f^v_{ijs1} : i,j \in I, v \in \mathcal{V}, s = 1-\tau,...,0\}$ gives the decisions made before the beginning of the planning horizon of the problem and is a part of the problem data.

Although the vector f_t captures the history relevant to the decisions made at time period t, the number of vehicles available at each location at time period t depends on the realizations of the travel times and is still a random variable. We define the following random variable.

A^v_{ijst} = Random variable representing the number of vehicles of type v that were dispatched from location i to j at time period s and that reach location j at time period t.

We note that f^v_{ijst} captures the number of vehicles of type v that were dispatched from location i to j at time period s and that have not reached location j before time period t, whereas A^v_{ijst} captures what "portion" of these vehicles actually reach location j at time period t. Therefore, we always have $A^v_{ijst} \leq f^v_{ijst}$. Section 4.5 gives a careful characterization of possible probability laws that may govern the random vector $A_t = \{A^v_{ijst} : i,j \in I, v \in \mathcal{V}, s = t-\tau,...,t-1\}$. For now, we view A_t as a random vector just like D_t whose value becomes known at the beginning of time period t.

Since $\sum_{j \in I} \sum_{s=t-\tau}^{t-1} A^v_{jist}$ is the number of vehicles of type v available at location i at time period t, the set of feasible decisions for any state vector f_t, arrival realizations A_t and load realizations D_t is given by

$$X(r_t, A_t, D_t) = \{x_t : \sum_{j \in I} \sum_{l \in \mathcal{L}} x^v_{ijlt} = \sum_{j \in I} \sum_{s=t-\tau}^{t-1} A^v_{jist} \quad i \in I, v \in \mathcal{V} \qquad (9)$$

$$(2), (3) \qquad \}.$$

Given the decisions x_t and the state vector f_t at time period t, the state of the system at the beginning of the next time period is given by

$$f^v_{ijt,t+1} = \sum_{l \in \mathcal{L}} x^v_{ijlt} \qquad i,j \in I, v \in \mathcal{V} \tag{10}$$

$$f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst} \qquad i,j \in I, v \in \mathcal{V}, s = t+1-\tau,...,t-1. \tag{11}$$

In alignment with the definition of f^v_{ijst}, the right side of (10) computes the number of vehicles of type *v* dispatched from location *i* to *j* at time period *t*, whereas the right side of (11) computes what "portion" of the vehicles of type *v* that were dispatched from location *i* to *j* at time period *s* still remain in-transit after observing the arrivals at time period *t*. We bring (9)-(11) together by

$$Y(f_t, A_t, D_t) = \{(x_t, f_t) : x_t \in X(f_t, A_t, D_t)$$
$$f^v_{ijt,t+1} = \sum_{l \in \mathcal{L}} x^v_{ijlt} \qquad i,j \in I, v \in \mathcal{V}$$
$$f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst}$$
$$i,j \in I, v \in \mathcal{V}, s = t+1-\tau,...,t-1\}.$$

Using f_t as the state variable, the problem can be formulated as a dynamic program as

$$V_t(f_t) = E\left\{ \max_{(x_t, f_{t+1}) \in Y(f_t, A_t, D_t)} c_t \cdot x_t + V_{t+1}(f_{t+1}) \,|\, f_t \right\}. \tag{12}$$

For any system state f_t, arrival realizations A_t and load realizations D_t, the optimal decisions for time period *t* can be found by solving the subproblem

$$V_t(f_t, A_t, D_t) = \max_{(x_t, f_{t+1}) \in Y(f_t, A_t, D_t)} c_t \cdot x_t + V_{t+1}(f_{t+1}). \tag{13}$$

We propose using separable value function approximations of the form

$$\hat{V}_t(f_t) = \sum_{i,j \in I} \sum_{v \in \mathcal{V}} \sum_{s=t-\tau}^{t-1} \hat{V}^v_{ijst}(f^v_{ijst}), \tag{14}$$

where each *value function approximation component* $\hat{V}^v_{ijst}(.)$ is a single-dimensional piecewise-linear concave function with points of nondifferentiability being a subset of integers. The approximation in (14) seems more complicated than the one in (7), but the next section shows that the approximate subproblem can be simplified to a great extent due to the separability of the approximation.

4.4 STRUCTURE OF THE APPROXIMATE SUBPROBLEMS AND PARALLELIZATION

Replacing the value function $V_{t+1}(.)$ in (13) by an approximation of form (14), the approximate subproblem for time period t can be written as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \sum_{i,j \in I} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j \in I} \sum_{v \in \mathcal{V}} \sum_{s=t+1-\tau}^{t} \hat{V}^v_{ijs,t+1}(f^v_{ijs,t+1}) \quad (15)$$

subject to (2), (3), (9), (10)

$$f^v_{ijs,t+1} = f^v_{ijst} - A^v_{ijst} \quad i,j \in I, v \in \mathcal{V}, s = t+1-\tau,...,t-1.$$

For the model with deterministic travel times in Section 4.3.1, Topaloglu and Powell (2006) show that the approximate subproblem (8) is a min-cost integer multicommodity network flow problem. Since the approximate subproblem (8) "spans" only one time period, it is a small min-cost integer multicommodity network flow problem, but the multicommodity characteristics still make it difficulty to obtain integer solutions and bring an unwelcome dimension of complexity. In this section, we show that the approximate subproblem (15) is a min-cost network flow problem.

We note that the last set of constraints in problem (15) set the decision variables $\{f^v_{ijs,t+1} : i,j \in I, v \in \mathcal{V}, s = t+1-\tau,...,t-1\}$ to constants. Therefore, by plugging their values in the objective function, we can drop these decision variables and the value function approximation components corresponding to them. This reduces problem (15) to

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}} \sum_{i,j \in I} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j \in I} \sum_{v \in \mathcal{V}} \hat{V}^v_{ijt,t+1}(f^v_{ijt,t+1}) \quad (16)$$

subject to (2), (3), (9), (10).

Section 4.6 uses the next remark for updating and improving the value function approximations.

Remark 1. Noting constraints (10), the decision variable $f^v_{ijt,t+1}$ captures the number of vehicles of type v dispatched from location i to j at time period t. Therefore, $\hat{V}^v_{ijt,t+1}(f)$ can be interpreted as the approximation to the expected future benefit from dispatching f vehicles from location i to j at time period t.

Letting R be the total number of available vehicles, the relevant domain of $\hat{V}^v_{ijt,t+1}(.)$ is $\{0,1,\ldots,R\}$ and we can represent $\hat{V}^v_{ijt,t+1}(.)$ by a sequence of

numbers $\{\eta^v_{ijt,t+1}(r) : r = 1,...,R\}$, where $\eta^v_{ijt,t+1}(r)$ is the slope of $\hat{V}^v_{ijt,t+1}(.)$ over $(r-1,r)$. That is, we have $\eta^v_{ijt,t+1}(r) = \hat{V}^v_{ijt,t+1}(r) - \hat{V}^v_{ijt,t+1}(r-1)$. In this case, we can write problem (16) explicitly as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, f_{t+1}, z_{t+1}} \sum_{i,j\in I}\sum_{l\in\mathcal{L}}\sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in I}\sum_{v\in\mathcal{V}}\sum_{r=1}^{R} \eta^v_{ijt,t+1}(r)\, z^v_{ijt,t+1}(r) \tag{17}$$

subject to

$$\sum_{j\in I}\sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{j\in I}\sum_{s=t-\tau}^{t-1} A^v_{jist} \qquad i\in I, v\in\mathcal{V} \tag{18}$$

$$\sum_{l\in\mathcal{L}} x^v_{ijlt} - f^v_{ijt,t+1} = 0 \qquad i,j\in I, v\in\mathcal{V} \tag{19}$$

$$f^v_{ijt,t+1} - \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = 0 \quad i,j\in I, v\in\mathcal{V} \tag{20}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} \le D_{ijlt} \qquad i,j\in I, l\in\mathcal{L} \tag{21}$$

$$z^v_{ijt,t+1}(r) \le 1 \qquad i,j\in I, v\in\mathcal{V}, r = 1,...,R \tag{22}$$

$$x^v_{ijlt}, z^v_{ijt,t+1}(r) \in Z_+ \qquad i,j\in I, l\in\mathcal{L}, v\in\mathcal{V}, r = 1,...,R. \tag{23}$$

Defining three sets of nodes $\mathcal{N}_1 = \{(i,v) : i\in I, v\in\mathcal{V}\}$, $\mathcal{N}_2 = \{(i,j,v) : i,j\in I, v\in\mathcal{V}\}$ and $\mathcal{N}_3 = \{(i,j,v) : i,j\in I, v\in\mathcal{V}\}$, the problem above can be visualized as a min-cost integer multicommodity network flow problem that takes place over a network with the set of nodes $\mathcal{N}_1 \cup \mathcal{N}_2 \cup \mathcal{N}_3 \cup \{\Phi\}$, where Φ denotes a dummy root note. In this network, there exists an arc corresponding to each decision variable in problem (17), and Table 4-1 shows the "tail" and "head" nodes for each one of these arcs. Pictorially, problem (17) is the min-cost integer multicommodity network flow problem shown in Figure 4-1, where we assume that $I = \{i_1, i_2\}$, $\mathcal{L} = \{l_1\}$, $\mathcal{V} = \{v_1, v_2\}$ and we label the nodes by $(i,v)\in I\times\mathcal{V}$, $(i,j,v)\in I^2\times\mathcal{V}$. Constraints (18), (19) and (20) are respectively the flow balance constraints for the nodes in $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$. The flow balance constraint for node Φ is redundant and is not included in problem (17). Constraints (21) put a limit on the total flow over a set of arcs and seemingly give problem (17) multicommodity characteristics. However, as the next

Table 4-1. Incidence relationships of the arcs corresponding to the decision variables in problem (17).

Arc	"Tail" node	"Head" node
x_{ijlt}^{v}	$(i,v) \in \mathcal{N}_1$	$(i,j,v) \in \mathcal{N}_2$
$f_{ijt,t+1}^{v}$	$(i,j,v) \in \mathcal{N}_2$	$(i,j,v) \in \mathcal{N}_3$
$z_{ijt,t+1}^{v}(r)$	$(i,j,v) \in \mathcal{N}_3$	Φ

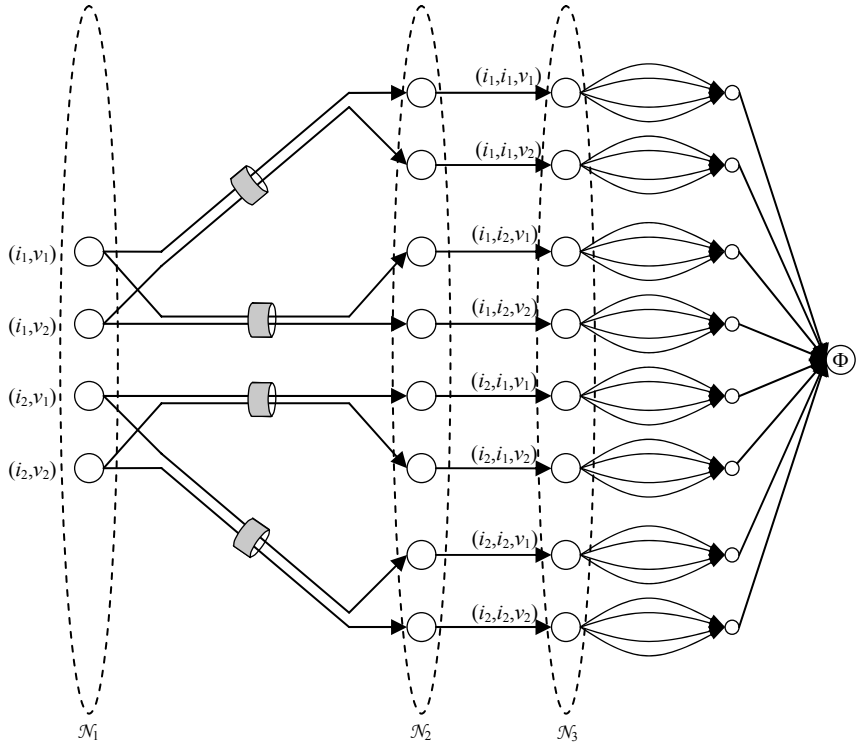


Figure 4-1. Problem (17) as a min-cost integer multicommodity network flow problem.

proposition shows, a simple transformation converts problem (17) into a min-cost network flow problem.

Proposition 1. Problem (17) can be solved as a min-cost network flow problem.

Proof. We combine constraints (19) and (20) into $\sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{r=1}^{R} z^v_{ijt,t+1}(r)$ to drop the decision variables $f^v_{ijt,t+1}$ for all $i,j\in I$, $v\in\mathcal{V}$. Adding the combined constraints for all $j\in I$, we can write constraints (18) as $\sum_{j\in I}\sum_{r=1}^{R} z^v_{ijt,t+1}(r) = \sum_{j\in I}\sum_{s=t-\tau}^{t-1} A^v_{jist}$ for all $i\in I$, $v\in\mathcal{V}$. We define new decision variables $\{y_{ijlt} : i,j\in I,\ l\in\mathcal{L}\}$ and split constraints (21) into $\sum_{v\in\mathcal{V}} x^v_{ijlt} - y_{ijlt} = 0$ and $y_{ijlt} \le D_{ijlt}$ for all $i,j\in I$, $l\in\mathcal{L}$. Therefore, problem (17) can be written as

$$\widetilde{V}_t(f_t, A_t, D_t) = \max_{x_t, z_{t+1}, y_t} \sum_{i,j\in I}\sum_{l\in\mathcal{L}}\sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in I}\sum_{v\in\mathcal{V}}\sum_{r=1}^{R} \eta^v_{ijt,t+1}(r)\, z^v_{ijt,t+1}(r) \tag{24}$$

$$\text{subject to} \quad \sum_{j\in I}\sum_{r=1}^{R} z^v_{ijt,t+1}(r) = \sum_{j\in I}\sum_{s=t-\tau}^{t-1} A^v_{jist} \qquad i\in I, v\in\mathcal{V} \tag{25}$$

$$\sum_{l\in\mathcal{L}} x^v_{ijlt} - \sum_{r=1}^{R} z^v_{ijt,t+1}(r) = 0 \qquad i,j\in I, v\in\mathcal{V} \tag{26}$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} - y_{ijlt} = 0 \qquad i,j\in I, l\in\mathcal{L} \tag{27}$$

$$y_{ijlt} \le D_{ijlt} \qquad i,j\in I, l\in\mathcal{L}$$

(22), (23).

Defining three sets of nodes $O_1 = \{(i,v) : i\in I, v\in\mathcal{V}\}$, $O_2 = \{(i,j,v) : i,j\in I, v\in\mathcal{V}\}$ and $O_3 = \{(i,j,l) : i,j\in I, l\in\mathcal{L}\}$, problem (24) is a min-cost network flow problem that takes place over a network with the set of nodes $O_1 \cup O_2 \cup O_3 \cup \{\Phi\}$. Table 4-2 shows the "tail" and "head" nodes for each one of the arcs corresponding to the decision variables in problem (24). Constraints (25), (26) and (27) are respectively the flow balance constraints for the nodes in O_1, O_2 and O_3. Figure 4-2 shows the general structure of problem (24), where we label the nodes by $(i,v)\in I\times\mathcal{V}$, $(i,j,v)\in I^2\times\mathcal{V}$ and $(i,j,l)\in I^2\times\mathcal{L}$.

The next remark will be useful for updating and improving the value function approximations.

Table 4-2. Incidence relationships of the arcs corresponding to the decision variables in problem (24).

Arc	"Tail" node	"Head" node
$z^{v}_{ijt,t+1}(r)$	$(i,v) \in O_1$	$(i,j,v) \in O_2$
x^{v}_{ijlt}	$(i,j,v) \in O_2$	$(i,j,l) \in O_3$
y_{ijlt}	$(i,j,l) \in O_3$	Φ

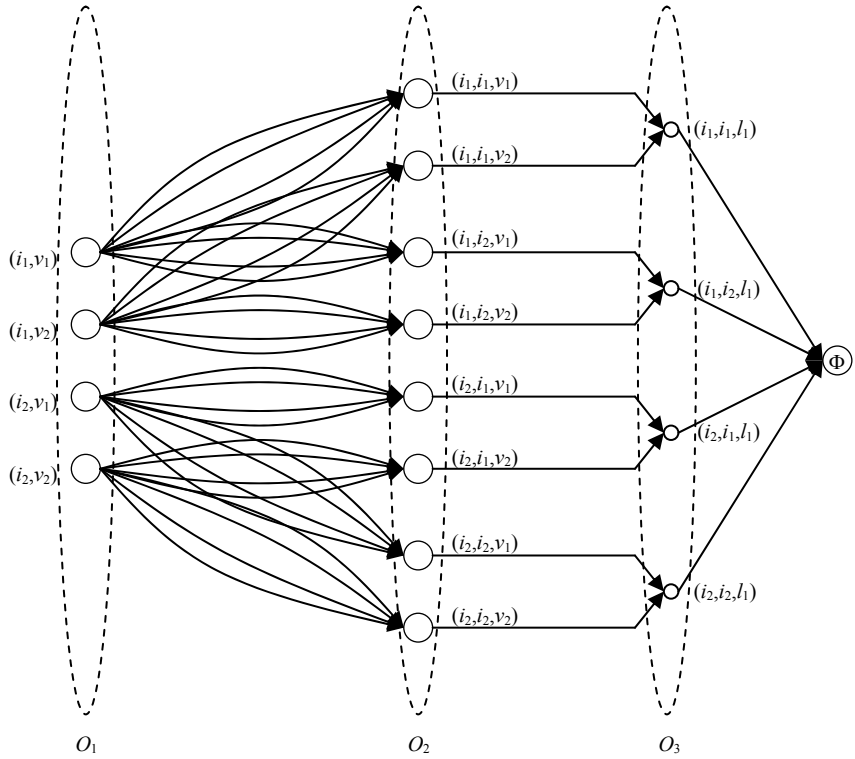


Figure 4-2. Problem (24) as a min-cost network flow problem.

Remark 2. Since problem (17) can be solved as a min-cost network flow problem, we can speak of the dual solution to problem (17).

The fact that problem (17) naturally yields integer solutions gives our model a dramatic runtime advantage. Furthermore, the objective function and the constraints of problem (17) decompose by $i \in I$, which implies that one can obtain a solution to problem (17) by solving $|I|$ smaller subproblems and these $|I|$ subproblems can be solved in parallel. This parallelization opportunity further boosts the runtime advantage of our model.

4.5 CHARACTERIZING THE ARRIVAL RANDOM VARIABLES

In this section, we describe two possible models for the arrival random variables. The first model assumes that the travel times for different vehicles are independent, whereas the second model assumes that the travel times for different vehicles traveling between the same origin-destination pair are perfectly dependent.

Independent Model. This model assumes that the travel times for different vehicles are independent. Given that a particular vehicle was dispatched from location *i* to *j* at time period *s* and it has not reached location *j* before time period *t*, the probability that this vehicle reaches location *j* at time period *t* is $P\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}$. The number of vehicles of type *v* that were dispatched from location *i* to *j* at time period *s* and that have not reached location *j* before time period *t* is given by f_{ijst}^{v}. A "portion" of these vehicles, which is captured by A_{ijst}^{v}, reach location *j* at time period *t*. Therefore, under the independent model, the number of vehicles of type *v* that were dispatched from location *i* to *j* at time period *s* and that reach location *j* at time period *t* is binomially distributed with parameters f_{ijst}^{v} and $P\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}$. In other words, we have

$$P\{A_{ijst}^{v} = a \mid f_{ijst}^{v} = f\} = \binom{f}{a} [P\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}]^{a} \times [P\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\}]^{f-a}$$

for all $i,j \in I$, $v \in \mathcal{V}$, $s = t - \tau,...,t-1$ and the random variables $\{A_{ijst}^{v} : i, j \in I, v \in \mathcal{V}, s = t - \tau,...,t-1\}$ are independent. This model is applicable when the travel times depend on conditions internal to the vehicles or the drivers, such as breakdowns and skill levels.

Perfectly Dependent Model. This model assumes that all vehicles dispatched from location *i* to *j* at time period *s* reach location *j* at the same time period. There are f_{ijst}^{v} vehicles of type *v* that were dispatched from location *i* to *j* at time period *s* and that have not reached location *j* before time period *t*. Under this model, either all of these vehicles reach location *j* at time period *t* (this happens with probability $P\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\}$) or none of these vehicles reach location *j* at time period *t* (this happens with probability $P\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\}$). Therefore, we have

$$P\{A_{ijst} = \alpha \mid f_{ijst} = \phi\} = \begin{cases} 1 & \text{if } \alpha = 0, \phi = 0 \\ P\{\tau_{ij} = t - s \mid \tau_{ij} \geq t - s\} & \text{if } \alpha = \phi, \phi \neq 0 \\ P\{\tau_{ij} > t - s \mid \tau_{ij} \geq t - s\} & \text{if } \alpha = 0, \phi \neq 0 \\ 0 & \text{otherwise,} \end{cases}$$

where $A_{ijst} = \{A^v_{ijst} : v \in \mathcal{V}\}$, $f_{ijst} = \{f^v_{ijst} : v \in \mathcal{V}\}$, α and ϕ are $|\mathcal{V}|$-dimensional vectors. We still assume that the random vectors $\{A_{ijst} : i, j \in I, s = t - \tau, ..., t - 1\}$ are independent. This model is applicable when the travel times depend on external conditions, such as weather and traffic.

The next remark will be useful for updating and improving the value function approximations.

Remark 3. Under both models, a vehicle dispatched from location *i* to *j* at time period *s* reaches its destination at time period *t* with probability $P\{\tau_{ij} = t - s\}$. This property holds for the first model because its probability law is equivalent to using the distribution of τ_{ij} to sample an independent travel time for each vehicle dispatched from location *i* to *j*. The second model also satisfies this property because its probability law is equivalent to using the distribution of τ_{ij} to sample one travel time for all vehicles dispatched from location *i* to *j* at a particular time period.

Clearly, both characterizations of the arrival random variables presented in this section have limitations and the travel times may have more complex dependencies in reality. For example, we assume that the travel times for the vehicles that travel between different origin-destination pairs or that start their trip at different time periods are independent. However, one would expect the travel times for the vehicles leaving or arriving at the same or nearby locations at the same or nearby time periods to be strongly correlated, especially if the travel times are affected by external conditions, such as weather or traffic. Nevertheless, the two characterizations we give serve as a good starting point. Furthermore, Remark 3 is the only property of these characterizations that we use in the rest of the paper. Therefore, our model should work with other possible characterizations of the arrival random variables that satisfy Remark 3.

4.6 UPDATING AND IMPROVING THE VALUE FUNCTION APPROXIMATIONS

The performance of the decisions made by solving approximate subproblems of form (15) depends on how "well" the value function approximations approximate the exact value function. We propose a sampling-based strategy that constructs the value function approximations in an iterative manner. In

particular, we let $\{\hat{V}_t^n(.): t \in \mathcal{T}\}$ be the set of value function approximations at iteration n. For each time period t in the planning horizon, we sample a realization of A_t and D_t, which we respectively denote by $\widetilde{A}_t^n$ and $\widetilde{D}_t^n$, and solve the approximate subproblem

$$(x_t^n, f_{t+1}^n) = \underset{(x_t, f_{t+1}) \in (f_t^n, \widetilde{A}_t^n, \widetilde{D}_t^n)}{\arg\max} \sum_{i,j \in I} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c_{ijlt}^v \, x_{ijlt}^v + \sum_{i,j \in I} \sum_{v \in \mathcal{V}} \sum_{s=t+1-\tau}^{t} \hat{V}_{ijs,t+1}^{vn}(f_{ijs,t+1}^v) \quad (28)$$

where f_1^n is initialized to reflect the initial state of the system. We note that solving approximate subproblems of form (28) for all $t \in \mathcal{T}$ is equivalent to simulating the behavior of the policy characterized by the value function approximations $\{\hat{V}_t^n(.): t \in \mathcal{T}\}$ under arrival realizations $\{\widetilde{A}_t^n : t \in \mathcal{T}\}$ and load realizations $\{\widetilde{D}_t^n : t \in \mathcal{T}\}$. The idea is to use the primal-dual solutions to the approximate subproblems to update and improve the value function approximations. The heuristic method that we use for this purpose is based on the following observations.

Remark 4. Comparing the approximate subproblems in (15) and (16) shows that we only need the value function approximation components $\{\hat{V}_{ijt,t+1}^v(.): i, j \in I, v \in \mathcal{V}, t \in \mathcal{T}\}$. The value function approximation components $\{\hat{V}_{ijs,t+1}^v(.): i, j \in I, v \in \mathcal{V}, s = t+1-\tau, ..., t-1, t \in \mathcal{T}\}$ do not affect our decisions at all and we do not need to improve them.

Remark 5. At iteration n, we dispatch $f_{ijt,t+1}^{vn}$ vehicles of type v from location i to j at time period t (see (10) and (28)). Recalling the interpretation of $\hat{V}_{ijt,t+1}^v(.)$ in Remark 1, this implies that the quantity $\eta_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn}+1) = \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn}+1) - \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}^{vn})$ approximates the expected future benefit from dispatching an additional vehicle of type v from location i to j at time period t.

Remark 6. Noting problem (16), the approximate subproblem solved at time period t is

$$\widetilde{V}_t(f_t, \widetilde{A}_t^n, \widetilde{D}_t^n) = \max_{x_t, f_{t+1}} \sum_{i,j \in I} \sum_{l \in \mathcal{L}} \sum_{v \in \mathcal{V}} c_{ijlt}^v \, x_{ijlt}^v + \sum_{i,j \in I} \sum_{v \in \mathcal{V}} \hat{V}_{ijt,t+1}^{vn}(f_{ijt,t+1}^v)$$

$$\text{subject to} \quad \sum_{j\in I}\sum_{l\in\mathcal{L}} x_{ijlt}^{v} = \sum_{j\in I}\sum_{s=t-\tau}^{t-1} \widetilde{A}_{jist}^{vn} \quad i\in I, v\in\mathcal{V} \tag{29}$$

(2), (3), (10).

The expression on the right side of constraints (29) is the number of vehicles of different types available at different locations at time period *t*. Since we can speak of the dual solution to the problem above due to Remark 2, we let $\{\theta_{it}^{vn} : i\in I, v\in\mathcal{V}\}$ be the optimal values of the dual variables associated with these constraints at iteration *n*. Therefore, θ_{it}^{vn} gives an estimate of the expected benefit from having an additional vehicle of type *v* at location *i* at time period *t*.

Remark 7. Noting Remark 3, a vehicle dispatched from location *i* to *j* at time period *t* reaches location *j* at time period *u* with probability $P\{\tau_{ij} = u-t\}$. In view of Remark 6, this implies that, at iteration *n*, we can use $\vartheta_{ijt}^{vn} = \sum_{u=t+1}^{t+\tau} P\{\tau_{ij} = u-t\}\,\theta_{ju}^{vn}$ to estimate the expected future benefit from dispatching an additional vehicle of type *v* from location *i* to *j* at time period *t*.

Step 1. For all $r = 1,\ldots,R$, let

$$q_{ijt,t+1}^{vn}(r) = \begin{cases} (1-\alpha^n)\,\eta_{ijt,t+1}^{vn}(r) + \alpha^n\,\vartheta_{ijt}^{vn} & \text{if } r = f_{ijt,t+1}^{vn}+1 \\ \eta_{ijt,t+1}^{vn}(r) & \text{if } r\in\{1,\ldots,f_{ijt,t+1}^{vn}, f_{ijt,t+1}^{vn}+2,\ldots,R\}, \end{cases}$$

where $\alpha^n \in [0,1]$ is the smoothing constant at iteration *n*.

Step 2. Let the vector $\eta_{ijt,t+1}^{v,n+1} = \{\eta_{ijt,t+1}^{v,n+1}(r) : r = 1,\ldots,R\}$, which characterizes the value function approximation component $\hat{V}_{ijt,t+1}^{v,n+1}(.)$ at the next iteration *n* + 1, be

$$\eta_{ijt,t+1}^{v,n+1} = \arg\min_{z} \quad \sum_{r=1}^{R} [z(r) - q_{ijt,t+1}^{vn}(r)]^2$$

$$\text{subject to} \quad z(r) - z(r-1) \le 0 \qquad r = 2,\ldots,R. \tag{34}$$

Figure 4-3. The method to update the value function approximation component $\hat{V}_{ijt,t+1}^{vn}(.)$.

Step 1. Initialize $n = 1$. Initialize $\{\hat{V}^{vn}_{ijt,t+1}(.) : i, j \in I, v \in \mathcal{V}, t \in \mathcal{T}\}$ to piecewise-linear concave functions with points of nondifferentiability being a subset of integers (possibly 0).

Step 2. Initialize $t = 1$. Initialize f^n_1 to reflect the initial state of the system.

Step 3. Given f^n_t, let $\widetilde{A}^n_t$ and $\widetilde{D}^n_t$ respectively be samples of A_t and D_t.

Step 4. Solve the approximate subproblem (15). Let

$$(x^n_t, f^n_{t+1}) = \max_{x_t, f_{t+1}} \sum_{i,j\in I}\sum_{l\in\mathcal{L}}\sum_{v\in\mathcal{V}} c^v_{ijlt}\, x^v_{ijlt} + \sum_{i,j\in I}\sum_{v\in\mathcal{V}}\sum_{s=t+1-\tau}^{t} \hat{V}^{vn}_{ijs,t+1}(f^v_{ijs,t+1}) \qquad (17)$$

$$\text{subject to} \quad \sum_{j\in I}\sum_{l\in\mathcal{L}} x^v_{ijlt} = \sum_{j\in I}\sum_{s=t-\tau}^{t-1} \widetilde{A}^{vn}_{jist} \qquad i \in I, v \in \mathcal{V} \qquad (35)$$

$$\sum_{v\in\mathcal{V}} x^v_{ijlt} \le \widetilde{D}^n_{ijlt} \qquad i, j \in I, l \in \mathcal{L}$$

$$f^v_{ijs,t+1} = f^{vn}_{ijst} - \widetilde{A}^{vn}_{ijst} \qquad i, j \in I, v \in \mathcal{V}, s = t+1-\tau, ..., t-1$$

(3), (10).

Let $\{\theta^{vn}_{it} : i \in I, v \in \mathcal{V}\}$ be the optimal values of the dual variables associated with constraints (35).

Step 5. Increase t by 1. If $t \le T$, then go to Step 3.

Step 6. For all $i, j \in I, v \in \mathcal{V}, t \in \mathcal{T}$, let $\vartheta^{vn}_{ijt} = \sum_{u=t+1}^{t+\tau} P\{\tau_{ij} = u - t\}\, \theta^{vn}_{ju}$.

Step 7. For all $i, j \in I, v \in \mathcal{V}, t \in \mathcal{T}$, use $\eta^{vn}_{ijt,t+1} = \{\eta^{vn}_{ijt,t+1}(r) : r = 1, ..., R\}$ and ϑ^{vn}_{ijt} in the smoothing method in Figure 4-3 to obtain the value function approximation component $\hat{V}^{v,n+1}_{ijt,t+1}(.)$ that will be used at the next iteration.

Step 8. Increase n by 1. If one more iteration is needed, then go to Step 2.

Figure 4-4. The complete solution method.

We now put Remarks 4-7 together. At iteration n, we dispatch $f^{vn}_{ijt,t+1}$ vehicles of type v from location i to j at time period t and $\eta^{vn}_{ijt,t+1}(f^{vn}_{ijt,t+1} + 1)$ approximates the expected future benefit from an additional vehicle of type v dispatched from location i to j at time period t (Remark 5). Through the solution of the approximate subproblems at iteration n, we estimate the same quantity by ϑ^{vn}_{ijt} (Remarks 6 and 7). We use this new information to update

and improve the value function approximation component $\hat{V}_{ijt,t+1}^{vn}(.)$ through the smoothing method in Figure 4-3. Step 1 in this figure smoothes the slope of $\hat{V}_{ijt,t+1}^{vn}(.)$ at the relevant point by using the new information. After smoothing, the function $Q_{ijt,t+1}^{vn}(.)$ characterized by the sequence of slopes $\{q_{ijt,t+1}^{vn}(r): r = 1,...,R\}$ is not necessarily concave. Therefore, Step 2 projects the function $Q_{ijt,t+1}^{vn}(.)$ onto the set of single-dimensional piecewise-linear concave functions with points of nondifferentiability being a subset of integers. Constraints (34) ensure that the value function approximation component $\hat{V}_{ijt,t+1}^{v,n+1}(.)$ at the next iteration is concave. This updating method is due to Powell *et al*. (2004). Figure 4-4 describes our complete solution methodology.

4.7 COMPUTATIONAL EXPERIMENTS

In this section, we test the quality of the solutions obtained by our model and investigate how the runtimes scale with different problem parameters.

4.7.1 Experimental Setup

We present results on three problem classes. The first problem class includes problems with deterministic load arrivals and deterministic travel times. These problems can be formulated as min-cost integer multicommodity network flow problems. The second problem class includes problems with random load arrivals and deterministic travel times. The model in Section 4.3.1 can be used as a benchmark for these problems. The third problem class includes problems with random load arrivals and random travel times. We use the so-called rolling horizon strategy and an extension of the model in Section 4.3.1 as benchmarks for these problems. We include problems with deterministic travel times in our experimental setup because there exist a variety of solution methods for these problems. This enables us to carefully test the performance of our model. All of the algorithms were coded in JAVA 1.4.1 and executed on a Pentium IV PC with 2.4 GHz CPU and 1 GB RAM running Windows XP.

In our experimental setup, we generate one basic problem and modify its certain attributes to generate different test problems. All of our test problems involve 5 vehicle types and 6 movement modes, one of which corresponds to empty repositioning and the other 5 correspond to serving different types of loads. In practice, the number of load types can be on the order of hundreds, but our model scales well with the number of load types since the number of

dimensions of the state variable does not depend on the number of load types. We let $c^v_{ij0t} = -c_0\ \delta(i,j)$ for all $i,j \in I$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, where c_0 is the "per-mile" empty repositioning cost and $\delta(i,j)$ is the distance between locations i and j. In practice, empty repositioning costs are often applied on a "per-mile" basis. Letting C be a 5×5-dimensional matrix, we let $c^v_{ijlt} = r\ \delta(i,j)\ C^v_l - c_0\ \delta(i,j)$ for all $i,j \in I$, $l \in \{1,\ldots,5\}$, $v \in \mathcal{V}$, $t \in \mathcal{T}$, where r is the revenue applied on a "per-mile" basis and $C^v_l \in [0,1]$ is the (v,l)-th entry of the matrix C. Consequently, there are more and less suitable vehicle types for each load type. Table 4-3 gives the different values we use for the matrix C. In practice, C^v_l may capture the willingness of the dispatcher to use a vehicle of type v to cover a load of type l, rather than a monetary discounting factor. The number of loads for each origin-destination pair, movement mode and time period is sampled from the Poisson distribution with the appropriate mean. The Poisson assumption is reasonable in many practical settings. We use the method described in Godfrey and Powell (2002a) to generate test problems where the number of loads inbound to a particular location is negatively correlated with the number of loads outbound from that location. We expect these problems to require plenty of empty repositioning movements in the optimal solution and naive solution methods should not give satisfactory results for them. We use $(T, |I|, f, D, c_0, C) \in \{30, 60, 90\}\times\{10, 20, 40\}\times\{100, 200, 400\}\times\{2000, 4000, 6000\}\times\{1.6, 4, 8\}\times\{C_1, C_2, C_3, C_4\}$ to denote the attributes of our test problems, where the six dimensions respectively describe the number of time periods in the planning horizon, the number of locations in the transportation network, the size of the fleet, the expected number of loads over the planning horizon, the "per-mile" empty repositioning cost and the matrix characterizing the compatibility between the vehicle and load types. Our basic test problem, which is the first test problem reported in Tables 4-4, 4-5 and 4-6, involves 60 time periods, 20 locations, 200 vehicles, 4000 loads, empty repositioning cost of 4 and matrix C_1 from Table 4-3.

Table 4-3. Matrices characterizing the compatibility between vehicle and load types.

C_1					C_2					C_3					C_4				
1	.8	.5	.3	0	1	0	0	0	0	1	.5	0	0	0	1	1	1	1	1
.7	1	.8	.3	0	1	1	0	0	0	.5	1	.5	0	0	1	1	1	1	1
.6	.6	1	.5	.5	1	1	1	0	0	0	.5	1	.5	0	1	1	1	1	1
0	.4	.7	1	.5	1	1	1	1	0	0	0	.5	1	.5	1	1	1	1	1
0	.4	.6	.6	1	1	1	1	1	1	0	0	0	.5	1	1	1	1	1	1

Table 4-4. Results for problems with deterministic load arrivals and deterministic travel times.

Problem	Performance ratio	CPU (sec.) for our model
(60, 20, 200, 4000, 4, C_1)	99.04	3.7
(30, 20, 200, 2000, 4, C_1)	98.63	1.9
(90, 20, 200, 6000, 4, C_1)	98.76	6.0
(60, 10, 200, 4000, 4, C_1)	99.53	1.2
(60, 40, 200, 4000, 4, C_1)	98.97	14.3
(60, 20, 200, 4000, 4, C_2)	98.94	3.8
(60, 20, 200, 4000, 4, C_3)	98.83	3.6
(60, 20, 200, 4000, 4, C_4)	99.24	3.8
(60, 20, 100, 4000, 4, C_1)	97.09	3.1
(60, 20, 400, 4000, 4, C_1)	98.66	3.9
(60, 20, 200, 4000, 1.6, C_1)	98.71	3.8
(60, 20, 200, 4000, 8, C_1)	99.12	3.8

Table 4-5. Results for problems with random load arrivals and deterministic travel times.

Problem	Performance ratio	CPU (sec.) for our model	CPU (sec.) for model in Section 4.3.1
(60, 20, 200, 4000, 4, C_1)	96.09	3.7	3.4
(30, 20, 200, 2000, 4, C_1)	95.13	1.9	1.8
(90, 20, 200, 6000, 4, C_1)	92.40	6.0	5.7
(60, 10, 200, 4000, 4, C_1)	97.01	1.2	0.9
(60, 40, 200, 4000, 4, C_1)	95.62	14.3	10.1
(60, 20, 200, 4000, 4, C_2)	96.88	3.8	4.6
(60, 20, 200, 4000, 4, C_3)	95.16	3.6	2.9
(60, 20, 200, 4000, 4, C_4)	98.47	3.8	5.2
(60, 20, 100, 4000, 4, C_1)	95.78	3.1	3.3
(60, 20, 400, 4000, 4, C_1)	96.53	3.9	3.1
(60, 20, 200, 4000, 1.6, C_1)	95.39	3.8	3.7
(60, 20, 200, 4000, 8, C_1)	94.72	3.8	3.5

Table 4-6. Results for problems with random load arrivals and random travel times.

Problem	Perf. rat. roll. hor.	Perf. rat. model in Sec. 3.1	CPU (sec.) our model	CPU (sec.) roll. hor.	CPU (sec.) model in Sec. 3.1
(60, 20, 200, 4000, 4, C_1)	105.21	103.84	3.7	237.1	3.4
(30, 20, 200, 2000, 4, C_1)	107.02	101.54	1.9	128.7	1.8
(90, 20, 200, 6000, 4, C_1)	103.17	103.28	6.0	334.9	5.7
(60, 10, 200, 4000, 4, C_1)	100.27	99.34	1.2	82.2	0.9
(60, 40, 200, 4000, 4, C_1)	105.92	104.29	14.3	1106.4	10.1
(60, 20, 200, 4000, 4, C_2)	103.85	102.17	3.8	1549.0	4.6
(60, 20, 200, 4000, 4, C_3)	108.05	101.14	3.6	233.5	2.9
(60, 20, 200, 4000, 4, C_4)	104.16	102.72	3.8	967.5	5.2
(60, 20, 100, 4000, 4, C_1)	106.89	105.99	3.1	274.9	3.3
(60, 20, 400, 4000, 4, C_1)	100.96	100.45	3.9	231.8	3.1
(60, 20, 200, 4000, 1.6, C_1)	104.15	103.61	3.8	228.3	3.7
(60, 20, 200, 4000, 8, C_1)	101.53	102.43	3.8	244.2	3.5

4.7.2 Computational Results

This section describes our computational results on the three aforementioned problem classes.

Problems with Deterministic Load Arrivals and Deterministic Travel Times. These problems can be formulated as a min-cost integer multicommodity network flow problem as

$$\max_{x} \sum_{t\in\mathcal{T}} \sum_{i,j\in I} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^{v}_{ijlt}\, x^{v}_{ijlt} \tag{30}$$

$$\text{subject to } -\sum_{\substack{j\in I:\\ t-\tau_{ji}\geq 1}} \sum_{l\in\mathcal{L}} x^{v}_{jil,t-\tau_{ji}} + \sum_{j\in I} \sum_{l\in\mathcal{L}} x^{v}_{ijlt} = \sum_{\substack{j\in I:\\ t-\tau_{ji}\leq 0}} f^{v}_{ji,t-\tau_{ji},1} \quad i\in I, v\in\mathcal{V}, t\in\mathcal{T}$$

$$\sum_{v\in\mathcal{V}} x^{v}_{ijlt} \leq D_{ijlt} \qquad i,j\in I, l\in\mathcal{L}, t\in\mathcal{T}$$

where we assume that the load arrivals and the travel times are deterministic, and we omit the integrality and nonnegativity constraints for brevity. The first set of constraints in problem (30) are the flow balance constraints. The expression on their right side is the number of vehicles of type v that reach location i at time period t and that were dispatched before the beginning of the planning horizon. As mentioned in Section 4.3.2, $f_1 = \{f^{v}_{ijs1} : i,j\in I, v\in\mathcal{V}, s = 1-\tau,...,0\}$ is a part of the problem data. We apply the algorithm in Figure 4-4 for 250 iterations and compare the objective value at the final iteration with the objective value of problem (30). Table 4-4 shows the ratio of the two objective values (multiplied by 100), along with the runtime per iteration of the algorithm in Figure 4-4. For majority of the test problems, our model yields results within 2% of the optimal solution. The runtime per iteration for our model increases linearly with the number of time periods and almost quadratically with the number of locations. We emphasize that the reported runtimes are for a serial implementation where the subproblems corresponding to different locations are solved sequentially. If one were to solve the subproblems corresponding to different locations in parallel, then the runtime per iteration would increase linearly with the number of locations as well. Figure 4-5 shows the progress of the objective value as a function of the iteration number for two test problems. The objective value increases smoothly over the first 50 iterations and stabilizes.

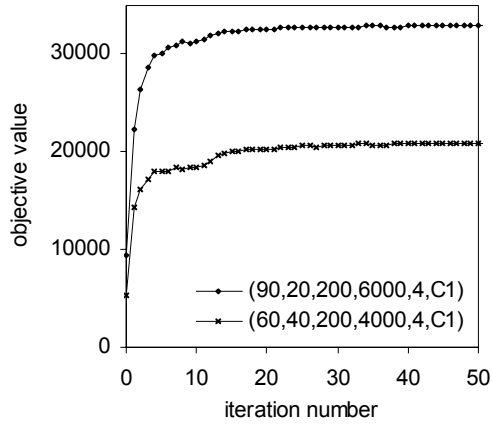


Figure 4-5. Progress of our model as a function of the iteration number for two test problems.

Problems with Random Load Arrivals and Deterministic Travel Times. For these problems, we use the model described in Section 4.3.1 as a benchmark. Topaloglu and Powell (2006) compare this model with a variety of benchmarks and report that it yields high-quality solutions.

In a stochastic setting, testing our model or the model described in Section 4.3.1 requires two sets of iterations. The first set of iterations, which we refer to as the training iterations, follow the algorithm in Figure 4-4 by solving the approximate subproblem (15) (or the approximate subproblem (8) if we are using the model in Section 4.3.1) for all time periods and updating the value function approximations. In the second set of iterations, which we refer to as the testing iterations, we stop updating the value function approximations and simply simulate the behavior of the policy characterized by the value function approximations obtained during the training iterations. The goal of the testing iterations is to evaluate the quality of the value function approximations that are obtained during the training iterations. We use 250 training iterations and 100 testing iterations.

In Table 4-5, the first column shows the ratio of the average objective values obtained in the testing iterations by our model and by the model described in Section 4.3.1 (multiplied by 100), whereas the second and third columns show the runtimes per iteration for the two models. For this problem class, our model lags behind the benchmark strategy with comparable runtimes. However, we emphasize that a parallel implementation speeds up the runtimes of our model by a factor of $|\mathcal{I}|$, but this is not possible for the model described in Section 4.3.1.

Problems with Random Load Arrivals and Random Travel Times. The first benchmark method we use for these problems is a common engineering practice called the rolling horizon strategy (see Topaloglu, 2005). This strategy assumes that future random variables will take on their expected values. It makes the decisions at time period t by solving a deterministic problem that "spans" the time periods t, t+1,…, $t + N$, where N is the rolling horizon length. In particular, for a given state vector f_t, arrival realizations A_t and load realizations D_t at time period t, the N-period rolling horizon strategy makes the decisions at time period t by solving the problem

$$\max_{x} \sum_{u=t}^{t+N} \sum_{i,j\in I} \sum_{l\in\mathcal{L}} \sum_{v\in\mathcal{V}} c^{v}_{ijlu}\, x^{v}_{ijlu} \tag{31}$$

$$\text{subject to} \sum_{j\in I} \sum_{l\in\mathcal{L}} x^{v}_{ijlt} = \sum_{j\in I} \sum_{s=t-\tau}^{t-1} A^{v}_{jist} \quad i\in I, v\in\mathcal{V} \tag{32}$$

$$-\sum_{\substack{j\in I:\\ u-\bar{\tau}_{ji}\geq t}} \sum_{l\in\mathcal{L}} x^{v}_{jil,u-\bar{\tau}_{ji}} + \sum_{j\in I} \sum_{l\in\mathcal{L}} x^{v}_{ijlu} = \sum_{\substack{j\in I:\\ t-\bar{\tau}_{ji}\leq t-1}} [f^{v}_{ji,u-\bar{\tau}_{ji},t} - A^{v}_{ji,u-\bar{\tau}_{ji},t}]$$

$$i\in I, v\in\mathcal{V}, u=t+1,...,t+N \tag{33}$$

$$\sum_{v\in\mathcal{V}} x^{v}_{ijlt} \leq D_{ijlt} \qquad i,j\in I, l\in\mathcal{L}$$

$$\sum_{v\in\mathcal{V}} x^{v}_{ijlu} \leq E\{D_{ijlu}\} \qquad i,j\in I, l\in\mathcal{L}, u=t+1,...,t+N,$$

where we use $\bar{\tau}_{ij}$ to denote the expected value of τ_{ij}. The problem above includes decision variables for time periods t, t+1,…, $t + N$, but we only implement the decisions corresponding to time period t and re-solve a similar problem when making the decisions for the next time period. Since $\sum_{j\in I} \sum_{s=t-\tau}^{t-1} A^{v}_{jist}$ gives the number of vehicles of type v available at location i at time period t, constraints (32) state that the total number of vehicles of type v that leave location i at time period t equals the number of vehicles of type v that are available at location i at time period t. Constraints (33) are the flow balance constraints analogous to the first set of constraints in problem (30). The expression $f^{v}_{ji,u-\bar{\tau}_{ji},t} - A^{v}_{ji,u-\bar{\tau}_{ji},t}$ on their right side is the number of vehicles of type v that were dispatched from location j to i at time period $u-\bar{\tau}_{ji}$ and that have not reached location i after having observed the arrivals at time period t. The rolling horizon strategy assumes that these vehicles will reach location i at time period u $(=u-\bar{\tau}_{ji}+\bar{\tau}_{ji})$. Since we

solve problem (31) after having observed the realizations of the random variables at time period t, the right side of constraints (33) involves known constants. The second benchmark strategy we use is a heuristic extension of the model described in Section 4.3.1. In particular, since this model cannot accommodate random travel times, we assume that the travel times take on their expected values and make the decisions accordingly. Topaloglu (2005) explains in detail how one can heuristically use the model described in Section 4.3.1 in the presence of random travel times. In Table 4-6, the first column shows the ratio of the average objective values obtained in the testing iterations by our model and by the rolling horizon strategy, whereas the second column shows the ratio of the average objective values obtained in the testing iterations by our model and the model described in Section 4.3.1 (all ratios are multiplied by 100). The third, fourth and fifth columns show the runtimes per iteration for the three models. The results indicate that our model performs noticeably better than both strategies. Interestingly, the runtimes for the rolling strategy can depend on the choice of the matrix C, but this does not seem to be an issue for our model. Also, with increasing number of locations, it is clear that implementing the rolling horizon strategy will be difficult.

4.8 CONCLUSIONS AND RESEARCH PROSPECTS

This chapter presented a dynamic fleet management model that can handle random load arrivals, random travel times and multiple vehicle types. Computational experiments showed that our model provides high-quality solutions within reasonable runtimes. An important feature of our model is that it decomposes the fleet management problem by time periods and by locations. When there are multiple dispatchers responsible from managing the vehicles at different locations, this feature allows them to concentrate on the vehicle supplies only at their own locations and they can coordinate their decisions with the help of the value function approximations.

The fleet management context provides a rich and challenging research area. There has been much advance in the last two decades but many practical issues remain unresolved. For example, we are not aware of a large-scale fleet management model that can handle load pick up and delivery windows and advance load information in a satisfactory manner, especially when the load arrivals are random. We heuristically used our model with some success in the presence of load pick up windows by simply keeping the loads in the system as long as their pick up windows are not expired. Nevertheless, the sound way to address load pick up and delivery windows is to include the loads in the state variable in the dynamic programming formulation, but it is not clear how to approximate the value

function when the state variable includes this extra load dimension. Other important issues that need attention are incorporating the terminal capacities, balancing the allocation of the vehicles over the network and the simultaneous management of different types of resources such as trucks, containers and drivers. The approximate dynamic programming paradigm combines the intelligence of optimization with the flexibility of simulation, and may provide remedies for some of these unresolved issues.

REFERENCES

Abara, J., 1989, "Applying integer linear programming to the fleet assignment problem", *Interfaces* **19**(4), 20-28.

Adelman, D., 2004, Price-directed control of a closed logistics queueing network, Technical report, The University of Chicago, Graduate School of Business.

Bourbeau, B., Crainic, T. G. and Gendron, B., 2000, "Branch-and-bound parallelization strategies applied to depot location and container fleet management problem", *Parallel Computing* **26**(1), 27-46.

Carvalho, T. A. and Powell, W. B., 2000, "A multiplier adjustment method for dynamic resource allocation problems", *Transportation Science* **34**, 150-164.

Chien, T. W., Balakrishnan, A. and Wong, R. T., 1989, "An integrated inventory allocation and vehicle routing problem", *Transportation Science* **23**(2), 67-76.

Crainic, T. G., Gendreau, M. and Dejax, P., 1993, "Dynamic and stochastic models for the allocation of empty containers", *Operations Research* **41**, 102-126.

Crainic, T. G. and Laporte, G., eds., 1998, Fleet Management and Logistics, Kluver Academic Publishers.

Dantzig, G. and Fulkerson, D., 1954, "Minimizing the number of tankers to meet a fixed schedule", *Naval Research Logistics Quarterly* **1**, 217-222.

Dejax, P. and Crainic, T., 1987, "A review of empty flows and fleet management models in freight transportation", *Transportation Science* **21**, 227-247.

Ferguson, A. and Dantzig, G. B., 1955, "The problem of routing aircraft - A mathematical solution", *Aeronautical Engineering Review* **14**, 51-55.

Frantzeskakis, L. and Powell, W. B., 1990, "A successive linear approximation procedure for stochastic dynamic vehicle allocation problems", *Transportation Science* **24**(1), 40-57.

Fumero, F. and Vercellis, C., 1999, "Synchronized development of production, inventory and distribution schedules", *Transportation Science* **33**(3), 330-340.

Godfrey, G. A. and Powell, W. B., 2002a, "An adaptive, dynamic programming algorithm for stochastic resource allocation problems I: Single period travel times", *Transportation Science* **36**(1), 21-39.

Godfrey, G. A. and Powell, W. B., 2002b, "An adaptive, dynamic programming algorithm for stochastic resource allocation problems II: Multi-period travel times", *Transportation Science* **36**(1), 40-54.

Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L. and Sigismondi, G., 1995, "The fleet assignment problem: Solving a large scale integer program", *Mathematical Programming* **70**, 211-232.

Jordan, W. and Turnquist, M., 1983, "A stochastic dynamic network model for railroad car distribution", *Transportation Science* **17**, 123-145.

Kenyon, A. S. and Morton, D. P., 2003, "Stochastic vehicle routing with random travel times", *Transportation Science* **37**(1), 69-82.

Laporte, G., Louveaux, F. and Mercure, H., 1992, "The vehicle routing problem with stochastic travel times", *Transportation Science* **26**(3), 161-170.

Powell, W. B., 1986, "A stochastic model of the dynamic vehicle allocation problem", *Transportation Science* **20**, 117-129.

Powell, W. B., 1988, A comparative review of alternative algorithms for the dynamic vehicle allocation problem, in B. Golden and A. Assad, eds., "Vehicle Routing: Methods and Studies", North Holland, Amsterdam, 249-292.

Powell, W. B., 1996, "A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers", *Transportation Science* **30**(3), 195-219.

Powell, W. B., Jaillet, P. and Odoni, A., 1995, Stochastic and dynamic networks and routing, in C. Monma, T. Magnanti and M. Ball, eds., "Handbook in Operations Research and Management Science, Volume on Networks", North Holland, Amsterdam, 141-295.

Powell, W. B., Ruszczynski, A. and Topaloglu, H., 2004, "Learning algorithms for separable approximations of stochastic optimization problems", *Mathematics of Operations Research* **29**(4), 814-836.

Puterman, M. L., 1994, *Markov Decision Processes*, John Wiley and Sons, Inc., New York.

Topaloglu, H., 2005, A parallelizable dynamic fleet management model with random travel times, Technical report, Cornell University, School of Operations Research and Industrial Engineering.

Topaloglu, H. and Powell, W. B., 2006, "Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems", *INFORMS Journal on Computing* **18**(1), 31-42.

White, W., 1972, "Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers", *Networks* **2**(3), 211-236.

White, W. and Bomberault, A., 1969, "A network algorithm for empty freight car allocation", *IBM Systems Journal* **8**(2), 147-171.

Chapter 5

INTEGRATED MODEL FOR THE DYNAMIC ON-DEMAND AIR TRANSPORTATION OPERATIONS

Yufeng Yao, Özlem Ergun and Ellis Johnson
School of Industrial Systems Engineering, Georgia Institute of Technology, GA 30332 USA

Abstract: On-demand air transportation is progressively obtaining the popularity with its flexibility, convenience, and guaranteed availability. However, its unique dynamic characteristics, such as short-noticed new demands and disruptive unscheduled maintenance, challenge the efficient operations, since they will significantly affect the priori algorithmic solutions. An integrated optimization model is presented to tackle the dynamic nature of the on-demand air transportation operations. A dynamic planning method together with a rolling-horizon approach is used to accommodate new demand. A realistic solution to recover from unscheduled maintenance events is also provided and demonstrated to be effective based on real world scenarios.

Keywords: air transportation; dynamic planning; on-demand; fleet assignment; aircraft routing; crew pairing

5.1 INTRODUCTION

In on-demand air transportation, passengers and cargo are transported by aircraft at a time designed by customers. The customers are able to fly directly anywhere within the network (for example, there are around 5,500 airports to support this service in the United States) anytime as they wish, with no delayed or cancelled flights, no check-in or security delays, no lost baggage concerns, and no naughty kids kicking their backs. With its flexibility, convenience, privacy, and guaranteed availability (certain advance notice required), this type of air transportation is widely used by private and corporate customers. It has a significant advantage over traditional commercial airline travel. Traveling with the executive business jets as needed has become progressively popular (Levere 1996; Keskinocak

1999; Michaels 2000), because customers can easily control their own schedule, avoid the loss of productive time, and get away from logistic constraints of standard commercial air travel.

Although convenient to customers, a management company needs to handle all the operational issues, such as scheduling, pilot training, aircraft maintenance, and so on. The management company faces enormous challenge that it must serve customer demands by efficiently utilizing its available resource, i.e. aircraft and crew. The system is operated in a nonscheduled mode, which means the demand is unknown in advance and dynamically received. A customer requests a flight, or a leg, by calling the company with the desired departure time, departure station, and arrival station only days or even hours ahead of time. The company generally does not change the customer's request and serve this requested flight on time. Typically, the company has to move empty aircraft frequently to pick up customers, which is called *reposition*. It is similar to the truckload pickup-and-deliver operations in trucking industry. On the other hand, if the company cannot cover the request flight by its own aircraft, a charter has to be rented with a much higher cost.

Besides the dynamic nature of the demands, the unscheduled maintenance is another factor of uncertainty significantly affecting the priori algorithmic solutions. When an aircraft requires unscheduled maintenance, the affected flights that are originally assigned to the aircraft have to be reassigned to other available aircraft to absorb the unexpected event.

The uncertainties limit the profit of the management companies with high reposition and charter costs. In this paper, we propose an integrated model to deal with the dynamic of the on-demand air transportation and minimize total operation costs. In addition, an experiment is provided and tested based on real world scenarios to recover from unscheduled maintenance with lower disruption and cost increase. This paper is organized as follows: section 5.2 describes the background and previous works in the field; section 5.3 introduces the integrated model and the proposed solution approach; section 5.4 presents computational results with real data sets; and section 5.5 wraps up the chapter with a conclusion.

5.2 BACKGROUND AND LITERATURE SURVEY

On-demand air transportation provides point-to-point (PTP) service at customer's request, and it has gain increasing popularity in recent years. One of the projects in NASA's Virtual Aerospace Modeling and Simulation Project (VAMS) aims to show PTP as a viable alternative to the traditional hub-and-spoke system, where it increases the capacity National Airspace System (NAS) significantly and provides a more desirable air transportation system (VAMS 2005). Few literatures have appeared to the public in this

new field of on-demand transportation. Cordeau et al. (2004) discussed the on-demand transportation service in four specific applications, which are dial-a-ride passenger transport service, urban courier service, dial-a-flight air charter service, and ambulance fleet management.

There are various forms of on-demand aviation, such as time-share, air charter, and fractional ownership of aircrafts. Among all the available forms, fractional ownership is the fastest growing program, and presents great potential (Sheehan 2003). To demonstrate the general operations of the on-demand air transportation, we focus on the fractional ownership program. Although the other forms are slightly different in their specific operations, the general principles discussed here are the same and can be easily extended to them.

In a fractional ownership program, a management company, providing the air transportation service, operates multiple types of aircraft, called fleets. Each fleet has its own specifications, such as seating capacity, fuel consumption, and speed. Only the pilots with specified certificate can legally fly it. Customers can purchase certain amount of flight time on a fleet. They are entitled to the time whenever they ask for, and the resource should be available as requested. Customers pay cost of ownership and a fixed monthly management fee, plus an hourly rate for all flights they requested.

5.2.1 Background of the Problem

Generally, the planning and scheduling process contains five phases in the airline industry: flight scheduling, fleet assignment, aircraft routing, crew scheduling, and crew assignment. Yu (1998) discussed a collection of articles on this field for commercial airline operations.

On-demand air transportation has its unique planning process. Typically, there is no prior *flight scheduling* phase since the flight is requested only days or hours ahead of time. In commercial airline, the *fleet assignment model* (FAM) determines for a specific fleet to fly a particular set of flights in order to maximize the profit. In the on-demand air transportation, it is mainly done according to the aircraft type the customer's requests, taking into account the fact that the company may provide complimentary upgrades with larger aircraft to reduce extra costs, such as charter or repositioning. *Aircraft routing* is to decide a sequence of customer flights and reposition flights flown by a specific aircraft with consideration of the scheduled and unscheduled maintenance events. In on-demand air transportation, reposition flights are generally inevitable.

Following the aircraft routing phase, *crew scheduling* or *crew pairing* is conducted. This phase is also very important since the crew cost is the second-largest operation expense after fuel cost. We define a *duty* as a sequence of flights and related activities, such as briefing and debriefing, within a crew work day. The *duty time* is the time span of a duty. A *pairing*

is a sequence of duties, which can be legally carried out by a crew over several days. A minimum *overnight rest* is required between two consecutive duties by the Federal Aviation Administration (FAA) regulation.

In addition, the fractional ownership requires a crew to have a seven-day on and seven-day off working plan. Therefore, a *duty period* for a crew is one week. After finishing the one-week duty period, the crew, called *off-duty* crew, goes back to its *crew base*, a designated station. The crew, called *coming-duty* crew, comes to work after one-week rest from its crew base. An off-duty or coming-duty crew can only travel by commercial airlines, and the travel is included in its duty. Some of pilots are willing to come to duty one day early or to stay one day late before or after their duty period. In both of these cases, the pilots are considered *overtime*.

5.2.2 Previous Work

The operation planning in the commercial airline industry has been addressed in numerous studies, and various solution methods have been developed for the commercial airlines. However, the literature on the scheduling problem for the on-demand air transportation service is just starting to appear. Keskinocak and Tayur (1998) study the fractional aircraft-scheduling problem for a single type of aircraft with no crew duty restrictions. Ronen (2000) presents a decision-support system for scheduling charter aircraft. Martin *et al*. (2002, 2003) extend the methods developed by Keskinocak and Tayur (1998) with multiple types of aircraft and crew constraints. Hicks *et al*. (2005) develop an optimization system for Bombardier Flexjet representing the aircraft itineraries and crew schedules.

Traditionally, sequential approaches are used for the scheduling optimization, where the optimal solution in each phase is based on the local optimal results from its previous phase. There are two major drawbacks: First, the optimal in each phase may not necessarily bring us to the global optimal; and the decisions made in preceding phases restrict the flexibility on finding feasible solutions in the following phases. As an alternative, integrating the phases is attractive, because all the phases are related, and better results can be achieved when everything is taking into account together. In commercial airline, the integration of two consecutive phases of the above three is discussed in recent literatures (Klabjan 2002, Mercier 2003, Cohn & Barnhart 2003). To the best of our knowledge, no research in the planning of the on-demand air transportation utilizes an integrated model.

5.3 THE INTEGRATED MODEL

Because the fleet assignment, aircraft routing and crew pairing are interdependent, it is ideal to consider them in a holistic manner. We put all three phases together in one comprehensive model to better reflect this requirement. In this model, the crew constraints in crew pairing are considered in the modified FAM, aircraft routing and aircraft maintenance are combined in crew pairing. Another advantage of the integrated model is that the crews are separated from aircraft, which increases the flexibility of the planning.

Generally, the management company prefers that a crew stay with a specific aircraft to save on travel time and cost, and simplify the operations. However, it restricts the utilization of the aircraft and crew. When an aircraft goes under maintenance, the crew associated with it becomes idle and wasting time on waiting. On the other hand, the aircraft is also not efficiently utilized. Since there is a mandatory maximum duty time per day for a crew, an aircraft has to stay on the ground when the crew has reached its duty time limit.

In our proposed model, the crew and the aircraft are no longer required to stay together all the time, so that an aircraft can be used by any available crew. For example, when crew A's original assigned aircraft goes under maintenance, crew A can take over crew B's aircraft to fly flight X, which crew B cannot do due to its duty time regulation.

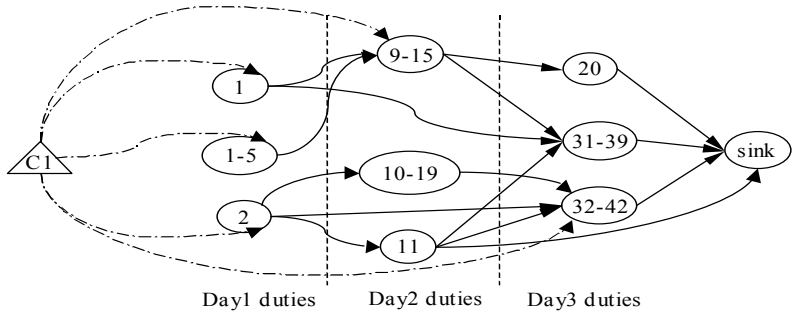
In addition, instead of focusing on a single day operation, we use three-day planning period to obtain near optimal arrangements across days. In the one day only optimization, the arrangement is a local optimal for that day, and the ending location of each aircraft will effect the reposition of the next day duties. Theoretically, the more days are included in the planning period, the closer a solution is to the global optimal. We select three days as our planning period based on the fact of the increasing number of unknown demands and high computational intensity after three days.

A rolling horizon approach is used to adapt the changing demands in the following days. In another word, after the first-day solution is obtained for a three-day problem, the assignment is fixed for the first day as part of the input, and the initial input for the next three-day problem is updated with new demand information.

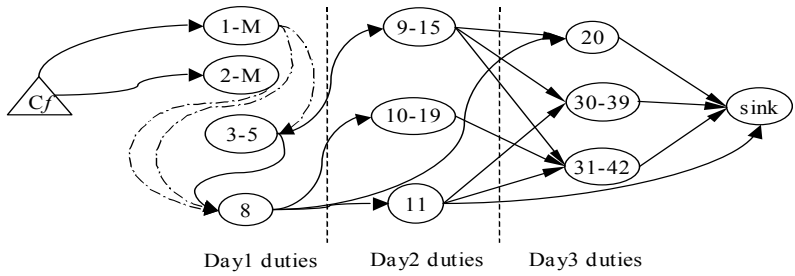
To build our model more efficiently, we create two types of networks: the crew network for each crew and the fleet-station time line. Details of the networks are discussed in the following sections.

5.3.1 Crew Network and Crew Reassignment

A crew network is constructed depending on the crew's remaining duty days in its duty period. In the crew network *G=(N, A),* the node set *N* consists of a source node *C* (or called *crew node*) representing the initial location of the crew, a set of duty nodes representing feasible duties that the crew can legally fly during the planning period, and a sink node representing the completion of a pairing. The customer flight cost is paid by the customer; hence, it is not in the optimization model. Two types of arcs are used to distinguish the activity costs of the crew traveling with commercial airlines (the ticket price and overtime cost if incurred) and covering the duty with reposition and upgrade. For example, if a crew goes off duty in the planning period, an arc exists from the last duty node the crew finished to the sink node with travel cost. Another example for a coming-duty crew is shown in Figure 5-1a. The arc, linking the crew node to the right side of a duty node with a dashed line, represents that the crew travels to the last arrival station of a finished duty to pick up a compatible aircraft. An arc connects two consecutive duties if the overnight rest requirement is satisfied. In addition, we allow the possibility that a crew stays on the ground for one or two days.



a. Crew comes on to duty on the second day.



b. Free crew pick up an aircraft at a station.

Figure 5-1 Partial crew networks for coming-duty and free crew.

When an aircraft needs to go under maintenance for a long time, the crew is then called *free crew*, and is free to be reassigned to another available aircraft. In Figure 5-1b, after the crew finishes flight 1 or 2 and takes its assigned aircraft to the maintenance station, it becomes a free crew. It can travel to the departure station of leg 8 if an idle aircraft located there and fly this duty. It can also travel to the last arrival station of a duty (including flights 3 and 5) that another crew finished and take over the aircraft there. In either case, the free crew transportation time and cost are taken into account. After the swap, the free crew becomes an on-duty crew who can fly either uncovered legs during the current day or the next day's early legs that cannot be legally flown by the original on-duty crew. In the meanwhile, the original on-duty crew becomes a new free crew that waits for reassignment.

The reassignment allows the crew to reach two duty nodes in one day. One is before the reassignment and the other one is after the reassignment. The whole duty for this crew should be indicated into two segments: an early duty whose last leg is the maintenance, and a later duty that the crew flies other flights with another aircraft. We give a simple example for one-day operations in a fleet in a time-space duty network (Figure 5-2).

Assume crews C1 and C2 are available at stations S1 and S2 respectively and three aircraft are available at stations S1, S2, and S3. The aircraft at S1 needs go under maintenance at S2. It is possible that C1 takes the aircraft to S2 for its maintenance service that starts after time B (early duty AB-MT). Here, MT represents maintenance leg. Note that MT only

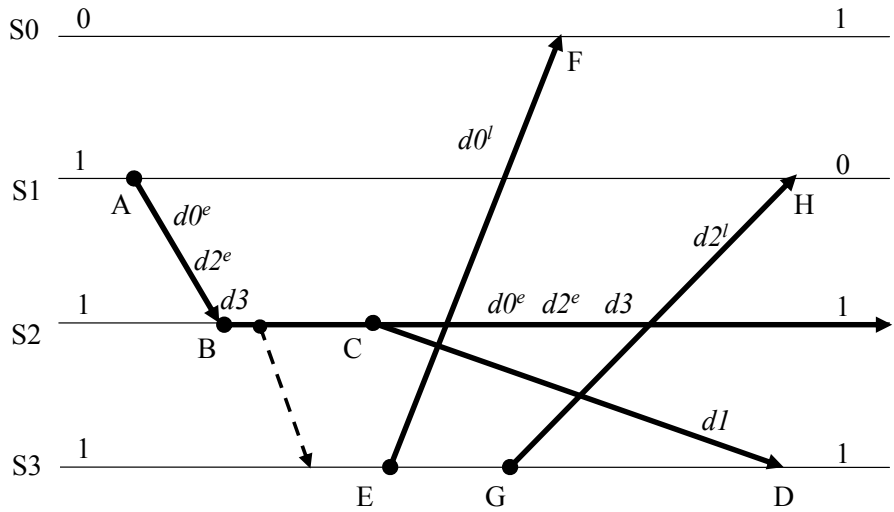


Figure 5-2 A time-space duty network for one fleet.

means the crew has to fly the aircraft to its maintenance station before the maintenance starts, not fly MT itself. Then C1 travels to S3 and finishes the later duty EF or GH with the idle aircraft. The dashed line indicates that the crew travels from S2 to S3 via commercial airline.

In this case, we have to divide both duties *d0* (AB-MT-EF) and *d2* (AB-MT-GH) into two segments to keep information about changing aircraft. If it is an AF duty or an AH duty directly, the solution will be infeasible since the segment EF or GH will have to be covered with an aircraft which is actually under maintenance. Another feasible duty for this C1 is *d3* (AB-MT) and then stays with the aircraft at the maintenance station S2. Crew C2 covers duty *d1* (CD). In Figure 5-2, the numbers on the left side are the number of initial planes on the ground at each station in the beginning of the planning. The numbers on the right side show the number of aircraft on the ground at the end of the first day given by the solution. The summation of all the left side numbers should equal to the summation of all the right side numbers to maintain aircraft conservation.

5.3.2 The Fleet-station Time Line

This time line records the departures and arrivals at the station for each fleet to preserve aircraft flow conservation. Hane (1995) originally creates it for commercial airline planning, and we reconstruct it to capture the characteristics of on-demand air transportation planning and meet the needs of our integration. A duty-based fleet-station time line, created based on the crew duty network, is shown in Figure 5-3. It contains ground arcs, crew's duty arcs, and nodes. A ground arc connects two consecutive nodes at one station in the time line. A duty arc indicates a crew's duty, containing a sequence of flights. In the duty-based fleet-station time line, a node represents the departure time of a duty or the ready time for the next take off. Note the start time of a duty is the latest time for a crew to cover the first leg in its duty on time. For instance, if the crew and aircraft are not at the first departure station, the reposition time needs to be considered in the start time of a duty. The ready time is the arrival time of the last flight in the duty plus minimum turn time.

Using the same example as in Figure 5-2, we demonstrate the duty activities in the fleet-station time line in Figure 5-3. The duties $d0^e$, $d2^e$, and d3 leave at point A and are ready at point BB when MT leg finishes its maintenance service. Duties $d0^l$, $d2^l$ and d1 leave at point E, G, and C and are ready at point F', H' and D' respectively. points F', H', and D' are shifted from F, H, and D by the minimum turn time. Arcs Z0 to Z11 are the ground arcs. Hence, given the initial number of aircraft on the ground at each station is expressed with $\{z_0, z_2, z_5, z_8\} = \{0, 1, 1, 1\}$, the final number of aircraft on the ground at each station is $\{z_1, z_4, z_7, z_{11}\} = \{1, 0, 1, 1\}$ in the solution.

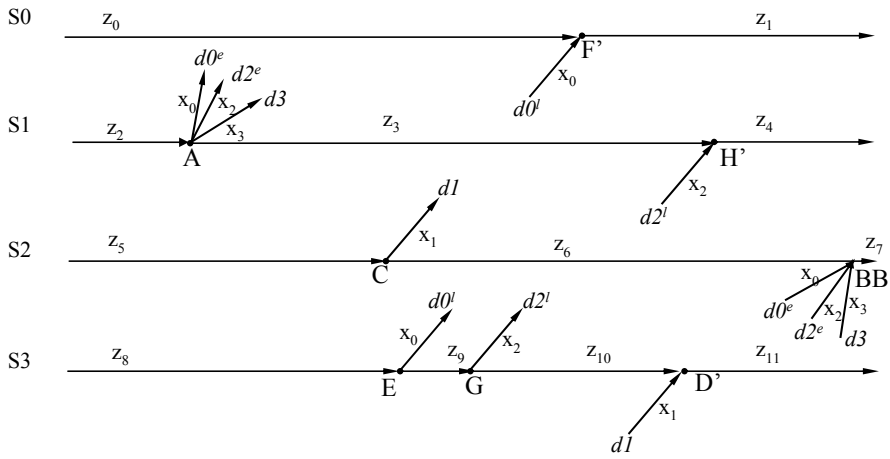


Figure 5-3 The station-fleet time lines for one fleet.

5.3.3 The Model Formulation

We present an integrated model with the objective to minimize the total cost, which consists of reposition cost, upgrade cost, travel cost and chartering cost. We define the following parameters:

L set of customer flights in the planning period,
W set of crews,
N^f set of nodes in the fleet-station time line network of fleet f,
T set of fleet type,
G^f set of ground arcs in the network of fleet f,
$G^f_{Initial}$ set of ground arcs before the first node in each station time line in fleet f,
CP^f set of all columns representing the possible pairings in fleet f,
$V(f)$ number of aircraft on the ground in fleet f in the beginning of a planning period,
c_j cost of column j, which includes reposition cost, upgrade cost and travel cost. A column is a feasible pairing for a fleet, since the crew for this pairing can only fly one specific fleet type.
r_k chartering cost for flight k
A_{kj} 1 if flight k is included in column j, and 0 otherwise.
B_{wj} 1 if crew w flies the sequence of flights in column j, and 0 otherwise.
C_{nij} 1 if pairing j has duty i and it enters node n, –1 if pairing j has duty i and it leaves node n in the net work of fleet f, and 0 otherwise.
D_{ngf} 1 if ground arc g leaves node n, –1 if ground arc g enters node n in the network of fleet f, and 0 otherwise.

The decision variables are:

x_j 1 if the solution picks a pairing at column j , and 0 otherwise.
s_k (slack variable) 1 if flight k is covered by a charter, and 0 otherwise.
z_{gf} the number of aircraft in fleet f on the ground arc g.

We order the flights in L with respect to the departure time in the planning period. Given the initial value of z_{gf} for each $g \in G^f_{Initial}$ in fleet f with $V(f)$ in the beginning of the planning period, The integrated model then is formulated as follows:

$$Min \quad \sum_{f\in T}\sum_{j\in CP^f} c_j x_j + \sum_{k\in L} r_k S_k$$

$$s.t. \quad \sum_{f\in T}\sum_{j\in CP^f} A_{kj} x_j + s_k = 1 \qquad \forall k \in L \qquad (1)$$

$$\sum_{f\in T}\sum_{j\in CP^f} F_{wj} x_j \le 1 \qquad \forall w \in W \qquad (2)$$

$$\sum_{i:\, j\in CP^f} C_{nij} x_j + \sum_{g\in G^f} D_{ngf} z_{gf} = 0 \qquad \forall n \in N^f, \forall f \in T \qquad (3)$$

$$z_{gf} = V(f) \qquad \forall g \in G^f_{initial}, \ \forall f \in T \qquad (4)$$

$$x_j \in \{0,1\} \qquad \forall j \in CP^f, \forall f \in T$$

$$s_k \in \{0,1\} \qquad \forall k \in L$$

$$z_{gf} \ge 0 \qquad \forall g \in G^f, \forall f \in T$$

Constraints (1) are the leg covering constraints, which require that every leg k in L to be covered either by a company's aircraft or by a charter aircraft. Constraints (2) are the crew constraints, which ensure that a crew is assigned to only one pairing. The aircraft balancing constraints (3) make sure the aircraft flow conservation. Constraints (4) initialize the number of planes available at the very beginning of each duty-based fleet-station time line.

5.3.4 Solution Algorithm

The proposed model is solved with column generation technology. We first solve the linear programming (LP) relaxation of the problem formulated in the above section. Initially, we enumerate all feasible duties that can be legally operated in a day, by using a depth first search algorithm. These duties are made up of customer and repositioning legs and scheduled maintenance events. Then we create an auxiliary network (described in section 5.3.1) for each available crew pair that is used to identify good pairings. The good pairings can be found with Dijkstra's shortest path

algorithm. Note that the nodes in the crew network are connected in a time-dependent order, and the Dijkstra's algorithm we implemented here will not form cycles. Shortest paths on these auxiliary networks are used to create a set of initial columns feeding into the initial LP. To maintain feasibility, slack variables indicating charters for the customer legs are attached to the initial LP. After solving the initial LP, we update the arc costs on the auxiliary networks using dual information provided by the LP, and solve a pricing problem by finding shortest paths on these networks with the new arc costs. We also construct the station-fleet time lines (described in section 5.3.2) for each fleet to ensure the aircraft conservation. While filling the selected columns, the ground arc information in the LP is recorded. The pricing out process terminates when there does not exist negative reduced cost column.

When we have an optimal solution for the LP relaxation, we feed all the columns present in the final LP into an Integer Programming solver. After the integer solution is obtained by solving the IP with all the existing columns in the final LP relaxation, the solution for the first day is obtained. The procedure is illustrated in Figure 5-4.

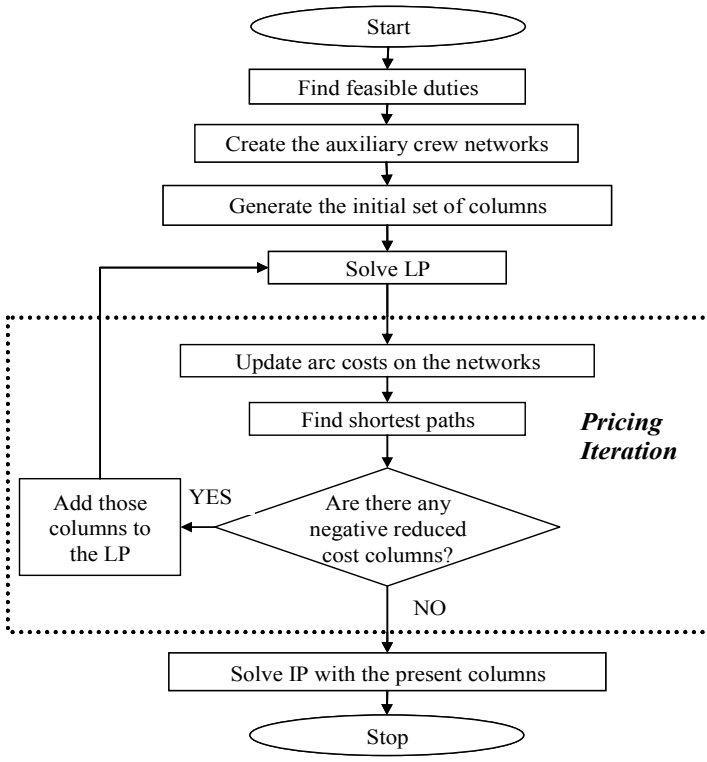


Figure 5-4. Flow chart of the solution procedure.

Table 5-1 Computational efficiency and effectiveness on different instance sizes.

# of fleets, crews, legs	Work load	Percentage of unscheduled MT	Solution time (sec)	Optimality gap (%)	Reposition ratio with MT
3, 35, 58	1.66	5.17	0.9	0.00	0.30
3, 61, 112	1.84	6.25	2.4	0.00	0.31
5, 75, 192	2.56	5.21	5.8	0.00	0.22
5, 131, 265	2.02	7.17	28.3	0.02	0.32
5, 150, 251	1.67	4.78	24..3	0.04	0.28
3, 61, 261	1.84	6.25	62.1	0.01	0.31
3, 75, 313	2.56	5.21	224.8	0.05	0.24
5, 150, 410	1.67	4.78	563.2	0.06	0.29

The rolling-horizon approach is then applied to continue the planning process, where the first day pairing for the three-day planning horizon is fixed and the procedure is repeated for the next three days using the first day information as initial conditions. More detail is described in Yao *et al.* (2007).

We carry out a set of experiments (Table 5-1) to evaluate the computational efficiency and effectiveness of our solution approach on different instance sizes. The instances for this study are generated based on the demand and maintenance data obtained from a fractional management company. Among the eight test scenarios, the first five rows are planning problems for a single day, and the last three rows have a three-day planning horizon. The size of the instance is given in first column as the number of fleets, crews, and legs. We define the crew *work-load* as the number of legs in the first day divided by the number of crews, which represents the average number of legs flied by a crew. The third column lists the percentage of the unexpected events, i.e. unscheduled maintenance, in the first day. The solution time given in column four is the total time required for solving the LP and the IP after preprocessing. To examine optimality, we use the value obtained from the LP as a lower bound on the optimal value of the IP. The fifth column is the gap between the value of the integer solutions and the LP lower bound. Column six lists the reposition ratio, a ratio of the reposition hours to the total flights hours including customer leg hours and reposition hours, when scheduled and unscheduled maintenance are taken into account. The test shows that our proposed approach is efficient and effective on different problem sizes and planning horizons.

5.3.5 Dynamic Plan Adjustment to Handle Uncertainty

The dynamic nature of the on-demand transportation mainly comes from two sources: late noticed demand information and unscheduled maintenance. We discuss them in the following sections.

5.3.5.1 Demand uncertainty

As mentioned before, the customer's demand could come as late as only eight hours before the desired departure time. Based on a historical data analysis, approximately 5% of the demand is still unknown for the first day when its schedule is released, 10% and 20% of the demand is unknown in the second and third day, respectively.

We explained the use of the rolling horizon approach on dealing with the unknown demands in the following days. However, for those 5% unknown demand in the first day, we need to adjust the plan dynamically. When a new request for the same day service comes, the flight assignments before the time when the request comes are unchangeable since they are already in execution. The optimizer reruns the model with a portion of the assignments fixed (which include legs, crews and aircraft), and the rest of assignments is re-calculated with updated demand information and availability information of the resources. It is worth to mention that the approach proposed here does not depend on the forecasted volume of the unknown demand, which is only used to simulate the dynamic input in our computation experiment. No matter the unknown demand is 5% or 20%, the quality of the results will not be affected, however, the increase in unknown demand will reduce the feasible space due to time constraints.

There are three rules to determine the resource available time and location. An aircraft that just finished a customer flight is available immediately at its current station. An aircraft that is flying a customer flight will be available after it lands at the arrival location; An aircraft which is repositioning on its way to a station to pick up a customer will be available after it finishes the customer leg at the arrival station.

A flexible departure time window, which shifts the flights by up to one hour earlier or later than its desired departure time, is an option to ease the planning with the short-noticed same-day demand if the customer agrees. For a demand that may need charter, we allow a limited flexibility by possibly shifting the departure time of a customer leg in a narrow time interval. The allowed flexibility in the departure times is taken into account when all the feasible duties are generated initially, and it produces more duties.

5.3.5.2 Uncertainty on aircraft availability

Unscheduled maintenance may happen any time during the day. In our model, both scheduled and unscheduled maintenance are taken into account. A maintenance service is treated as a mandatory leg that a specific aircraft must cover. The scheduled maintenance is considered in the priori plan, so that the aircraft will be flown to the specific station by the schedule time. However, the unscheduled maintenance is an unexpected

event, and it causes higher charter and travel costs. When an aircraft is down, we have to reassign the affected legs.

Similar to cover a new demand, we first fix those assignments that started before the disruption, and run the optimizer with updated available time. The difference is that the unscheduled maintenance occurs suddenly and usually need to be taken care right away, while the new demand has at least eight hours advanced notice. It posts a greater challenge to the planning on reducing the disruption. Since our integrated model separates the crew and the aircraft, the disabled aircraft will not hold the crew, and we will have more chance to recover from the disruptive events.

5.4 COMPUTATIONAL EXPERIMENTS

In this section, we use the proposed model to solve the dynamic scheduling problem. Based on the real operational data provided by a fractional ownership management company, we first optimize the schedule, and then illustrate the capability of our approach to reduce the impact of the first day new demand and unscheduled maintenance on the established schedule. Then we test the possibility to fly a flight that originally has to be covered by a charter with flexible departure time window.

5.4.1 New Demand without Time Window

With total 60 aircraft in 3 fleets, we have 186 (91/57/38) customer flights and 25 maintenance services scheduled in three days. To simulate the real-time dynamic scenario, the historical data is converted to include uncertainty. Assume we carry out the planning at the night before the first day. Since the demands are not completely known at that point, we randomly hide 4/6/6 of customer legs, which is 5%, 10%, and 20% of demand in 91/57/38 flights, from first/second/third day demand respectively, and use the remaining data for our initial planning. The removed legs will be added back later as the new demands.

After we have the results from first round optimization, the schedule and route arrangements are sent to the crews for execution. In the morning of the first day, the crews start with either customer legs or reposition legs. New customer demands could come in anytime during a day. For instance, at 10:30am, a new demand to fly from BOS to PDK departure at 8:30pm in the current day is received, which is in fact a customer leg we previously removed from the given data. Since it has a 10-hour advance notice, the company will try to make it fit in the existing plan. From the time the planner is notified with the new demand, we need to adjust the input to our program to reflect the change. Usually the crews get to know their duty flights during the briefing time. The notice to a crew for changing flights

should be given at least two hour before the change. Therefore, all assignments or related activities started before 12:30pm will be fixed as originally planned, which means they will not be active in new planning. The available times and locations of crew and aircraft after 12:30pm are checked and updated. The new solution will be reached after re-run the model with the new inputs. The columns we generated in the initial planning are treated as follows: The columns representing fixed assignments are taken as a part of the updated solution, and will not reenter the optimization; the other columns are not reused since the input has changed.

Table 5-2 shows the results of this process. The break down solution for the original planning is listed in the first row, the solution after adding 4 new customer flights is shown in the second row, and the impacts when two unscheduled maintenance services occur during mid day are indicated in the third row. One good evaluation criteria is the gross profit margin, which is the profit over the cost. However, the revenue information is unavailable, we use the reposition ratio as a solution quality measurement combined with the total cost. The results shown below are better than the company's current practice.

The original plan handles 87 customer flights with two charters, six upgrades from fleet 1 to fleet 2 and one upgrade from fleet 2 to fleet 3. The updated real-time solution after adding four new flights does not require additional charter with a 13.7% increase on total cost, which is mainly due to the increase on reposition cost. Eleven crews need to change its original assignment with a two-hour advance notice.

To demonstrate the benefit from the proposed model, we compare our solution with other possible solutions. One obvious solution is to keep the original plan as it is, and use some unassigned aircraft to cover the new legs if possible. Fortunately, there are two aircraft free in the original plan. However, this solution posts over 40% improvements in total cost with two additional charters and two long repositions. Currently, the planner manually adjusts the assignments and tries to fit in the new demands; however, it is hard to achieve optimal results in such a complex problem. With the proposed model, significant cost saving can be created with quick re-optimization.

In the experiment, two aircraft are unexpected down before all demands are realized. The effect of the unscheduled maintenance is demonstrated in the third row of the Table 5-2. Although one of the four affected customer flights has to be chartered, all other three flights originally assigned to the above two aircrafts can be reassigned to other aircrafts in the same fleet with longer repositions after the incidence. Hence, the recovery is efficient, and the impact is limited.

Table 5-2 Break down solution on different scenarios.

	Rep. Ratio	Rep. cost	Upgrade cost	Charter cost	Total cost
Original plan	31.35%	113,849	1,445	37,310	152,604
Add new legs	34.17%	134,053	1,450	37,960	173,463
Unscheduled MT	34.34%	134,231	1,450	53,530	189,211

Table 5-3 Comparison between without and with a time window.

	Rep. Ratio	Rep. cost	Upgrade cost	Charter cost	Total cost
No time window	34.34%	134,231	1,450	53,530	189,211
With time window	34.34%	134,231	1,450	37,960	173,641

5.4.2 New Demand with Time Window

In the above section, some flights have to be covered by charter after the optimization. If a demand is allowed within a specific time window, the chance of covering is increased significantly. However, not all customers allow a time window. Assume we can use a flexible departure time to one chartered flight. A flag that indicates a one-hour time window is taken into account for the specific demand, the feasible duty nodes are increased from 8244 to 8260, which means sixteen extra options are available. The results are compared in Table 5-3 between a rigid pick up time and a flexible pickup time, and the later one covers the new demand without chartering. The total cost decreases by 8%, which is a substantial saving, after the pick up time shifts 28 minutes.

5.5 CONCLUSIONS

In this chapter, an integrated model is introduced to solve the fleet assignment problem simultaneously, aircraft routing and crew pairing for the on-demand air transportation services. In the model, crew duty networks and fleet-station time lines are created for each fleet so that the crew and aircraft information are embedded in the fleet assignment problem, and the crew can be separated from the aircraft during the planning. To avoid enumerating millions of potential pairings, we generate columns by using a specialized shortest path search on the network.

Short-noticed new demands and unscheduled maintenance present the uncertainty in the on-demand air transportation service, and prevent resource to be utilized optimally. To adapt the dynamic feature of the service, special strategies are presented to reassignment of crew and aircraft when the new demand and unscheduled maintenance occur during the current day. The rolling horizon period is used in the optimizer to capture the unknown demand in the following days. The computational experiments show the capability of model to effectively deal with disruptions in the real world scheduling, and indicate the flexible pick up time benefit the company to avoid costly charters.

REFERENCES

Cohn, A.M. and Barnhart, C., 2003, Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions, *Operations Research,* **51**(3)**:**87-396.

Cordeau, J-F., Stojkovic, G., Soumis, F., and Desrosiers, J., 2001, Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling, *Transportation Science* **35:**375-388.

Cordeau, J-F., Laporte, G., Potvin, J-Y., Savelsbergh, M.W.P., 2004, Transportation on Demand; http://www2.isye.gatech.edu/~mwps /publications/TransOnDemand.pdf.

Hane, C., Barnhart C., Johnson E.L., Marsten, R., Nemhauser, G., and Sigismondi, G. 1995, The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming,* **70:**211-232.

Hicks, R., Madrid, R., Milligan, C., Pruneau, R., Kanaley, M., Dumas, Y., Lacroix, C., Desrosiers, J., Soumis, F., 2005, Bombardier Flexjet Significantly Improves Its Fractional Aircraft Ownership Operations, *Interfaces*, **35**(1)**:**49-60.

Keskinocak, P., 1999, Corporate high flyers, *OR/MS Today*.

Keskinocak, P. and Tayur, S., 1998, Scheduling of Time-Share Aircraft, *Transportation Sciences*, **3:**277-294.

Klabjan, D., Johnson, E.L., Nemhauser, G.L., Gelman, E., and Ramaswam, S., 2002, Airline crew scheduling with time windows and plane count constraints, *Transportation Science*, **36:**337-348.

Levere, J., (July 21, 1996), Buying a Share of a Private Aircraft, *The New York Times*.

Martin, C., Jones, D., and Keskinocak, P., 2003, Optimizing On-Demand Aircraft Schedules for Fractional Aircraft Operators, *Interfaces* **33:**22-35.

Martin, C., Jones, D., and Keskinocak P., (November 17-20, 2002), Bitwise Fractional Airline Optimizer, *INFORMS*, San Jose.

Mercier, A., Cordeau, J.–F., and Soumis, F., A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*.

Michaels, D., 2000, Fractional Ownership Gets Easier and Cheaper in Europe, *The Wall Street Journal*.

Ronen, D., 2000, Scheduling Charter Aircraft, *Journal of Operational Research Society* **51:**258-262.

Sheehan, J.J., 2003, *Business and Corporate Aviation Management: On-Demand Air Transportation*, MCGRAW-HILL, New York, NY 10121-2298.

VAMS (Virtual Aerospace Modeling and Simulation Project) (2006) *Concept PTP: Massive Point-to-Point On-Demand Air* Transportation. http://vams.arc.nasa.gov/activities/ptp.html

Yao, Y., Ergun, O., Johnson, E., Schultz, W., Singleton, J.M., Strategic Planning in Fractional Aircraft Ownership Programs, *European Journal of Operational Research*, to appear.

Yu, G., 1998, *Operations Research in the Airline Industry*, Kluwer Academic Publishers.

Chapter 6

AN INTERMODAL TIME-DEPENDENT MINIMUM COST PATH ALGORITHM

With an Application to Hazmat Routing

Elaine Chang[1], Evangelos Floros[2] and Athanasios Ziliaskopoulos[2]
[1]Jeppesen, 1800 McGill College Ave. #1930, Montreal, QC, H3A 1R9, Canada; [2]Department of Industrial Engineering, University of Thessaly, Volos, Greece

Abstract: Transportation problems, in terms of both passenger and freight applications, are increasingly being addressed with inter-modal solutions. This chapter discusses the problem of computing optimum paths on a network with many modes of transport and time-varying link costs and travel times, accounting for the fixed schedule modes and mode-switching delays. An efficient algorithm is introduced that computes optimum path trees from all nodes and possible discrete departure times, while accounting for travel and transfer delays, as well as differences in perceived costs associated with specific modes and transfers. The algorithm, called the time-dependent inter-modal minimum cost path (TDIMCP) algorithm, is extended to set the necessary framework for solving the problem of inter-modal routing of hazardous materials, taking into consideration both risk and cost at the transfer points and travel links. Travel and transfer risk associated with hazmat routing are incorporated into the cost calculation of the TDIMCP problem, considering both the likelihood of an incident and the consequences of that incident. The inter-modal hazmat routing algorithm is then applied to a series of scenarios on a test network to illustrate the behavior of the algorithm.

Keywords: hazardous materials; minimum cost paths; multi-objective; shortest paths.

6.1 INTRODUCTION

Transportation problems, in terms of both passenger and freight applications, are increasingly being addressed with inter-modal solutions to take advantage of different speeds, safety levels, reliability levels, costs and geographic reaches of different modes. Further, technological advances have produced sophisticated tools for monitoring and communicating vehicle

locations and network interruptions in real-time, so routing decisions may be optimized based on precise real-time information.

While the fleet management problem typically considers a fleet of vehicles in a single mode, some systems, such as major courier or shipping companies, include multimodal fleets, and must thus optimize several modes of vehicle routes to support inter-modal movements. In such cases, efficient calculation of inter-modal routes is fundamental component of a transportation management center's fleet routing task. Further, even for single-mode fleets planners of fleet routes would benefit from an understanding of travelers' or shippers' full transportation needs, which may include transfers to and from other modes. Moreover, time-varying network conditions and vehicle location data allow for more efficient routing, and add further flexibility if such detailed information is available on a real-time basis.

This chapter discusses the problem of computing optimum paths on a network with many modes of transport and time-varying link costs and travel times, accounting for the fixed schedule modes and mode-switching delays. An efficient algorithm is introduced that computes optimum path trees from all nodes and possible discrete departure times, while accounting for travel and transfer delays, as well as differences in perceived costs associated with specific modes and transfers. Each mode has a different cost structure (actual or perceived) which is not only a function of the travel time but incorporates factors such as fare, level-of-service, walking and waiting time.

In this chapter, we use the terms "time-varying" or "time-dependent" to describe variables and parameters that may take different values at discrete time intervals. In contrast, this chapter reserves the term "dynamic" to describe a model that may be used to capture or optimize a system in real-time. Real-time dynamic models are typically solved repeatedly on a rolling horizon that captures or optimizes a certain time period into the future. Dynamic models require time-dependent variables and parameters in order to capture temporal detail and changes expected in the system; however, time-dependent variables may be equally applied to static (rather than dynamic) models that may be solved off-line, rather than in real-time. Since static models are solved off-line, they do not capture real-time changes in the system, so their solutions may not account for current realities. Dynamic models are more powerful than static models, since they are able to adapt to changes in the system; however, they require real-time data, as well as solution algorithms that can be quickly solved. Assuming that real-time data is available, this chapter addresses the second issue by presenting an algorithm that is computationally efficient, making it suitable for application to actual networks for both planning and real-time operations.

The algorithm, called the time-dependent inter-modal minimum cost path (TDIMCP) algorithm, is extended to set the necessary framework for solving the problem of inter-modal routing of hazardous materials, taking into consideration both risk and cost at the transfer points and travel links. Hazmat transportation is an important economic activity in industrialized countries due to the need to move a large number of hazmat shipments from production to consumption sites. With globalization, these distances tend to increase as production sites shift to countries with more favorable labor conditions. In addition to these concerns, the threat of a deliberately caused hazmat incident by terrorists is also a constant concern. The need to provide safer living conditions in every aspect of human activity is a priority for all nations. Minimizing hazmat transportation risk serves this common goal. Given, for example, that rail transportation is considered a safer mode than highway modes but that the highway network is more extensive than the rail network, intermodal/multimodal transportation is a very attractive alternative to single mode hazmat transportation.

Travel and transfer risk associated with hazmat routing are incorporated into the cost calculation of the TDIMCP problem, considering both the likelihood of an incident and the consequences of that incident. The inter-modal hazmat routing problem is then efficiently solved using the proposed algorithm.

In other words, a framework for considering risk in intermodal paths is developed. It specifically provides the fleet manager in the real world with a tool that produces off-line the least risky intermodal routes for hazmat movements based on available risk information data of all possible routes. With the framework and algorithm, it is also possible to exploit any incoming on-line information concerning risk parameters changes and recalculate the best routes because of its computational efficiency.

A review of shortest path and hazmat literature are presented in the next section, followed by the formulation of the TDIMCP problem. Next the TDIMCP algorithm is presented, along with a discussion of the algorithm's properties. The proposed algorithm is then extended to the hazmat problem of routing with minimum risk, and the approach is applied to a series of scenarios on a test network. The chapter closes with concluding remarks and a discussion of continuing research.

6.2 BACKGROUND

The time-varying inter-modal minimum cost path algorithm builds on the area generally known as shortest path algorithms. Further, the application of this algorithm to the hazmat routing problem builds on past studies in hazmat routing.

6.2.1 Shortest Path Algorithms

Many time-dependent optimum path algorithms have been introduced in the literature (Ziliaskopoulos and Mahmassani, 1993; Kaufman and Smith, 1993, Chabini, 1998; Ahuja *et al.*, 2002); most of these algorithms tend to be concerned with a single mode (usually highway) in which case link costs are typically assumed to consist only of travel time. Introducing multiple modes, however, entails accounting for the perceived costs of driving versus waiting at a bus stop, paying a fare and riding a bus. In such cases, accounting for link costs is not as straightforward as replacing time with cost and then running a time minimizing algorithm, as is often done in the static case, but rather it requires designing a new algorithm to maintain consistency of both cost and time across time-dependent movements.

The literature on intermodal shortest path computations is mainly limited to static transit networks (Dial, 1967; Nguyen, Morello and Pallotino, 1988; Spiess and Florian, 1989; Nguyen, Pallotino and Malucelli, 2001). Most of these approaches approximate transfer waiting time based on headway or by creating artificial waiting time arcs. In freight applications (Crainic and Rousseau, 1986; Jourquine and Beuthe, 1996) multimodal paths are calculated based on costs and estimated delays, but are not time-dependent, and thus do not consider movement schedules.

For the passenger assignment problem, Pallotino and Scutella (1998) proposed a chronological algorithm, which expands the network temporally, including only paths that are viable in terms of the transit schedule. The algorithm considers the waiting time at nodes based on scheduled arrival times, and further, tracks the number of modal transfers in a path as an attribute in the multicriteria shortest path, such that paths with excessive numbers of transfers could be excluded. Battista, Lucertini and Simeone (1996) and Lozano and Storchi (2001) proposed multimodal shortest path algorithms, in which paths with illogical sequences of used modes were eliminated. Recently, Lozano and Storchi (2002) extended their algorithm to calculate the hyperpath of viable paths. Similarly, Sherali, Hobeika, Kangwalklai (2003) proposed an algorithm in which labels denoting particular modes of travel must conform with admissible strings of labels. Abdelghany and Mahmassani (2001) proposed an algorithm that calculates intermodal paths based on a multi-objective shortest path algorithm, where the set of non-dominated paths is computed, from which an optimum path is selected based on a generalized cost function.

A time-dependent least intermodal time path (TDLITP) algorithm was proposed by Ziliaskopoulos and Wardell (2000). This algorithm extends the time dependent shortest path algorithm of Ziliaskopoulos and Mahmassani (1993) to detect time-dependent intermodal least-time paths on a multimodal

transit network with dynamic travel times and mode-transfer delays. The TDLITP algorithm operates in a label-correcting manner assuming that dynamic travel times of links and actual schedules of transit or freight lines are known.

The work presented in this chapter extends the TDILTP algorithm to calculate time-dependent intermodal minimum cost paths (TDIMCP) from all origins and departure times to a destination node, based on time-dependent and fixed travel and transfer costs. The TDIMCP appears similar to the least time algorithm, however, it solves a different problem as cost rather than time is optimized; unlike the well-known time-invariant problems where cost and time can be used interchangeably, on a time varying network both time and cost labels need to be maintained at each node, since the cost is time-dependent. In other words, the TDIMCP problem treats time as an index, which determines the feasibility of a path, while a separate cost function is optimized.

6.2.2 Hazmat Transportation Problem

Hazardous material (hazmat) refers to any material whose transportation has the potential to cause harm to people, property or the environment. While moving hazmat is necessary, authorities are increasingly concerned about the risks associated with these movements and the catastrophic consequences of possible accidents. Attempts to address the problem focus on setting the necessary framework to achieve two complementary goals: (i) to develop risk parameters and methods of quantifying transportation risk and (ii) to efficiently formulate and solve the problem of routing hazmat, so that risk is minimized without unreasonably increasing transportation cost.

Alp (1995) provides a comprehensive analysis of quantitative risk assessment methodologies as applied to hazmat transportation. He presents commonly used techniques and provides a review of measures of risk acceptability. Literature on hazmat routing is limited to single mode minimum risk computations, both static and time-dependent. List *et al.* (1991) provide a thorough survey of the literature regarding risk analysis, routing/scheduling of shipments and facility location; their review emphasizes the need to take into account a variety of competing factors. They further recommend analysis of the transportation network in close relation to its environment. Most approaches incorporate these guidelines in trying to calculate optimum paths for hazmat transportation by minimizing cost and a variety of risk estimates. Huang and Fery (2005) introduced a single mode, static approach that treats the problem as one with multiple objectives, computing multiple solutions and selecting the Pareto optimal ones. McCord and Leu (1995) formulated the single hazmat shipment

routing problem in terms of multi-attribute utility theory. They considered the attributes of transportation cost and exposed population and showed that the single mode problem is solved with a static shortest path algorithm that serves as an effective route generator.

In addition, a widely accepted methodology is to use a weighted average function of transport risk and corresponding transport cost. The problem thus is transformed into a single objective one and is solved with an appropriate algorithm (Brainard *et al.*, 1996). Zografos and Androutsopoulos (2004) considered the problem as a bi-objective one (cost and risk), with specific service time windows and proposed a heuristic algorithm. Frank *et al.* (2000) developed a method for selecting the appropriate route among a set of routes with parameters being the exposed population and travel time. Sivakumar, Batta and Karwan (1995) consider the problem of finding a set of routes for shipping an extremely hazardous material to minimize the expected risk of the first accident, subject to constraints on the expected a priori risk, transport cost and equality of risk. The problem is formulated as an integer programming problem and a column generation technique is devised for solution. Furthermore, the SafeStat website (Federal Motor Carrier Safety Administration, 2005) provides an automated, data-driven analysis system to measure relative motor carrier fitness. The model considers state-reported motor carrier crashes, driver compliance reviews, closed enforcement cases, vehicle roadside inspections and census information to rate motor carrier safety. SafeStat is not intermodal and does not calculate intermodal routes.

To address the need for analytical tools to support hazmat decision-making in multimodal networks, the proposed TDIMCP algorithm is applied. The proposed approach computes the intermodal/multimodal path with the minimum time-varying weighted average of travel risk and travel cost. Travel risk is considered to include the consequences of an accident weighted by the likelihood of occurrence of such an accident and takes into consideration risk both at transfer points and travel links, since a number of past incidents have shown that the risk of transfers is at least as great as the risk of the transportation itself.

6.3 PROBLEM FORMULATION

The TDIMCP algorithm finds the minimum intermodal cost path on a multimodal network, given time-dependent link and transfer travel times, and fixed transportation costs. The network is represented by a directed graph, *G=(N,A,T,M)*, where *N* is the set of nodes, *A* is the set of arcs, *T* is the discretized time period of interest, and *M* is the set of modes. Each arc *(i,j)* is assigned a set of non-negative travel times $\tau_{ij}^{m}(t)$ associated with mode *m* when departing node *i* at time *t*. In addition, arcs can be assigned a fixed

travel cost, $\mu_{ij}^{m}(t)$, to represent tolls that must be paid to travel on arc *(i,j)* using mode *m*, when departing from node *i* at time *t*. The arc cost function, Equation (1), is a combination of the travel time-related and fixed travel costs, where the travel time-related cost is assumed to be a linear function of the travel time. The parameter, α^{m}, reflects the level of dislike that travelers tend to have for a certain mode. For example, buses, which are less comfortable than private cars, would have a higher value of α^{m}, resulting in a higher travel cost for the same amount of travel time.

$$\phi_{ij}^{m}(t) = \mu_{ij}^{m}(t) + \alpha^{m} \cdot \tau_{ij}^{m}(t) \tag{1}$$

Travel costs and times may be assigned to discrete time intervals, *t*. The size of the time interval dictates the amount of temporal detail captured in the model, and as always, a trade-off exists between computational efficiency and model detail. The appropriate level of discretization will be determined by the specific application; for example, to capture the effect of roadway congestion on truck movements, a five-minute time interval might be used, while schedule-based rail, sea or air movements might only require discretization to one hour or longer. Since the model is intermodal, the mode with most detailed level discretization required will dictate the discretization for the whole model.

Mode-switching arcs are also assigned travel times and costs. Switching delays between modes are represented by $\xi_{ijk}^{m_1 m_2}(t)$, to describe the time required to switch from mode m_1 to mode m_2 when traveling from node *i* through node *j* to node *k*, departing from node *j* at time *t*. For example, $\xi_{ijk}^{m_1 m_2}(t)$ might represent the delays associated with a transfer at *j* = Chicago, between a rail leg (= m_1) from Seattle (= *i*) to Chicago (= *j*) and a truck leg (= m_2) from Chicago (= *j*) to Gary (= *k*). This example shows that it is necessary to know the starting node *i*, in order to know the arrival time of the train in Chicago, and thus the transfer delay at Chicago (node *j*). Fixed transfer costs, such as bus fares and parking fees, are represented by $v_{ijk}^{m_1 m_2}(t)$ to describe the fixed cost associated with switching from mode m_1 to mode m_2 when traveling from node *i* through node *j* to node *k*, departing from node *j* at time *t*. Again, it is necessary to index this variable by nodes *i*, *j* and *k*, and modes m_1 and m_2, as transfer costs may differ depending on the modes, origins and destinations of the two legs. As with travel times and costs, mode transfer times and costs are also assigned to discretize time intervals, *t*.

The mode switching cost function, Equation (2), is a combination of the transfer time-related and fixed transfer costs, where the transfer time-related cost is assumed to be a linear function of the transfer time. The parameter, $\beta^{m_1 m_2}$, reflects the level of dislike that travelers tend to have for a certain transfer. For example, waiting inside a train station might be less onerous than waiting at an open bus stop, so a transfer to the train mode would have a lower value of $\beta^{m_1 m_2}$, and would thus incur a lower transfer cost.

$$\kappa_{ijk}^{m_1 m_2}(t) = \nu_{ijk}^{m_1 m_2}(t) + \beta^{m_1 m_2} \cdot \xi_{ijk}^{m_1 m_2}(t) \tag{2}$$

Travel and transfer cost functions in the network are illustrated in Figure 6-1. At origin nodes, parallel entry nodes *i'*, are introduced to capture possible transfer delays $\xi_{i'ij}^{m_1 m_2}(t)$, fixed costs $\nu_{i'ij}^{m_1 m_2}(t)$, and total costs $\kappa_{i'ij}^{m_1 m_2}(t)$, incurred when entering the network. For example, waiting for a bus and paying bus fare at the beginning of a trip would be captured by a transfer from the entry node *i'* to the origin node *i*. Further, node *i'* is connected to *i* by a link *i'i* on an imaginary trip start mode m_s. Let the set of all entry nodes be represented by *N'*. Similarly, an exit node *D"* is added to the destination node *D* to capture trip end costs such as parking delays and fees. Further, node *D* is connected to *D"* by a link *DD"* on an imaginary trip finish mode m_f. Let the set of all exit nodes be defined by *N"*. Figure 6-1 shows the relationship of entry and exit nodes and modes.

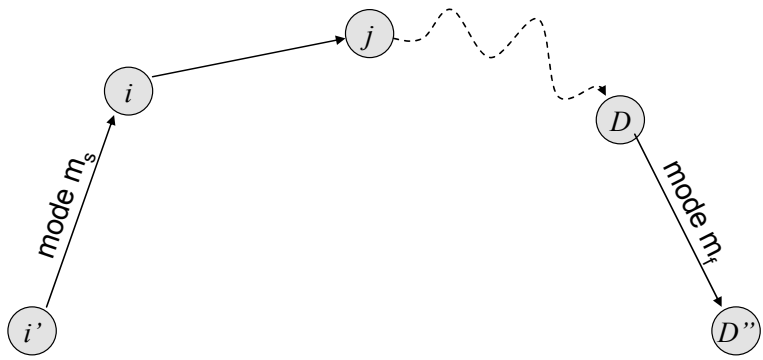


Figure 6-1. Entry and exit nodes and modes.

6.4 ALGORITHM AND PROPERTIES

The TDIMCP algorithm maintains cost and travel time labels for each node. The label, $\theta_{ij}^{m}(t)$, denotes the optimal path cost from node *j* to the destination exit node *D"* when arriving at node *j* from node *i* on mode *m* at time interval *t*. The label, $\gamma_{ij}^{m}(t)$, denotes the total travel time on this optimum path from node *j* to the destination exit node *D"* when departing from node *j* at time interval *t*. Figure 6-2 illustrates the significance of cost and time labels in the network.

The necessary and sufficient condition for a cost label, $\theta_{ij}^{m}(t)$, to be optimal is shown in Equation (3), with terminal conditions shown in Equation (4). These conditions are a straightforward extension of Bellman's principle of optimality (1957). Specifically, cost labels indicate the cost of the minimum cost path, but in this case indices maintain consistency of time across travel and transfer movements.

$$\begin{aligned}\theta_{ij}^{m_1}(t) \le \kappa_{ijk}^{m_1 m_2}(t) + \phi_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t)\right) \\ + \theta_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t) + \tau_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t)\right)\right) \\ \forall ij \in A, jk \in A, m_1 \in M, m_2 \in M, t \in T\end{aligned} \tag{3}$$

$$\theta_{iD}^{m_l}(t) = \kappa_{iDD''}^{m_l m_f}(t) \quad \forall i \in \left\{\Gamma^{-1}(D), D'\right\}, \forall m \in M, \forall t \in T \tag{4}$$

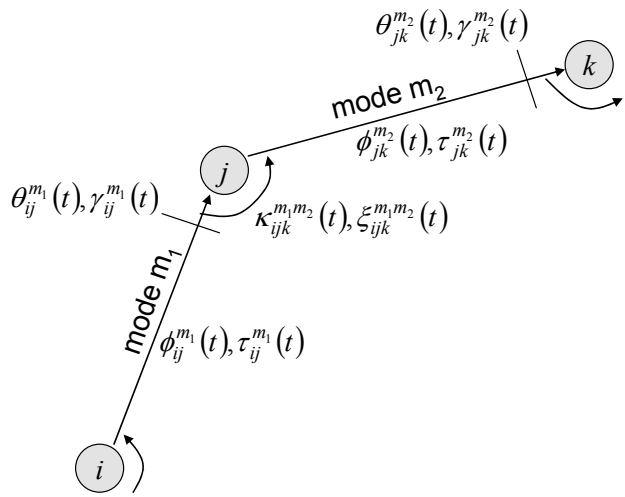


Figure 6-2. Cost and time labels along with link and transfer costs and times.

where $\Gamma(i)$ is the set of successor nodes to node i, $\Gamma^{-1}(i)$ is the set of predecessor nodes to node i, D is the destination node, D' and D'' are the entry and exit nodes at D, respectively, m_f is the exit mode between D and D'', and $N\backslash D''$ is the set of all nodes except exit node D''.

The steps of the TDIMCP algorithm are shown in Figure 6-3. The TDIMCP algorithm starts from the destination exit node and iteratively scans all nodes k for optimality. The scanning process consists of examining all predecessor nodes j to check whether extending the path backward from the current node k to a predecessor node j provides a lower cost path from those nodes j to the destination for all modes and time intervals. If such an extension does provide a better path for a predecessor node j for at least one mode, time interval and arc combination, node j is considered to have the potential to improve the paths to its predecessor nodes. As such, its cost and time labels are updated, along with corresponding successor node labels to indicate the next node in least-cost path. The node j is then marked as eligible to be scanned.

Proof of the algorithm's correctness is provided in Chang and Ziliaskopoulos (2005). Specifically, it is proven that at every stage of the computation, the cost labels of a node are either infinite or finite numbers. An infinite label means that no path exists from that node to the destination node for that time step. A finite label is an upper bound on the least-cost path from that node to the destination node at that time step. Next, it is proven that the algorithm terminates in a finite number of iterations, and that upon termination, the following relation holds for every cost label $\theta_{ij}^{m}(t)$:

$$\begin{aligned}\theta_{ij}^{m_1}(t) \le \kappa_{ijk}^{m_1 m_2}(t) + \phi_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t)\right) \\ + \theta_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t) + \tau_{jk}^{m_2}\left(t + \xi_{ijk}^{m_1 m_2}(t)\right)\right) \\ \forall ij \in A, jk \in A, m_1 \in M, m_2 \in M, t \in T\end{aligned} \tag{5}$$

In other words, upon termination of the TDIMCP algorithm, the optimality conditions is satisfied, where every cost label $\theta_{ij}^{m}(t)$ is either an infinite number, meaning that no path exists from node j to the destination node for this predecessor node, mode and time interval combination, or a finite number that represents the least-cost path from this predecessor node, mode and time interval combination, to the destination exit node D'' and exit mode m_f.

Step 1

Initialize the labels as follows:
$\gamma_{DD"}^{mf}(t) = 0, \forall t \in T$
$\theta_{DD"}^{mf}(t) = 0, \forall t \in T$
$\gamma_{ij}^{m}(t) = \infty, \ \forall j \in N \backslash \{D"\} \ \forall i \in \{\Gamma^{-1}(j), j'\}, \ t \in T$
$\theta_{ij}^{m}(t) = \infty, \ \forall j \in N \backslash \{D"\} \ \forall i \in \{\Gamma^{-1}(j), j'\}, \ t \in T$
Insert the destination node *D"* into the "Scan Eligible" (SE) list.

Step 2

If the SE list is empty, go to Step 4; otherwise, delete the first node *k* from the SE list and do the following:
For every node $j \in \Gamma^{-1}(k)$ do the following:
For every mode $m1 \in M$ do the following:
For every mode $m2 \in M$ do the following:
For every node $i \in \{\Gamma^{-1}(j), j'\}$ do the following:
For all time intervals $t \in T$ do the following:

If $$\theta_{ij}^{m1}(t) > \left\{\kappa_{ijk}^{m1,m2}(t) + \phi_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right) + \theta_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t) + \tau_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right)\right)\right\}$$

Then

Set $$\theta_{ij}^{m1}(t) = \left\{\kappa_{ijk}^{m1,m2}(t) + \phi_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right) + \theta_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t) + \tau_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right)\right)\right\}$$

Set $$\gamma_{ij}^{m1}(t) = \left\{\xi_{ijk}^{m1,m2}(t) + \tau_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right) + \gamma_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t) + \tau_{jk}^{m2}\left(t + \xi_{ijk}^{m1,m2}(t)\right)\right)\right\}$$

Mark node *j* as eligible to be scanned;
Otherwise,
Do nothing with this time interval and modal combination.
If node *j* has been marked as eligible to be scanned and is not currently in the SE list, insert it into the SE list.

Step 3.

Go to Step 2.

Step 4.

Terminate the algorithm.

Figure 6-3. Time Dependent Intermodal Minimum Cost Path Algorithm.

It is also important to note that if all link travel times are positive, the algorithm is guaranteed not fall into an infinite cycle, since with each cycle, time elapses and until the model time period eventually run out. While the algorithm is guaranteed not to fall into infinite cycles, unrealistic loops may

occur if cost parameters are not carefully defined. For example, if the cost parameter associated with walking time is less than the cost parameter associated with transit waiting time, then the algorithm will assume that a traveler would prefer to walk in circles until a bus arrives, rather than stand at the transfer node. In addition, unrealistic looping behavior may also occur if the cost of travel along a certain link decreases with time. For example, if the cost of traversing a link *ij* is 10 at time *t* and only 1 at time *t+x*, then the optimal alternative would be to wait at *i* until time *t+x* to traverse link *ij* at a cost of 1. However, since the algorithm does not explicitly allow for stationary waiting, the optimal path found would include a loop from node *i* that can be repeated until *x* amount of time elapses without exceeding a cost of 9 through the looping process. While it might be realistic that travel costs decrease with time, the looping behavior estimated by the algorithm is not realistic in general. Further research is necessary to develop an algorithm capable of limiting the number of transfers.

Further, as with any algorithm that relies on Bellman's principle of optimality. the TDIMCP algorithm maintains costs in an additive fashion, and thus considers only linear costs. This is reasonable for many travel costs; however, some travel costs, such as walking time, might be better modeled as non-linear functions of time. Discounted transit transfer fare structures are another common example of non-linear costs. Specifically, many transit agencies allow passengers to pay reduced fares for transfers on a single trip; however, the algorithm does not track transfers, and thus does not account for transfer fares beyond the first transfer. In addition, transfers themselves are often considered non-linear costs, as many travelers perceive additional transfers to be increasingly onerous.

The algorithm's running time is bounded by O ($|T|^2|H||X|^2|N|$), where the T is the number of time steps, H is the number of modal links in the network, X is the maximum number of predecessor modal links to each node, and N is the number of nodes in the network. As expected, computational tests on different sizes of networks revealed linear relationships between computational time and number of modal links, as well as with number of nodes. However, a sub-quadratic relationship of computational time with number of time steps, although the worst case bound shows it to be quadratic. This occurs because of the property of dynamic networks that the best paths between a given origin-destination pair are often repeated for many time steps. Further details on the algorithm's running time can be found in Chang and Ziliaskopoulos (2005). An example set of fifty numerical tests from that article shows that the TDIMCP algorithm may be solved for one destination of a network with 600 nodes, 1,400 modal links and 200 time steps, in an average of 30.3 seconds. This

would be a realistic problem size, and could be solved in real-time for rolling horizon periods lasting even just a few minutes.

6.5 EXTENSION OF THE TDIMCP ALGORITHM TO MINIMUM RISK HAZMAT ROUTING

The problem is stated as follows: A hazmat shipment is to be carried through a multimodal transportation network with transshipment stations and time-varying transport risk and transport time parameters; the objective is to compute the minimum risk path between origin and destination by effectively combining the available modes, while accounting for travel costs and risks. In formulating the problem, the following need to be considered: i) every combination of link and mode is associated with a travel risk and cost and ii) every combination of hazmat transshipment and mode sequence is associated with a transfer risk and cost.

The cost of risk may be captured in a time-dependent cost parameter. According to this approach, risks and resulting costs can be varied by time and for each link and transfer, such that conditions and costs in the hazard area can be varied by time through appropriate definition of the time-dependent input parameters. In addition, transport cost may be effectively represented by transport time, since in commercial activities cost is closely connected to required time.

In setting the necessary framework for the Hazmat approach the proposed TDIMCP algorithm is extended. The arc and transfer cost functions are now formulated as the weighed average of corresponding risk and time cost items. In arc cost function fixed travel cost $\mu_{ij}^{m}(t)$ is replaced by time-varying travel risk cost item $\rho_{ij}^{m}(t)$ and parameter α^{m} is considered the weigh factor of travel time cost item $\tau_{ij}^{m}(t)$. Thus, $\rho_{ij}^{m}(t)$ is the travel risk associated with traveling on arc *(i,j)* using mode *m*, when departing from node *i* at time *t*. Similarly, in mode switching cost function fixed transfer cost $\nu_{ijk}^{m_1 m_2}(t)$ is replaced by time-varying transfer risk cost item $\rho_{ijk}^{m_1 m_2}(t)$ and again parameter $\beta^{m_1 m_2}$ is considered the weigh factor of transfer time cost item. Thus, $\rho_{ijk}^{m_1 m_2}(t)$ is the transfer risk associated with switching from mode m_1 to mode m_2 when traveling from node *i* through node *j* to node *k*, departing from node *j* at time *t*. It is obvious that equations (1) and (2) maintain exactly the same structure as shown in equations (6) and (7). Consequently, equations (3) and (4) also maintain their structure.

$$\Phi_{ij}^{m}(t) = \rho_{ij}^{m}(t) + \alpha^{m} \cdot \tau_{ij}^{m}(t) \tag{6}$$

$$\mathrm{K}_{ijk}^{m_1 m_2}(t) = \rho_{ijk}^{m_1 m_2}(t) + \beta^{m_1 m_2} \cdot \xi_{ijk}^{m_1 m_2}(t) \tag{7}$$

In addition, the use of artificial entry and exit nodes on artificial entry and exit modes is still needed, in order to represent risks and delays that occur when hazmat shipments enter or exit the transportation network, such as the time needed and risk taken during the initial loading or the final unloading of the shipment respectively.

The proposed formulation for addressing the Intermodal Time-dependent Hazmat Routing problem uses a weighed average approach to form the objective function under minimization which as mentioned previously is a commonly applied methodology in addressing similar problems. Since this approach is primarily concerned in addressing risk, by taking into account that the more the parameters that are combined in a multi-objective cost function the less the impact of each one of them in the computed result, the consideration of only the risk parameter combined with the weighed time parameter is justified. Furthermore, the incorporation into the cost function of travel and transfer times as a measure of cost results in computing routes that don't unreasonably increase transportation costs and that are likely to be followed by the carriers to the satisfaction of the fleet manager.

It is also straightforward that the above formulation doesn't alter in any way the structure of the Intermodal Time-dependent Minimum Cost problem and the TDIMCP algorithm can be applied to compute optimum routes. Moreover the correctness and optimality properties of the TDIMCP algorithm remain unaffected.

One important point that has to be mentioned is that risk needs quantification prior to calculate the risk parameter to use in computations. The quantification is a very critical process that plays an important role in effective problem solving. It is necessary to consider all aspects of risk and apply the proper methodology. Probabilistic Risk Assessment models are the most widely used because they are easy to apply. Risk is expressed as the product of two factors: (i) probability of occurrence P of an incident and (ii) consequences C of that accident: $r = PC$.

After risk quantification is completed the produced risk parameters can be treated as any other cost parameters and can be easily applied in forming the cost function. In that sense they can be added across routes to provide an indication of total risk throughout each route.

6.6 NUMERICAL TESTS OF HAZMAT ROUTING PROBLEM

To demonstrate the behavior of the algorithm on a hazmat problem, it was applied to a test network for which we selectively changed the transport risk of specific links. The computed routes were then examined to investigate how the algorithm adapts to the altered input data. Consider the transportation network of Figure 6-4. There are 5 nodes with transshipment capabilities, 15 link-mode elements, 3 alternative transport modes, 3 discrete time intervals, origin node 0 and destination node 4. To apply the algorithm first we add to every node an artificial entry node and an artificial exit node, connected to the node in question on artificial modes S and T respectively. Although the algorithm calculates optimum paths from all nodes and for all time intervals to the destination node, we focus our analysis on the path from origin node 5 for all possible departure times.

Applying the algorithm to the initial network we obtain the results shown in Table 6-1. The computed sequence of nodes and modes is the same for all three departure times. From node 5 to node 0 on mode S, from node 0 to node 3 on mode A, from node 3 to node 4 on mode B and from node 4 to node 14 on mode T. The total path cost is considerably lower for the first departure time. The optimum path includes a mode change in intermediate node 3 from A to B.

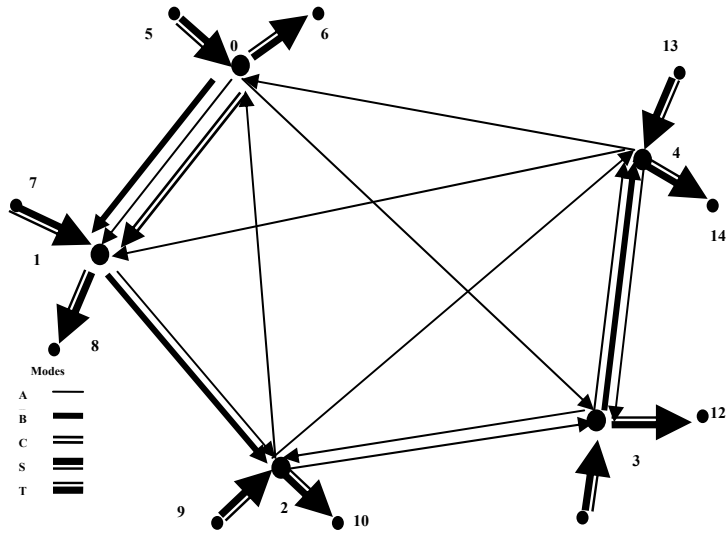


Figure 6-4. Initial network.

Table 6-1. Initial Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→3→4→14	S→A→B→T	18.27
t1	5→0→3→4→14	S→A→B→T	26.15
t2	5→0→3→4→14	S→A→B→T	26.65

Next, six tests were performed under different scenarios, as illustrated in Figure 6-5. The first test scenario examines the impact of adding a permanent high-risk facility in the network. Specifically, suppose that close to link (0,3) a hospital center is established. We set travel cost of link (0,3) to a considerably higher value for all three time intervals. As a result, the algorithm excludes that link from the optimum path, as shown in Table 6-2.

The second scenario examines the impact of adding a time-varying risk facility. Specifically, suppose on link (0,3) an all-day school is established instead of the hospital. As a result this link has a high travel risk during the day, when it is full of students. Increasing travel risk for the first two time periods (day) and leaving it unchanged for the third one (night) the algorithm adapts to the new input data. Table 6-3 shows that only for the first departure time the algorithm excludes link (0,3), since in that case the school is open during the shipment transportation. For both the other two time intervals link (0,3) is traveled while the school is closed and is part of the optimum route.

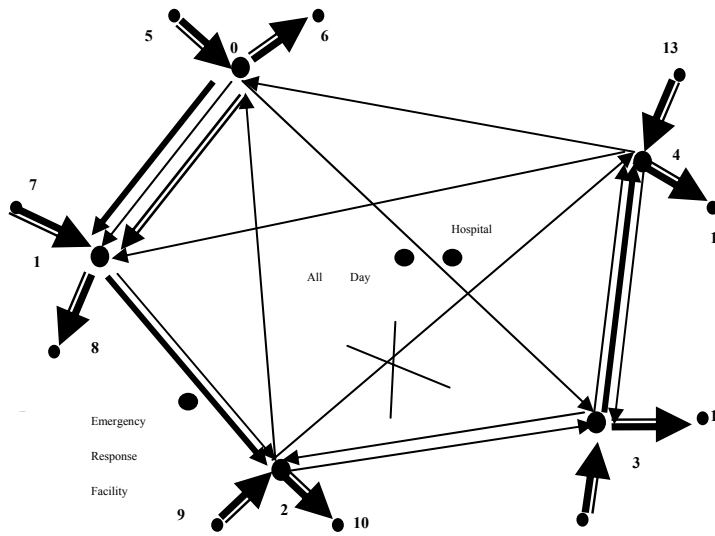


Figure 6-5. Scenarios 1 through 6.

Table 6-2. Scenario 1 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→4→14	S→C→A→A→T	35.75
t1	5→0→1→2→4→14	S→C→A→A→T	37.24
t2	5→0→1→2→4→14	S→B→B→A→T	36.74

Table 6-3. Scenario 2 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→4→14	S→C→A→A→T	35.75
t1	5→0→3→4→14	S→A→B→T	26.15
t2	5→0→3→4→14	S→A→B→T	26.65

Next, the impact of adding a risk reduction facility is examined. Specifically, we built on scenario 1 by supposing that apart from establishing a hospital on link (0,3), an emergency facility is located for link (1,2) on mode B, such that travel risk on that mode is significantly reduced compared travel on mode A. Again, the algorithm perceives this change and reflects the suitability of mode B in the computed path, as shown in Table 6-4.

In Scenario 4, the effect of a link break-down is explored. Specifically, suppose that link (2,4) is closed, for example because of a catastrophic earthquake. Table 6-5 shows that in the optimal solution under this scenario, link (2,4) on mode A is replaced with sub-path (2,3) on mode A and (3,4) on mode B. Total travel cost is significantly increased, since the algorithm has to choose a less satisfying solution compared to the previous case.

The effect of traffic deterioration was examined in Scenario 5. Specifically, suppose that statistical analysis of the number of accidents on mode B on link (3,4) indicates a significant increase in accident probability. Increasing travel cost on link (3,4) for the more risky mode B and reapplying the algorithm, the solution shown in Table 6-6 is obtained. Once again the algorithm takes into account the new network characteristics and replaces mode B with mode A in the optimal path. As expected, total cost is increased.

Table 6-4. Scenario 3 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→4→14	S→C→B→A→T	27.67
t1	5→0→1→2→4→14	S→A→B→A→T	26.57
t2	5→0→1→2→4→14	S→B→B→A→T	24.74

Table 6-5. Scenario 4 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→3→4→14	S→C→B→A→B→T	38.37
t1	5→0→1→2→3→4→14	S→A→B→A→B→T	37.27
t2	5→0→1→2→3→4→14	S→B→B→A→B→T	35.45

Table 6-6. Scenario 5 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→3→4→14	S→C→B→A→A→T	44.82
t1	5→0→1→2→3→4→14	S→A→B→A→A→T	43.72
t2	5→0→1→2→3→4→14	S→B→B→A→A→T	41.89

Table 6-7. Scenario 6 Solution.

Departure Time	Path	Mode Sequence	Total Cost
t0	5→0→1→2→3→4→14	S→C→B→A→A→T	37.57
t1	5→0→1→2→3→4→14	S→C→B→A→A→T	38.88
t2	5→0→1→2→3→4→14	S→C→B→A→A→T	38.88

In Scenario 6 the effect of improved transshipment equipment is examined. Notice that in scenario 5 the algorithm selects a different mode for link (0,1) based on departure time step. Suppose that transport mode C is updated and equipped with technologically advanced transshipment capabilities. Consequently, loading and unloading from C to every other mode becomes a safer and faster procedure. Reducing transfer costs from mode S to C at node 0 and from mode C to B and A at node 1 we get the results shown in Table 6-7. The table shows that mode C is selected for link (0,1) for all departure times, and total transport cost is significantly reduced compared to the previous case.

6.7 CONCLUDING REMARKS

This paper described a time-dependent intermodal minimum cost path (TDIMCP) algorithm that extends traditional shortest path algorithms by accounting for intermodal transfer delays and costs on a time-dependent basis. The algorithm's correctness is discussed, along with cycling properties. For example, while paths with infinite cycles are impossible on networks with positive travel and transfer times, paths with unrealistic loops may occur as a result either of the combination of parameters in the generalized cost function, or of link costs that decrease with time. Next, the presented algorithm was used to address the problem of optimum multimodal/intermodal time-dependent hazmat routing in terms of minimizing a weighted average of transport risk and cost on both transfer and travel phases. The algorithm's effectiveness is demonstrated for different scenarios on a test network.

Future research in this direction includes testing of the proposed algorithm on realistic networks and problems, to observe the behavior of the results. Further, development of algorithms that calculate routes with limited transfers may be explored.

REFERENCES

Abdelghany, K.F. and H.S. Mahmassani. 2001. Dynamic trip assignment-simulation model for intermodal transportation networks. *Transportation Research Record* **1771**: 52-60.

Ahuja, R.K., J.B. Orlin, S. Pallottino, and M. Scutella. 2002. Minimum time and minimum cost path problems in street networks with traffic lights. *Transportation Science* **36**: 326-336.

Alp, E. 1995. Risk-Based Transportation Planning Practice: Overall Methodology and a Case Example. *INFOR* Feb 1995.

Battista, M.G., M. Lucertini and B. Simeone. 1996. Path composition and multiple choice in a bimodal transportation network. In *World Transport Research: Proceedings of the 7th World Conference on Transport Research, Volume 2*. New York: Pergamon.

Bellman, R. 1957. *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Brainard, J., A. Lovett and J. Parfitt. 1996. Assessing Hazardous Wastes Transport Risk using a GIS. *International Journal of Geographic Information Systems* **10**: 831-849.

Chabini, I. 1998. Discrete dynamic shortest path problems in transportation applications: complexity and algorithms with optimal run time. *Transportation research record*, **1645**: 170-175.

Chang, E. and A.K. Ziliaskopoulos. 2005. A Time-Dependent Intermodal Minimum Cost Path Algorithm. Submitted to *Transportation Research, Part B*, June 2005.

Crainic, T.G. and J.M. Rousseau. 1986. Multicommodity, multimode freight transportation: a general modeling and algorithmic framework for the service network design problem. *Transportation Research, Part B* **20** (3): 225-242.

Dial, R.B. 1967. Transit pathfinder algorithm. *Highway Research Record* **205**: 67-85.

Federal Motor Carrier Safety Administration. 2005. SafeStat webpage available online at http://ai.volpe.dot.gov/SafeStat/SafeStatMain.asp?PageN=result2&link. Date last accessed: November 26, 2005.

Frank, W.C., J.-C. Thill and R. Batta. 2000. Spatial Decision Support System for Hazardous Material Truck Routing. *Transportation Research, Part C* **8** (1/6): 337-359.

Huang, B. and P. Fery. 2005. Aiding Route Decisions for Hazardous Material Transportation. *84th Annual Meeting of the Transportation Research Board, 2005*. Pre-print CD ROM.

Jourquine, B. and M. Beuthe. 1996. Transportation policy analysis with a geographic information system: the virtual network of freight transportation in Europe. *Transportation Research, Part C* **4** (6): 359-371.

Kaufman, D.E. and R.L. Smith. 1993. Fastest Paths in Time-Dependent Networks for Intelligent Vehicle Highway Systems Applications. *IVHS Journal* **1** (1): 1-11.

List, G.F., P.B. Mirchandani, M. Turnquist and K.G. Zografos. 1991. Modeling and Analysis of Hazardous Materials Transportation: Risk Analysis, Routing/Scheduling and Facility Location. *Transportation Science* **25** (2): 100-114.

Lozano, Angelica and Giovanni Storchi. 2001. Shortest viable path in multimodal networks. *Transportation Research, Part A* **35** (3): 225-241.

Lozano, Angelica and Giovanni Storchi. 2002. Shortest viable hyperpath in multimodal networks. *Transportation Research, Part B* **36** (10): 853-874.

McCord, M.R. and A.Y-C. Leu. 1995. Sensitivity of Optimal Hazmat Routes to Limited Preference Specification. *Information Systems and Operational Research* **33** (2): 68-83.

Nguyen, S., E. Morello and S. Pallotino. 1988. Discrete time dynamic estimation model for passenger origin/destination matrices on transit networks. *Transportation Research, Part B* **22** (4): 251-260.

Nguyen, S., S. Pallotino and F. Malucelli. 2001. A modeling framework for the passenger assignment on a transport network with timetables. *Transportation Science* **35** (3): 238-49.

Pallotino, Stefano and Maria Grazia Scutella. 1998. Shortest path algorithms in transportation models: Classical and innovative aspects. *Equilibrium and Advanced Transportation Modelling*, Patrice Marcotte and Sang Nguyen eds. Boston, MA: Kluwer Academic Publishers.

Sherali, Hanif D., Antoine G. Hobeika and Sasikul Kangwalklai. 2003. *Transportation Science* **37** (3): 278-293.

Sivakumar, R.A., R. Batta and M.H. Karwan. 1995. "A Multiple Route Conditional Risk Model for Transporting Hazardous Materials". *Information Systems and Operational Research* **33** (1): 20-33.

Spiess, H. and M. Florian. 1989. Optimal strategies: A new assignment model for transit networks. *Transportation Research, Part B* **23** (2): 83-102.

Ziliaskopoulos A.K. and W. Wardell. 2000. An Intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. *European Journal of Operational Research* **125**: 486-502.

Ziliaskopoulos, A.K. and H.S. Mahmassani. 1993. Time-dependent, shortest path algorithm for real-time intelligent vehicle highway systems applications. *Transportation Research Record* **1408**: 94-100.

Zografos K.G. and K.N. Androutsopoulos. 2004. A heuristic algorithm for solving hazardous material distribution problems. *European Journal of Operational Research* **152** (2): 507-519.

Chapter 7

REAL-TIME EMERGENCY RESPONSE FLEET DEPLOYMENT: CONCEPTS, SYSTEMS, SIMULATION & CASE STUDIES

Ali Haghani and Saini Yang
Department of Civil and Environmental Engineering, University of Maryland, College Park, Maryland, 20742 3021

Abstract: Dynamic response to emergencies requires real time information from transportation agencies, public safety agencies and hospitals as well as the many essential operational components. In emergency response operations, good vehicle dispatching strategies can result in more efficient service by reducing vehicles' travel times and system preparation time and the coordination between these components directly influences the effectiveness of activities involved in emergency response. In this chapter, an integrated emergency response fleet deployment system is proposed which embeds an optimization approach to assist the dispatch center operators in assigning emergency vehicles to emergency calls, while having the capability to look ahead for future demands. The mathematical model deals with the real time vehicle dispatching problem while accounting for the service requirements and coverage concerns for future demand by relocating and diverting the on-route vehicles and remaining vehicles among stations. A rolling-horizon approach is adopted in the model to reduce the relocation sites in order to save computation time. A simulation program is developed to validate the model and to compare various dispatching strategies.

Keywords: Emergency Vehicle, Response Time, Deployment, Real-Time, Optimization, Simulation

7.1 INTRODUCTION

The emergency response process includes a sequence of activities, such as alert and warning, damage assessment, emergency operation, evacuation, and rescue. This chapter is focused on the deployment of emergency

response vehicle fleet. The emergency response process in real world operation can be summarized in Figure 7-1.

In this process, the duration of an emergency event is an important index of event's negative impacts, especially for severe ones. The duration of an emergency event can be divided into 4 phases: detection time, preparation time, travel time and treatment time, as shown in Figure 7-2. Response time is the most important factor. Response time can be defined as the duration from the time an emergency call arrives at the station to the time an emergency vehicle arrives at the scene. This is the sum of preparation time and travel time. The National Fire Protection Agency (NFPA) has developed a series of codes that serve as guidelines in real world operations. Table 7-1 outlines the NFPA's standards for response times. According to this standard at least 90% of emergencies should be dealt by the first arriving responder unit within 5 minutes. When the ambulance with advanced life support (ALS) is needed, 90% of the emergencies should be reached by an ALS within 9 minutes.

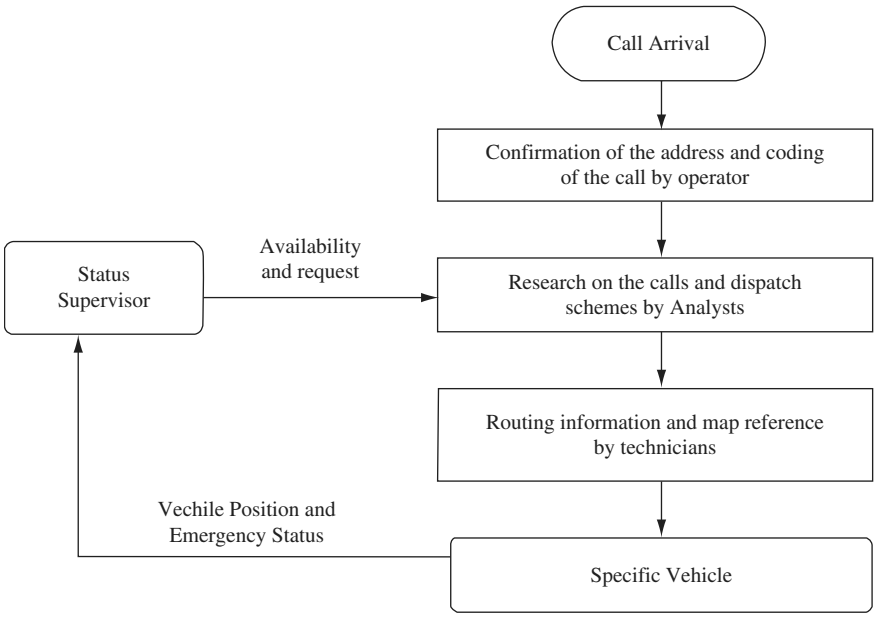


Figure 7-1. Emergency response process.

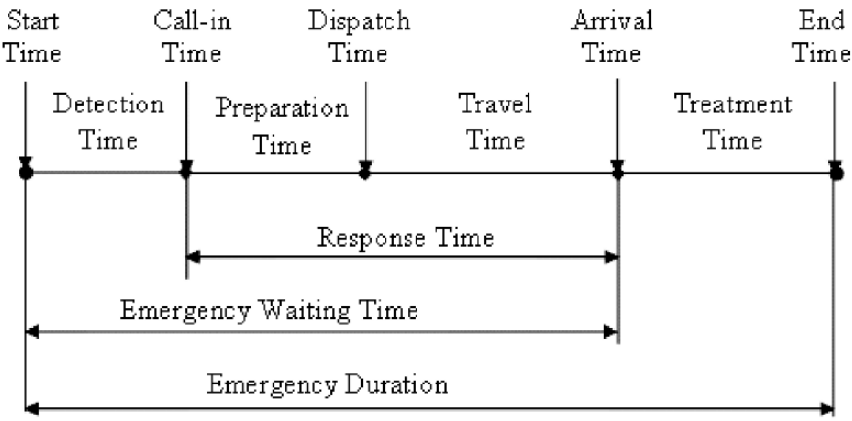


Figure 7-2. Emergency response time.

Table 7-1. NFPA guidelines of response times.

Fire Suppression Incident		**Emergency Medical Incident**	
First Arriving Engine Company Total Response Time	Full First Alarm Assignment Total Response Time	First Responder Unit Total Response Time	Advanced Life Support (ALS) Unit Total Response Time
5 minutes	9 minutes	5 minutes	9 minutes
90% Achievement Rate	90% Achievement Rate	90% Achievement Rate	90% Achievement Rate

7.2 AN INTEGRATED EMERGENCY VEHICLE FLEET MANAGEMENT SYSTEM

From the early 70's, researchers have known that better allocation of emergency facilities can help reduce the response times and improve the service levels. Chaiken and Larson (1998) provided a survey of methods for allocating urban emergency units, which discussed four aspects of allocation: (1) determining the number of units to have on duty, (2) locating the units and facilities, (3) designing their response areas or patrol areas and (4) planning preventive-patrol patterns for police cars. Later, various allocation models were developed and different dispatching strategies were applied in simulation models and real operation, such as First Come First serve (FCFS), Nearest Origin (NO) or Highest Priority First serve (HPFS). Recently, the interoperability and exchange of data across all public safety (e.g. fire and rescue personnel, paramedics and police) and transportation

agencies (e.g. state or county transportation department, local transportation commission) is becoming common practice. This provides an extraordinary opportunity to improve coordination of activities of these agencies that play key roles in emergency response service. Cooperation and coordination among these agencies for improving emergency response service has never been explored in previous research. The current paradigm for interoperability and data and information exchange among agencies has created a tremendous opportunity for a major research contribution by developing a more integrated emergency response system.

The potential benefits of an integrated Emergency Vehicle Fleet Management System include:

- Reduce emergency response time: The loss of life and assets mainly depends on the planning phase and response operation. Proper planning of location and fleet can help speed up the emergency response. Fast and accurate response to an emergency can save precious time and improve the efficiency of the system.
- Reduce operation cost: With better planning and operation guidance, the same crew and fleet will be able to handle the responsibility more efficiently and with higher performance levels.
- Improve information utilization: The efficiency of the response system is heavily based on the information. Real-time traffic volume on streets has great influence on travel speed. The fleet surveillance system can track the status and location of vehicles and help in developing vehicle assignment plans for new emergency calls. A GIS maps can precisely locate the emergency site, and a GIS database can help establish the magnitude of life, property and effort involved, determining the risk zones based on land use data, and building and activity in tune with the National Building Code guidelines.
- Evaluate the efficiency and the effectiveness of services: An integrated fleet management can help record all of the necessary information needed for evaluation, so as to improve the performance in the future. Also the system will be able to record the causes and effects of the emergency so that more effective mitigation can be applied.

Based on the operation routine, we propose a system which is composed of 4 internal modules: travel time predictor, shortest path calculator, dispatching optimizer and simulator, and three external modules: Traffic Data Receiver, Emergency Data Receiver and evaluator and other optimizers. Figure 7-3 shows the structure of the system. The Dispatching Optimizer module is the mathematical models that optimize system

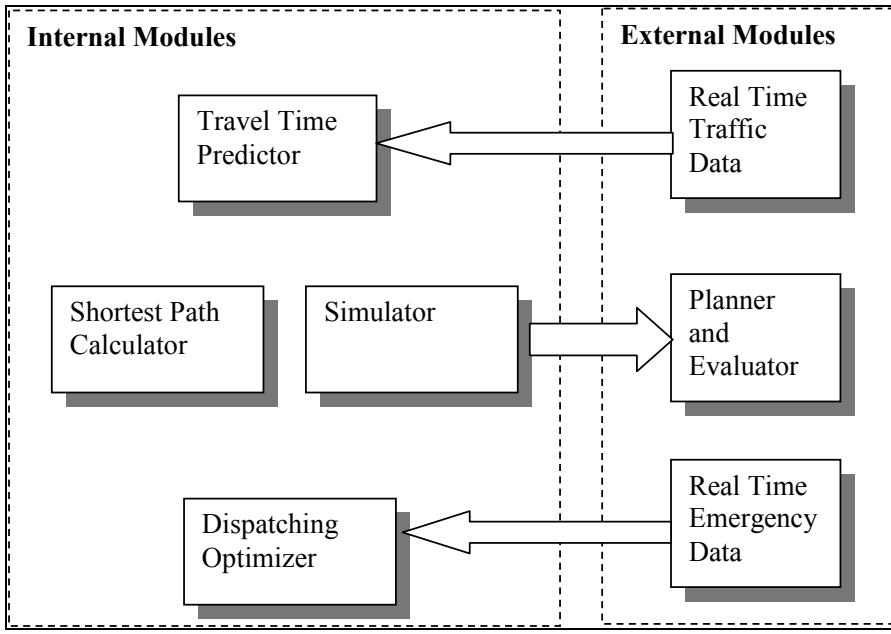


Figure7-3. The structure of an emergency response system.

operations and assist the dispatch personnel in their daily emergency response operations. In this module, multiple vehicle types and multiple emergency types are taken into consideration. The external modules can be used by decision makers for planning purposes and system evaluation. Each module in this system can be a specific topic for further research.

7.3 PROBLEM STATEMENT

In most major cities, emergency response services are provided by a fleet of emergency vehicles, which is mainly composed of ambulances, fire trucks and police cars. Certain fixed dispatching strategies are used in real operation. Due to the limited number of emergency vehicles, whenever a vehicle is dispatched to a call, it may leave a significant fraction of the population without proper coverage, e.g. the future calls will experience long waiting times greater than the pre-set limit. Therefore it is necessary to introduce more flexible dispatching strategies, such as to allow dispatched emergency vehicles on route switch to a new emergency call that is more sever (*diversion*) or to relocate the idle vehicles (*relocation*) in order to maintain a proper coverage for future demands, as well as allow vehicles to change the route to destinations (*rerouting*) based on the real traffic information.

Figure 7-4 illustrates how the online information can be utilized in diversion and rerouting to improve the operations. In a response area with four

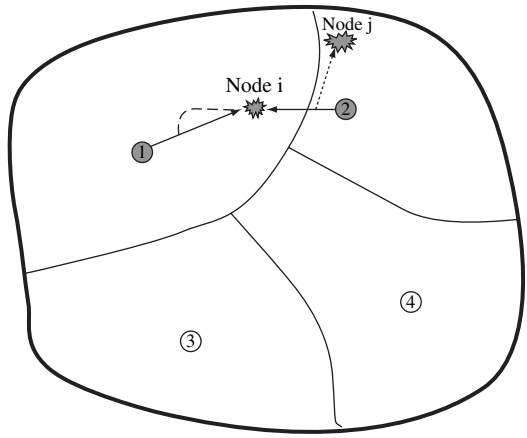


Figure 7-4. A simple example of dispatching and routing problem.

zones, there is one emergency station in each zone with one vehicle. At time t, an emergency occurs on node i. The vehicle in station 2 is assigned to deal with it. At time t+1, another emergency at node j occurs. Since the vehicle in station 2 has already been assigned, the closest vehicle to the new emergency is the vehicle in station 1. If we reassign and reroute the vehicle from station 2 to the new emergency and assign the vehicle from station 1 to the earlier one, we are able to avoid the long travel time from station 1 to node j. This *diversion* makes the response time for both emergencies shorter. When real-time traffic information is available and congestion on the pre-select routes is detected, *rerouting* of vehicles can help avoid the possible delays. It will be very helpful if we can develop an online model that can handle the real-world operation requirements as well as optimizing the process.

When emergency response vehicles arrive at emergency sites and are busy responding to incidents, gaps in the service area will be created which cannot be effectively covered. This means new emergency calls from these areas may experience long delays in response. *Relocation* can be applied under this situation. In Figure 7-5, the two circles show the contours around the stations that can be reached within t_{max} minutes, where t_{max} is the pre-set coverage time. If vehicles from stations 1 and 2 are assigned to emergency calls at nodes i and j, the blank areas in zones 1 and 2 are without coverage (within t_{max} minutes), and some points such as node m may experience very long travel times. *Relocation* can be applied to avoid this situation. By relocating vehicle 2 from zone 3 to zone 1, both the size of the uncovered area and the longest travel time (from stations 1 and 4 to node n) will decrease as a result as shown in Figure 7-6.

These examples show that proper dispatching strategies and efficient optimization may improve the service provided to general population.

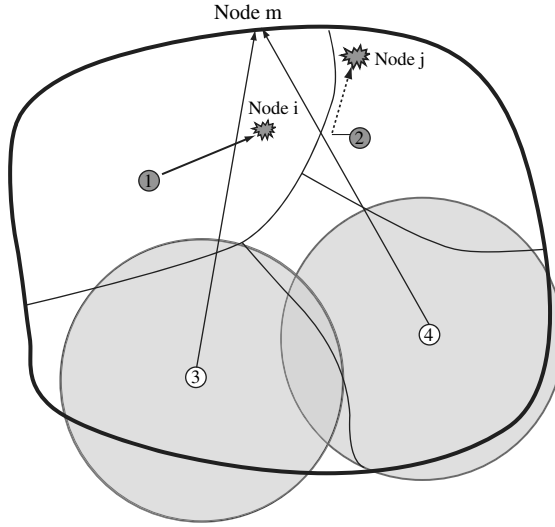


Figure 7-5. A simple example of coverage problem (a).

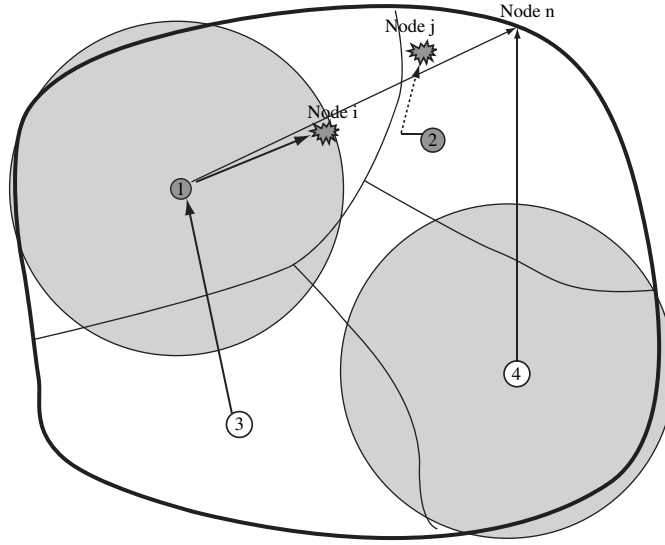


Figure 7-6. A simple example of coverage problem (b).

7.4 LITERATURE REVIEW

Most of the literature in emergency response is focused on Emergency Medical Service (EMS) systems, and deals with the study of location, fleet size, and operational performance. These have been important subjects for operations researchers and management scientists. Similar research also includes many other public services such as emergency repair and traffic incident management.

One of the key problems in an emergency response system is the Vehicle Dispatching Problem. When emergency calls arrive at the emergency response system, the most important responsibility of the dispatcher is to decide the number and types of required vehicles, and to dispatch these vehicles to emergency locations. Besides the tests of some static dispatching strategies (FCFS, NO, HPFS), limited literature exists that relates to this specific problem. (Haghani *et al*., 2003) proposed a mathematical model that deals with the time-dependent EMS dispatching and re-routing. In their model, the vehicle dispatching problem is formulated as an integer model with an objective function that minimizes the total travel time in the system. A time dependent shortest path algorithm is used in the calculation of travel times. One type of vehicle is considered in the model and a simplifying assumption that each emergency call needs one and only one vehicle is made.

When taken the priority of emergencies and vehicle types into consideration, the vehicle dispatching problem is similar to a Generalized Assignment Problem (GAP). The Generalized Assignment Problem deals with the question of how to assign *n* tasks to *m* machines in the best possible way. The Generalized Assignment Problem (GAP) is a well-known, NP-complete combinatorial optimization problem (Fisher, 1985). Early work on the generalized assignment problem has concentrated on exact solutions to the problem using enumerative schemes with the bounding methods (Fisher *et al.*, 1986; Martello *et al*., 1984; Ross *et al*., 1975). However, these types of methods usually are computationally expensive. Since GAP is a NP-complete problem, it is unlikely to find any efficient method for finding an exact solution. Later, more heuristics were developed. Catrysse *et al.*, 1993 surveyed the heuristics for the GAP. Quite a few of them are based on the linear relaxation of the General Assignment Problem (Brown *et al.*, 1985; Lorena *et al.*, 1996; Narciso *et al.*, 1999; Nulty *et al.*, 1988; Trick, 1992), and genetic algorithm heuristics (Chu *et al.*, 1997; Lorena *et al*., 2002).

A large portion of the existing literature is focused on the Emergency Facility Siting Problem and the most common approaches are to use mathematical programming and queuing methods. The allocation of emergency vehicles is an important part in this research. Hakimi, 1964, was

the earliest researcher who considered the siting problems. Comprehensive review and perspective on these models are provided by surveys (Marinov *et al.*, 1995; Revelle, 1989; Revelle, 1997; Schilling *et al.*, 1993). The models can be grouped into 3 categories:

- Basic deterministic covering models (Church *et al.*, 1974; Toregas *et al.*, 1971; Toregas *et al.*, 1974), which seek to position the least number of facilities needed to cover all points of demand within S distance or time units;
- Deterministic models (Daskin, 1983; Hogan *et al.*, 1986; Martello *et al.*, 1984; Schilling *et al.*, 1979), which consider the value of additional covering servers; and
- Probabilistic models (Ball *et al.*, 1993; Larson, 1974; Larson, 1975; ReVelle *et al.*, 1988), which allow randomness in server availability.

When taking the service coverage concern into consideration in the Vehicle Dispatch Problem, vehicle relocation is needed for better coverage of the service area and avoiding possibly extremely long travel times. Instead of seeking a single solution to a static or probabilistic model for Emergency Facility Location Problem, a new problem is introduced to dynamically relocate vehicles in real-time as vehicles are dispatched to calls, namely, Emergency Vehicle Relocation Problem. An early dynamic model was proposed (Kolesar *et al.*, 1975a) for the relocation of fire companies. Each relocation amounts to solving a static model subject to side constraints on vehicle moves. Gendreau *et al.* (2001) developed a dynamic ambulance relocation model which can be applied in real-time through the use of tabu search algorithm and parallel-computing. Later, an a priori methodology and the appropriate solution were proposed (Gendreau *et al.*, 2001) for the dynamic relocation problem, in which several solutions are pre-computed in anticipation of future events. One important issue in dynamic dispatching is the computation time. A new solution will be needed within a short time period when a call arrives or when the traffic information is updated. This can be time consuming or even infeasible when calls arrive with high frequency throughout the day. Brotcorne *et al.* (2003) provide a survey of the Emergency Vehicle Relocation Problem.

Compared to optimization models, simulation models enable us not only to find a good solution to some decision problems, but also to observe a system under different sets of assumptions. They also provide the possibility to test new operational strategies such as different ambulance locations or dispatching rules. In the past 30 years, simulation models have been developed and commonly used to evaluate the performance of emergency response systems.

Early simulation models (Carter *et al.*, 1970; Fitzsimmons, 1973; Ignall *et al.*, 1978; Savas, 1969) are based on the First-In-First-Out system. No queue is considered in the simulation and these simplifications greatly reduced the precision of the models. The link between simulation and analytical models has been further analyzed and evaluated by Shantikumar *et al.* (1983) in which they suggested the use of a hybrid approach that embedded analytical models in a simulation procedure. Goldberg *et al.* (1990) developed a multi-server queuing system on a First-In-First-Out basis without considering priority scheduling of calls. Later, Goldberg *et al.* (1991a) and Goldberg *et al.* (1991b) extended the previous work by allowing stochastic travel times, unequal vehicle utilizations, various call types, and service times that depend on call location. Response time was selected as the performance measure in Zografos *et al.* (1992) and Zografos *et al.* (1994), where two dispatching policies (First-In-First-Out and Nearest Origin) are studied.

Because of the real time feature of the system, shortest path travel time plays an essential role as the base criterion for on-line vehicle dispatch and routing. Since the travel time on the links is time dependent, to select the shortest travel route for each possible origin-destination pair, two issues need to be addressed: (1) shortest path algorithms, and (2) short term travel time prediction models.

When travel time is relatively stable, the travel time by static shortest path algorithms may provide quality solution (e.g. Dijkstra, 1959). Hall (1986) proved that static shortest path algorithms are not applicable to the problem with fluctuating traffic speed. To take advantage of real-time traffic information, it is necessary to use a more sophisticated shortest path algorithm such as a Dynamic or a Stochastic Shortest Path Algorithm (Chabini, 1998; Cooke *et al.*, 1966; Ziliaskopoulos *et al*., 1993).

Since we are more interested in the travel time that the drivers *will* encounter, the precision of travel time prediction results determine the reliability of dispatching and routing schemes. Besides historical data-based algorithm (Hoffman *et al*., 1988; Kaysi *et al.*, 1993; Stephanedes, 1981), time-series analysis technique is the most discussed travel time prediction method (Fitzimmons, 1973; Cragg *et al.*, 1995; Eldor, 1977; Gafarian *et al.*, 1977; Nahi, 1973, Nicholson *et al.*, 1974). Many simulation models [METANET, SIMRES (Simulation of the Regulation of a Reservoir), STM (Statistical Traffic Model) and DYNASMART] have been developed for travel time prediction. Unfortunately, they cannot support the online application in short term travel time prediction. Recently, prediction models based on Artificial Neural Networks are becoming widely used in short term prediction as well (Chang, 1999; Huisken, 2003; Smith *et al*., 1995).

7.5 SIMULATION

To test the system we proposed, a simulation model is developed and the conceptual framework is shown in Figure 7-7. In this simulation model, the following assumptions are made:

- The real street network can be abstracted as a graph with n nodes and m directed arcs and the emergencies are assumed to happen at nodes only. This assumption is reasonable when the segment of street network is detailed enough.

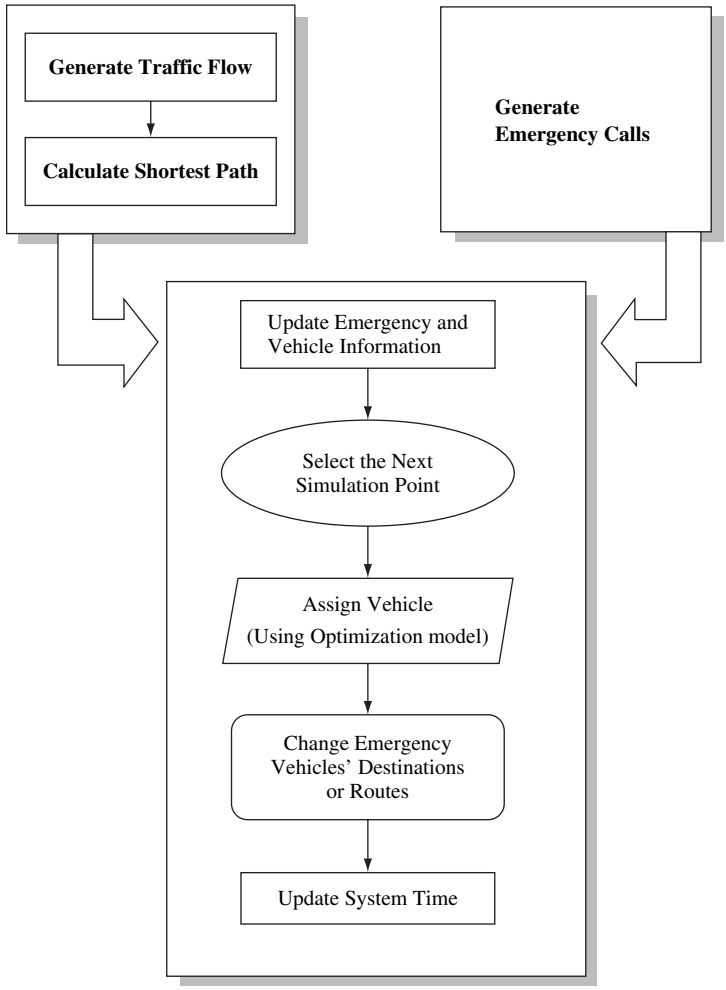


Figure 7-7. Conceptual simulation flow chart.

- *K* types of vehicles (e.g. ambulances, fire trucks and police cars) are considered.
- Emergency calls are classified into *J* types. Each type of emergency has a limit on maximum response time, hospital requirements, and the number and type of vehicles required. Each type of emergency vehicles has a certain coverage area, which is represented by the portion of nodes that can be reached by that type of vehicles within certain time limit.
- Vehicles that are in stations, on the way to an emergency location, on the way back to station, or are leaving to respond to an emergency, are characterized as having "*divertible*" status. This means that these vehicles can be reassigned to a new destination if the overall system benefits from the re-assignment.
- Real-time traffic information is assumed to be known, which includes the average traffic flow for non-peak hours, and AM and PM peak hours.

The simulation is driven by both events and fixed time steps (incremental time points). The time stamps of events refer to those times at which a vehicle changes its status or an emergency changes its status. The fixed time steps refer to the times at which the traffic information is updated. We rank the time stamps of events and fixed time points and select the earliest one as the next simulation time point. In each simulation point, the program will update the emergency and vehicle information. The information to update for vehicles includes: the current location, the route to take, the destination, the time stamp of next status change, current status, next proposed status, etc. The vehicle status is tightly related to the emergency status. Furthermore, some vehicle status changes may result in the reconsideration of the dispatching decision. For instance, if a vehicle finishes its task and it is on its way back to station, that means the vehicle is available at this point. We may assign it to an emergency location or relocate it to another location for better coverage. Optimization is needed upon this event.

An essential step in developing the simulation model is to generate different modules, such as emergency module and vehicle module. The data structures for emergencies and vehicles in the program are both lists that contain all kinds of necessary information.

7.5.1 Emergency Module

The emergency list contains the following information: (a) the temporal distribution of the service calls; (b) the spatial distribution of the service calls; (c) the priority distribution of the service calls. The priority of an emergency defines the required number of each emergency vehicle type and the service time for each type of emergency vehicles.

7.5.2 Vehicle Module

Each response vehicle in the fleet represents a working crew and provides emergency medical service. Various types of vehicles have varying functionality and ability to respond to particular types of request. In this simulation model, three types (ambulance, fire engine and police car) are considered. But it is easy to add more vehicle type by assigning more values to vehicle type attribute. Vehicle activities are described by keeping track of the location, the status, the destination and the path to destination for each vehicle.

7.5.3 Optimizer Module

At each simulation time point (emergency status change, vehicle status change, or at an incremental time point), the optimizer module makes decision about the movement of all vehicles according to an assignment strategy and the result of simulation. So it is the key module of the operation, receiving and processing all service calls and controlling all activities.

7.5.4 Calibration of Simulation Model

The data used in the calibration are generated from real-world operational data for the ambulances and the medical units during November and December of 2000. More than 3000 records are analyzed. Each record stands for one dispatched vehicle. A series of variables describe various information associated with the vehicle including the time at which the emergency call arrived, vehicle identification number, dispatching time, arrival time and call type. The locations of emergency calls are matched to GIS locations to obtain the historical spatial distribution of emergency calls. ARENA Analyzer is used to find the best fitted distributions for historical data.

7.5.5 Output Analysis

To develop a simulation system to test various dispatching strategies and facility location/allocation plan, plenty of time and energy are spent on the conceptual model development, coding and system calibration. Actually, to get precise estimates of the system performance measures, it is important to appropriately analyze the simulation output. One simulation run is a computer-based statistical sampling experiment. Each run only produces a realization of a set of random variables, which may be far from the true system characteristics. To ensure an appropriate statistical analysis from

simulation results, a number of simulation replications are necessary. The number of replications needed depends on the specified precision, degree of confidence and sample variance.

In addition, when a simulation run starts at time 0, it goes through a transient period, and eventually achieves a steady state with steady demand if the system capacity is not exceeded. Because the output process from the steady-state distribution is considered, it is necessary to discard a specific transient time, which is sometimes called the warm-up period, in which the state of system is not yet stable. The convergence rate depends on the initial condition and the system structure. Based on preliminary experimental results, we select one day as warm-up period.

7.6 MATHEMATICAL MODEL

The key module in the simulation framework is the one that can solve the real-time vehicle dispatching problem. An integer optimization model which considers the vehicle relocation and dispatching decisions jointly has been proposed by Yang *et al.* (2006). Assigning and relocation are considered simultaneously considering the priority of emergency calls.

7.6.1 Notation

V The set of emergency vehicles in the system

K The set of emergency vehicle types in the system, $k = 1,2,...,|K|$

V_k The set of type k emergency vehicles in the system

V_k^1 The subset of type k emergency vehicles in V_k that are staying at home station with "idle" status

V_k^2 The subset of the emergency vehicles in V_k that are moving to an emergency site

V_k^3 The subset of the emergency vehicles in V_k that are servicing an emergency site

V_k^4 The subset of the emergency vehicles in V_k that are leaving for hospitals after finishing on-site service

V_k^5 The subset of the emergency vehicles in V_k that are staying at hospitals

V_k^6 The subset of the emergency vehicles in V_k that are moving to stations

j The index of vehicles in set V_k, $j = 1,2,...,|V_k|$

W The set of emergencies in the system

W^0 The subset of incidents in W that are currently being served by some emergency vehicle

W^1 The subset of incidents in W that are waiting for service

i The index of emergencies in set W, $i = 1,2,...,|W|$

S The set of emergency vehicle stations

s The index of station in set S, $s = 1,2,...,|S|$

H The set of hospitals in the system

h The index of hospital in set H, $h = 1,2,...,|H|$

L The set of nodes in the area

l The index of node in L, $l = 1,2,...N_n$

$N^R_{V_{jk}}$ The set of nodes that can be reached by type k vehicle j within required time

Coefficients

T_{ki} The upper bound of waiting time for type k vehicle to reach emergency i

$d_{kji}(t)$ The predicted travel time for type k vehicle j to arrive at emergency i while departing at time t

$d_{kjh}(t)$ The predicted travel time for type k vehicle j to arrive at hospital h while departing at time t

$d_{kjs}(t)$ The predicted travel time for type k vehicle j to arrive at station s while departing at time t

A_{ik} The penalty associated with the type k vehicle dispatched to waiting emergency i whose travel time is longer than T_{ki}

B_{ik} The penalty associated with type k vehicle deficiency for emergency i

C_{kji} The cost of the type k vehicle j to travel to emergency i, which is a function of travel time $d_{kji}(t)$ and related to the emergency property and vehicle type property.

C_{kjh} The cost of type k vehicle j travel to h^{th} hospital, which is a function of travel time $d_{kjh}(t)$ and related to the property of vehicle type.

C_{kjs} The cost of type k vehicle j travel to s^{th} station, which is a function of travel time $d_{kjs}(t)$ and related to the property of vehicle type

$CH_h(t)$ The vacancy of h hospital at time t

D_k The penalty associated with type k vehicle coverage deficiency for the area

M A large number

N_{ik} The required number of type k vehicle for emergency i

τ The reassignment criterion; when the saving of travel time from reassigning a vehicle is larger than τ, the route change will be performed, otherwise, the assignment will not change, this is to avoid divert/relocation too many vehicles at once or moving the same vehicle too often over a short period

$NC_{lsk}(t)$ The identifier of if node *l* can be covered by type *k* vehicle at station *s* at time *t*, =1 if travel time for type *k* vehicle travel from station *s* to node *l* $t_{ksl}(t) <= T_{v_k}$; =0 otherwise

ρ_k The required coverage rate for type *k* vehicles

A series of variables X^0 stand for the destinations of emergency vehicles in the system in last iteration. Since the values of these variables from last step are known, they are treated as coefficient in the current step.

X^0_{kji} =1 if the type *k* vehicle *j* was dispatched to emergency *i* at last step;
=0 otherwise

X^0_{kjh} =1 if the type *k* vehicle *j* was dispatched to hospital *h* at last step;
=0 otherwise

X^0_{kjs} =1 if the type *k* vehicle *j* was dispatched to station *k* at last step;
=0 otherwise

Decision Variables

$X_{kji}(t)$ =1 if the type *k* vehicle *j* is dispatched to an incident emergency *i* at time *t*;
=0 otherwise

$X_{kjh}(t)$ =1 if the type *k* vehicle j is dispatched to hospital *h* at time *t*;
=0 otherwise

$X_{kjs}(t)$ =1 if the type *k* vehicle *j* is dispatched to station *s* at time *t*;
=0 otherwise

Y_{kj} =1 if the type *k* vehicle *j* is re-assigned;
=0 otherwise

Z_k =1 if the coverage rate of type *k* vehicle is lower than ρ_k.;
=0 otherwise

P_{kji} =1 if the travel time for the type *k* vehicle *j* to the emergency *i* is longer than T_{ik} ;
=0 otherwise

Q_{ik} =1 if the emergency i does not receive required number of type k vehicles;

=0 otherwise

R_{lk} =1 if the node l can be reached by type k vehicles within critical time;

=0 otherwise

7.6.2 Mathematical Model

The mathematical model is as follows:

$$\begin{aligned} Min \quad & \sum_i \sum_j \sum_k (X_{kji}(t) \cdot C_{kji}(t)) + \sum_k \sum_j \sum_h (X_{kjh}(t) \cdot C_{kjh}(t)) + \sum_k \sum_j \sum_s (X_{kjs}(t) \cdot C_{kjs}(t)) \\ & + \sum_i \sum_k (\sum_j P_{kji} \cdot A_{ik}) + \sum_i \sum_k (Q_{ik} \cdot B_{ik}) + \tau \cdot \sum_k \sum_j Y_{kj} + \sum_k D_k \cdot Z_k \end{aligned}$$

Subject to

$$\sum_i X_{kji}(t) + \sum_s X_{kjs}(t) + \sum_h X_{kjh}(t) = 1 \qquad \forall\, j \in V_k, k \in K \qquad (1)$$

$$N_{ik} - \sum_{j \in V_k} X_{kji}(t) \le M \cdot Q_{ik} \qquad \forall \quad i \in W, k \in K \qquad (2)$$

$$X_{kji}(t) \cdot t_{kji}(t) - T_{ki} \le M \cdot P_{kji} \qquad \forall \quad i \in W, j \in V_k, k \in K \qquad (3)$$

$$\sum_k \sum_{j \in V_k^5} X_{kjh}(t) \le CH_h(t) \qquad \forall \quad h \in H, \quad k \in K \qquad (4)$$

$$X_{kjh}(t) = X_{kjh}^0 \qquad \forall \quad i \in W^0, j \in V_k^5 \qquad (5)$$

$$X_{kji}(t) = X_{kji}^0 \qquad \forall \quad i \in W^0, j \in V_k^3, k \in K \qquad (6)$$

$$\sum_h X_{kjh}(t) = 1 \qquad \forall \quad j \in V_k^4, k \in K \qquad (7)$$

$$\sum_s (NC_{lsk}(t) \cdot \sum_j X_{kjs}(t)) - R_{lk} \ge 0 \qquad \forall \quad k \in K, l \in L \qquad (8)$$

$$\rho_k \cdot N_n - \sum_l R_{lk} \le M \cdot Z_k \qquad \forall \quad k \in K \qquad (9)$$

$$1 - X_{kji}(t) \cdot X^{0}_{kji} \le M \cdot Y_{kj} \qquad \forall \quad i \in W, j \in V^{2}_{k}, k \in K \tag{10}$$

$$\sum_{i} d^{0}_{kji} \cdot X^{0}_{kji} - \sum_{i} d_{kji}(t) \cdot X_{kji}(t) - \tau \le M \cdot Y_{kj} \quad \forall j \in V^{2}_{k}, k \in K \tag{11}$$

The objective is to minimize the weighted total travel time at any time t, with consideration of number and type of required vehicles, reassignment criterion, and area of uncovered region. The weighted total travel time is composed of the weighted travel times to emergencies waiting for service, the weighted travel times to hospitals and the weighted total travel times to relocation sites for each type of vehicle. Higher weights are assigned to more severe emergency types so that they are served faster. The penalties of deficiency in required vehicle number, and low coverage can be translated into travel time using appropriate coefficients.

Constraints (1) guarantee that a vehicle can only be dispatched to an emergency or a hospital or stay at a station.

Constraints (2) state that each emergency should be served by the required number of vehicles of appropriate type. A penalty will be applied if there is any deficiency.

Constraints (3) bring a penalty coefficient to the formulation: when the waiting time of any emergency is more than the upper bound of the waiting time, a penalty will be applied.

Constraints (4) state that the number of patients sent to a hospital cannot exceed the vacancy of that hospital at time t. It is assumed that one vehicle will only carry one patient to the hospital.

Constraints (5) and (6) state that the vehicles serving at an emergency site cannot be re-assigned.

Constraints (7) state that the vehicles traveling to hospitals can change their destination to other hospitals.

Constraints (8) state that the node is defined as covered by type k vehicle when there are greater or equal than one type k vehicle that can reach it within critical time.

Constraints (9) bring a penalty coefficient to the formulation: when the coverage for the area is lower than the required percentage for type k vehicle, a penalty will be applied.

Constraints (10) and (11) state that the vehicles can be reassigned when the reassignment can bring a travel time saving that is larger than pre-set criterion τ, otherwise, the vehicles will be not reassigned.

7.6.3 Cost of Travel Time C_{kji}, C_{kjh}, C_{kjs}

As described in the notation, C_{kji}, the cost of the type *k* vehicle *j* traveling to emergency *i* is a function of travel time $t_{kji}(t)$, the emergency type and vehicle type. Here, we define

$$C_{kji} = t_{kji}(t) \cdot EP_i \cdot VP_k \tag{12}$$

Where EP_i is the type of emergency *i*, the emergency with a higher priority will have a larger value. VP_k is a coefficient associated with type *k* vehicle that indicates the relative importance of type *k* vehicle in response to emergency *i*. Since the travel speed varies with the vehicle type, this coefficient adjusts the travel speed of each type of emergency response vehicle.

Similarly, the cost of vehicle *j* traveling to hospital *h* is derived as (13).

$$C_{kjh} = t_{kjh}(t) \cdot EP_i.VP_k \tag{13}$$

The vehicle traveling to station has no emergency to deal with, so in (14), only the coefficient associated with type *k* vehicle is considered.

$$C_{kjs} = t_{kjs}(t) \cdot VP_k \tag{14}$$

7.6.4 Coverage Rate ρ_k

When considering whether the area is under good coverage of the emergency response fleet, we address two aspects, one is the average performance and the other is the worst-case performance. To represent the average performance, we use the total number of nodes that can be reached by certain type of assignable vehicles within required time limits. This approach guarantees that the emergency calls from those nodes are served within the time requirements. After calculating the all-to-all travel times, the nodes to which type k vehicles in the stations have travel times less than T_{v_k} will be recorded as a covered node by vehicle type *k*. We define ρ_k as the total number of covered nodes divided by the total number of nodes in the network.

7.6.5 Penalty Associated with the Coverage Deficiency D_k

The penalty associated with the coverage deficiency D_k plays an important role in the model and its value directly influences the structure of the solution. When D_k is too large, relocation of vehicles for service coverage will have higher priority than dispatching vehicles to respond to emergency calls. The solution is dominated by the coverage penalty and the problem is transformed into a sequential combination of the Maximum Covering Problem and the Generalized Assignment Problem. As a result, the response times for emergencies will increase. When D_k is too small, vehicle relocations may not happen because the benefits accrued from vehicle relocations maybe smaller than the relocation costs. The number of nodes in the network and the number of vehicles need to be considered at the same time when choosing the value of D_k. This range of the penalty coefficient is decided through experiments.

Exact solution can be obtained by solving the model using optimization software, e.g. CPLEX. In this way, the optimizer module in simulation model is to input the formulation into, and read solution from CPLEX. When the size of fleet or the candidate relocation sites is large, this approach has its limitation in getting optimal solution within reasonable computation time. For fire trucks, the relocation sites may be fire stations only, while for police cars, the candidate relocation sites can be any node in the network. To deal with this limitation, a tabu-search based algorithm is introduced.

In this study, we use a rolling-horizon approach to reduce the number of potential relocation sites for police cars. Instead of using all nodes in the network as relocation sites, only those nodes reachable by a police car within a time contour are considered as its potential relocation sites in the optimization model. Note that the system is updated after every small time interval (e.g. <= 5 minutes), therefore, the size of the relocation sites is effectively reduced and CPLEX can provide exact solution within reasonable computation time.

7.7 CASE STUDY

A real street network is used in this case study. The network represents a part of the Washington DC metropolitan area that has 1757 nodes and 2146 links. The nodes represent the intersections of streets or important locations in the streets. The lengths of the links range from 0.01 mile to 1.7 mile. Less than 2 percent of the links are longer than 0.3 mile. There are 10 fire stations and 2 hospitals in the area and their locations are mapped to the nearest nodes in the network. The data reported in this chapter is carefully generated from

real world operational data. We have ensured that the data used in the test problems represents the same characteristics of the real world data. The simulation model is calibrated by the real operation data.

Based on the analysis of historical data, the emergency calls are grouped into 4 types of priorities, the inter-arrival times of emergency calls follow an exponential distribution with an average inter-arrival time of 30 minutes and the service times of emergency calls follow a normal distribution. It should be noted that the number of emergency calls in a sub-region is proportional to the number of nodes in that sub-region according to the historical data. Therefore, we assume emergency calls are uniformly distributed spatially. With these fitted distributions, emergency calls are generated accordingly in the simulation model.

7.7.1 Comparison of Dispatching Strategies

To compare alternative dispatching strategies under different emergency frequency, we consider average response time, maximum response time and the ratio of emergency events whose response time exceeds the pre-specified response time limit to total accidents. The average response time is the main criterion to judge a dispatching strategy since it plays a crucial role in minimizing the adverse impacts. Several sets of problems were solved under four scenarios: (a) FCFS, (b) NO, (c) optimized dispatching without coverage concern, and (d) optimized dispatching with coverage concern. The same series of emergency calls were used in all scenarios. These problems were solved using CPLEX version 9.0. The results of ambulances' performance measures are shown as Table 7-2.

It is obvious that the results for ambulances under NO is close to the results under optimized dispatching scheme, while under FCFS strategy the response time is much longer than the other scenarios. The average response time under scenario (d) is about 2% less than that of scenario (c). However, the number of vehicles that arrive at the emergency sites later than the maximum allowable waiting time for the emergencies decreases by 5%. This indicates that vehicle relocation allows more vehicles to reach emergency sites within the desired response time. For fire engines and police cars, similar patterns can be observed.

Table 7-2. Summary of computation results.

Discipline	Average Response Time (minutes)	Longest Response Time (minutes)	% vehicles with Response time $>T_{ijk}$
(a)	8.63	20.10	21%
(b)	3.51	14.37	16%
(c)	3.07	9.78	14%
(d)	3.02	7.90	9%

7.7.2 Computation Time

The computation time is a major issue for online applications. As shown in Table 7-3, for scenarios (a), (b) and (c), the computation times are almost negligible even when the fleet size and number of emergency calls are large enough. While for scenario (d), the computation times increase exponentially with the size of relocation site. Based on the real world data, we designed a set of test problems with relocation node set N', fleet set V and emergency set E. The locations and types of emergencies in these problems are generated according to the historical possibility density function.

When the vehicles are only allowed to be relocated to stations (N'=10), the computation time is very small when number of emergencies waiting for service is less than 8 and the overall fleet size is 80. In real world some emergency response vehicles can have many more candidate sites for relocation other than stations. For example, ambulances can stay at hospitals and police cars can be located almost anywhere in the network. For police cars especially, the size of relocation site can be very large (>500). This factor has a significant contribution to increasing the computation times of CPLEX. In this case study a rolling-horizon approach is used. The number of potential relocation sites is effectively reduced and the computation speed is fast enough for real operation. For larger networks, an efficient heuristic is needed to provide quality solutions promptly.

Table 7-3. Comparison of computation times.

Discipline	(c) by CPLEX	(d) by CPLEX
Size (||N||, ||V||, ||E||)	Computation Time (seconds)	Computation Time (seconds)
(10, 30, 1)	0.03	0.52
(10, 30, 3)	0.03	0.91
(10, 30, 5)	0.02	1.05
(10, 30, 8)	0.01	1.48
(10, 50, 1)	0.03	1.02
(10, 50, 3)	0.05	1.05
(10, 50, 5)	0.06	1.09
(10, 50, 8)	0.03	1.13
(10, 80, 1)	0.08	1.28
(10, 80, 3)	0.09	1.29
(10, 80, 5)	0.02	1.30
(10, 80, 8)	0.08	1.32
(100, 80, 10)	0.12	19
(500, 80, 10)	0.36	241

7.7.3 Comparison of Shortest Path Algorithms on Average Response Time (ART)

When comparing the shortest path algorithms, we solved a set of problems based on two scenarios. In the first scenario we always used the optimal route obtained by using a static shortest path algorithm based on the off-peak time traffic information. This is the case in real-world operation, where drivers are provided with the routes calculated by off-the-shelf mapping software which use speed limits as travel speeds all the time. In the second scenario we used the time dependent shortest path algorithm. 120 problems were generated to test the impact of use of different shortest path algorithms on emergency response time. The solutions of the problem sets include more than 500 response times during an interval of 24 hours, and the average of these response times are used in the analysis. Time dependent shortest path algorithm takes advantage of the traffic fluctuation. When travel speeds on links are stable, the travel times under these two scenarios are same. As shown in Table 7-4, when comparing the average response time during an entire day, the difference is not very impressive. However, if we consider the average response time during peak hours, the advantage of time dependent shortest path algorithm is clear. For some extreme cases the response times decrease around 20% by utilizing time dependent shortest path and online traffic information.

The results shows that using a time dependent shortest path algorithm will benefit vehicle dispatching and routing only if the travel speeds on links vary over the duration of the trip. When the travel times are stable, the $t_{kji}(t)$ calculated by both static and time dependent shortest path algorithms are the same.

7.7.4 Impact of Penalty Coefficients

For the optimized dispatching with coverage constraints, the penalty coefficients in the objective function have a potential impact on the average

Table 7-4. Comparison of shortest path algorithm based on average response time.

Time Interval	ART (Static) (minutes)	ART (DSP) (minutes)	Average improvement (minutes)	Average Improvement (%)
7:30 am - 8 am	3.34	3.22	0.12	3.6
8:00 am -10 am	3.71	3.51	0.2	5.4
Non-peak Time	2.44	2.44	0	0
All Day	3.01	2.98	0.03	1.0

system performance measures. When the coverage penalty is much larger than the penalty coefficients for assignment requirements, the system is more likely to dispatch more vehicles to the candidate relocation sites instead of dispatching them to some less important emergency calls. Therefore, it is important to perform sensitivity analysis on these penalty coefficients. Here two groups of experiments are designed. In the first group, the weight of travel time for all types of emergency calls are the same, and we increase the coverage penalty coefficient. In the second one, we designate variable coefficients to each type of emergency calls. Table 7-5 summarizes the performance measures of ambulances under various penalty coefficient combinations. With the uniform travel time coefficient values for all types of calls, the shortest average response times can be achieved but the longest response time and the percentage of emergency calls exceeding waiting time limits are the highest. When the value of coverage penalty coefficient increases, the average response time slightly increases but the other two performance measures improve.

7.7.5 Impact of Location Plans of Station

With a given fleet, the location plan of stations plays an important role in improving the system performance. Especially, for ambulances and fire engines, stations are their major potential relocation sites. Figure 7-8 shows the variation of average response time with respect to different number of stations at arbitrary locations. When the number of stations reduce from 10 (current situation) to 5, the average response time increases by about 9%. When only one station can be operated, the average response time increases by 99%.

Table 7-5. Sensitivity analysis of penalty coefficients for ambulances.

Coefficient Ratios (EP_1:EP_2:EP_3:EP_4:D_c)	Average Response Time (minute)	Longest Response Time (minute)	% of Emergency Exceed Waiting Time Limit
(1:1:1:1:1)	2.78	10.65	13.79
(1:1:1:1:5)	2.83	10.61	13.61
(1:1:1:1:10)	2.81	8.72	9.84
(1:1:1:1:20)	3.18	8.47	9.17
(1:1:1:1:50)	3.23	8.34	9.00
(1:2:3:4:1)	2.94	9.63	12.51
(1:2:3:4:5)	2.95	9.63	12.32
(1:2:3:4:10)	3.02	7.91	8.92
(1:2:3:4:50)	3.32	7.63	8.27
(1:2:3:4:100)	3.46	7.52	8.12

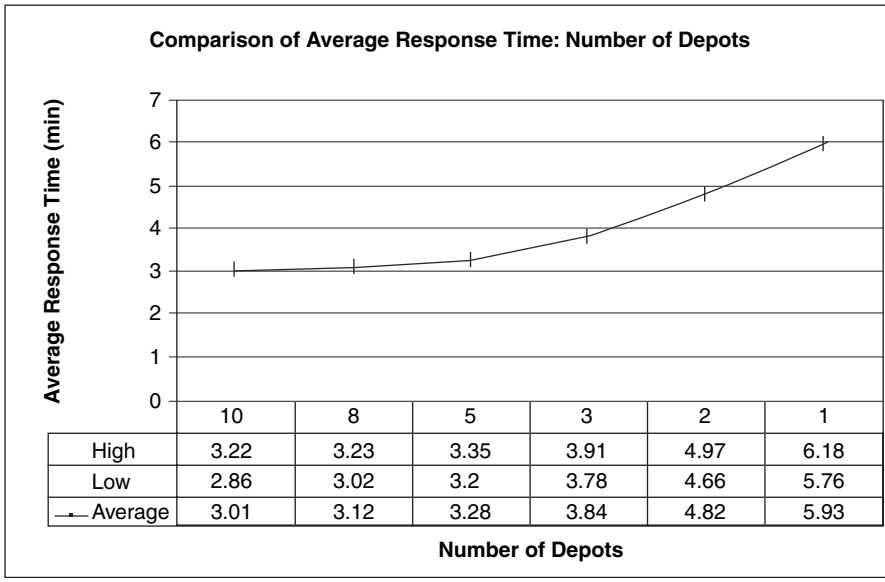
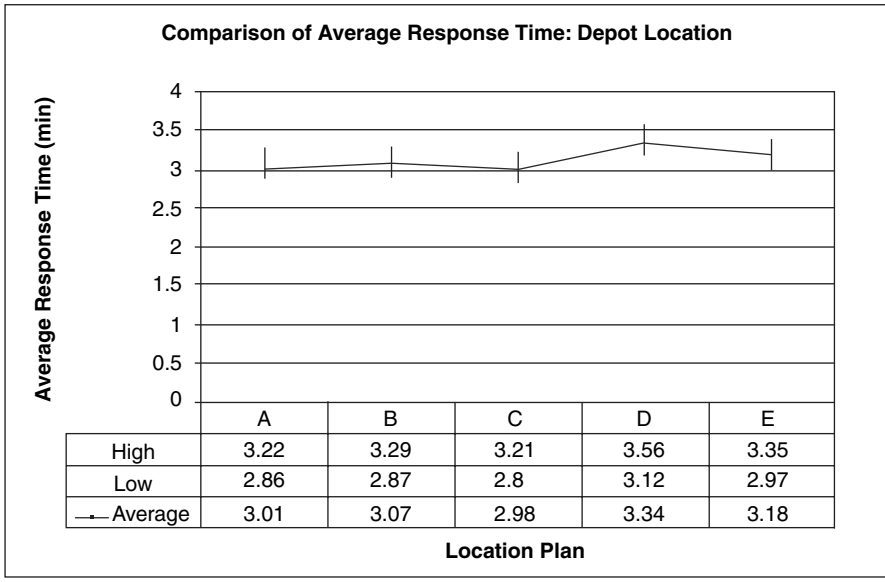


Figure 7-8. Impact of mumber of depots on average response time.

With a given number of stations and a fixed fleet, the locations of stations have significant impact on system performance. In addition to the existing 10 station locations (plan A), we randomly selected other location plans for 10 stations (plans B-E), and tested their performance. Figure 7-9 shows a better location plan can save significant amount of response time. In this set of experiments, the best scenario (plan C) has an average response time of 2.98 minutes while the worst case (plan D) has an average response time of 3.34 minutes. The difference is more than 11%.

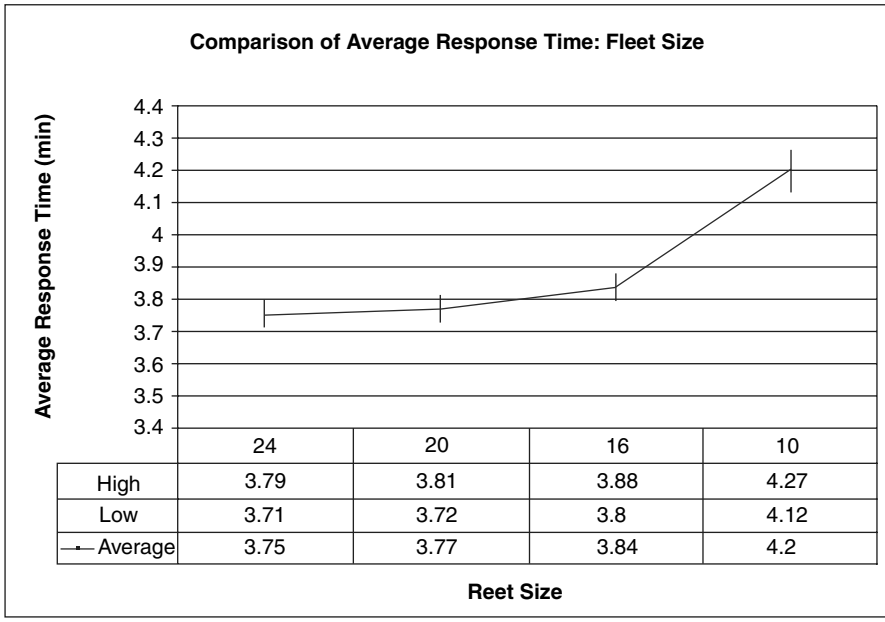
Figure 7-10 shows the results from another group of experiments with given a station location plan but variable fleet size. When the fleet size decreases from 24 to 10, the average response time increases 12%.

It is important to note that the simulation results also indicate that when the location plan and fleet size plan can be optimized together, the benefit might be very promising.



	A	B	C	D	E
High	3.22	3.29	3.21	3.56	3.35
Low	2.86	2.87	2.8	3.12	2.97
Average	3.01	3.07	2.98	3.34	3.18

Figure 7-9. Impact of station locations on average response time.



	24	20	16	10
High	3.79	3.81	3.88	4.27
Low	3.71	3.72	3.8	4.12
Average	3.75	3.77	3.84	4.2

Figure 7-10. Impact of fleet size on average response time.

7.8 CONCLUSIONS AND FUTURE RESEARCH

In this chapter, an extensive literature review of real-time emergency vehicle deployment is provided. An integrated emergency vehicle fleet management system is proposed and a simulation model is developed based on the system framework. Several dispatching strategies are examined in the simulation model. A mathematical model for real time vehicle dispatching and routing problem is presented. Exact solution of the dispatching scheme that minimizes the expected total wait over a large network can be obtained with a short computation time. The performance of the proposed approach is promising. When the number of relocation sites increases in the operation of specific emergency response vehicle, the computation time will increase dramatically. By applying a rolling horizon approach in the model, the number of relocation sites can be effectively reduced so that the exact solutions can be obtained in short time. However for very large and dense road networks, the computation time may still exceed limit. In such cases, an efficient heuristic algorithm is needed to provide quality solutions in short computation times. Since multiple objectives are weighted in the heuristic, an economic analysis of the tradeoffs between the operational costs and the benefits gained from vehicle relocations and reassignments is an interesting area for future research. Bi-level optimization can be another alternative approach to the problem. When we take the regional traffic impact of a severe emergency into account, this problem becomes much more complicated. In that case, the traffic volume will be influenced by the emergency event and the travel times on links will vary and it is difficult to capture this type of variation using short term predication models. Parallel processing and meso-scopic simulation models may be necessary to predict the potential travel time fluctuations.

REFERENCES

Ball, M. O and F.L. Lin, Reliability Model Applied to Emergency Service Vehicle Location, *Operations Research* 41, pp.18-36 (1993).

Brotcorne L, G. Laporte and F. Semet, Ambulance Location and Relocation Models, *Eur J Opl Res,* Vol 147, pp. 451-463 (2003).

Brown, G.G. and R. McBride, Solving Generalized Networks, *Management Science*, Vol. 20, pp. 1497-1523 (1985).

Carter, G. and E. Ignall, A Simulation Model of Fire Department Operations, *IEEE System Science and Cybernetics*, Vol. 5, pp. 282-293 (1970).

Catrysse, D. and Van Wassenhove, L.N, A Survey of Algorithms for the Generalized Assignment Problem, *European Journal of Operational Research,* Vol. 60, pp. 260-272 (1993).

Chabini, I., Discrete Dynamic Shortest Path Problems in Transportation Application, *Transportation Research Record*, No. 1645,pp. 170-175 (1998).

Chaiken, J. and R. Larson, Methods for Allocating Urban Emergency Units: A Survey, *Management Science*, 19, pp. 110-130 (1998).

Chang, E., Traffic Estimation for Proactive Freeway Traffic Control, *Transportation Research Record*, No.1679, pp. 81-86 (1999).

Chang, M. F. and D. C. Gazis, Traffic Density Estimation with Consideration of Lane Changing, *Transportation Science*, Vol. 9, No. 4, pp. 308-320 (1975).

Chu, P.C. and J.E. Beasley, A Genetic Algorithms for the Generalized Assignment Problem, *Camp. Operations Research* 24 (1), pp.17-23 (1997).

Church, R. L. and C. Revelle, The Maximal Covering Location Problem, *Papers of the Regional Science Association,* Vol. 32 , pp. 101-118 (1974).

Cooke, K. and E. Halsey, The Shortest Route Through a Network with Time-Dependent Internodal Transit Times, *Journal of Mathematical Analysis and Applications*, Vol. 14, pp. 493-498 (1966).

Cragg, C. A., and M. J. Demetsky, Final Report: Simulation Analysis of Route Diversion Strategies for Freeway Iincident Management, VTRC 95-R11, Traffic Research Advisory Committee, FHWA, USDOT (1995).

Daskin, M., A Maximum Expected Covering Location Model Formulation, Properties and Heuristic Solution. *Transportation Science*, Vol. 17, 48-70 (1983)

Dijkstra, E. W., A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, 1, pp. 269-271 (1959).

Eldor, M., Demand predictors for computerized freeway control systems", Proceedings of the 7th International Symposium on Transportation and Traffic Theory, Kyoto, Japan, pp. 341-358 (1977).

Fisher, M. L., An Applications Oriented Guide to Lagrangian Relaxation, *Interface*, Vol. 15, pp. 10-21 (1985).

Fisher, M.L., R. Jaikumar and L.N. Wassenhove, A Multiplier Adjustment Method for the Generalized Assignment Problems". *Management Science* 32, 1986, pp. 1095-1103 (1986).

Fitzsimmons, J., A Methodology for Emergency Ambulance Deployment, *Management Science*, Vol. 19, No. 6, pp. 627-636 (1973).

Gafarian, A.V., J. Paul, and T. L. Ward, Discrete Time Series Models of a Freeway Density Process, Proceedings of the 7th International Symposium on Transportation and Traffic Theory, Kyoto, Japan, pp.387-411 (1977).

Gendreau, M, G. Laporte and F., Semet, A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation, *Parallel Comput,* 27, pp. 1641–1653 (2001).

Gendreau, M, G. Laporte, and F. Semet, The Maximal Expected Coverage Relocation Problem for Emergency Vehicles, *Journal of Operation Research Society,* Vol. 57, (1), pp. 22-28 (2005).

Goldberg, J., Dietrich, R., Chen, J., M. Mitwasi, Validating and Applying a Model for Locating Emergency Medical Vehicles in Tucson, AZ, *European Journal of Operational Research*, No.49, pp. 308-324 (1990).

Goldberg, J. and F. Szidarovszky, Method for Solving Nonlinear Equations Used in Evaluating Emergency Vehicle Busy Probabilities, *Operations Research,* Vol. 39, pp. 903-916 (1991a).

Goldberg, J., and L. Paz, Locating Emergency Vehicle Bases when Service Time Depends on Call Location, *Transportation Science*, Vol. 25, No.4, pp. 264-280 (1991b).

Haghani, A., H. Hu, and Q. Tian, An Optimization Model for Real-Time Emergency Vehicle Dispatching and Routing, *Proceeding CD of the 82nd annual meeting of the Transportation Research Board*, Washington, D.C., 2003.

Hakimi, S., Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph, *Operations Research* 12, pp. 450-459 (1964).

Hall, R., The Fastest Path through a Network with Random Time-Dependent Travel Times", *Transportation Science*, Vol. 20, No. 3, pp. 182-188 (1986).

Hoffman, C. and Janko, J., Travel Time as a Basis of the LISB Guidance Strategy, *Proceedings of IEEE Road Traffic Control Conference,* IEEE, New York, pp. 6-10 (1988).

Hogan, K. and C. ReVelle, Concepts and Applications of Backup Coverage, *Management Science*, Vol. 32, pp. 1434-1444 (1986).

Huisken, G., Soft-Computing Techniques Applied to Short-term Traffic Flow Forecasting, *Systems Analysis Modeling Simulation*, Vol.43-2, pp. 165-173 (2003).

Ignall, E.D., P. Kolesar, and W.E. Walker, Using Simulation To Develop and Validate Analytic Models: Some Case Studies, *Operations Research*, Vol. 26, No. 2, pp. 237-253 (1978).

Kaysi, I., M. Ben-Akiva and H. Koutsopulos, An Integrated Approach to Vehicle Routing and Congestion Prediction for Real-Time Driver Guidance, *Transportation Research Record*, Vol. 1408, pp. 66-74 (1993).

Kolesar, P., W. E. Walker, J. Hausner, Determining the Relation between Fire Engine Travel Times and Travel Distances in New York City Companies, *Oper. Res.,* 23(4), pp. 614–627 (1975a).

Larson, R., A Hypercube Queuing Model for Facility Location and Redistricting in Urban Emergency Services, *Comput. & Ops. Res*., Vol. 1, pp. 67-95 (1974).

Larson, R., Approximating the Performance of Urban Emergency Service Systems, *Operations Research*, Vol.23, No.5, pp. 845-868 (1975).

Lorena, L.A.N. and M.G. Narciso, Relaxation Heuristics for a Generalized Assignment Problem, *European Journal of Operational Research*, Vol. 19, No. 3, pp. 600-610 (1996).

Lorena, L., M. G. Narciso, J. E. Beasley (2002); A Constructive Genetic Algorithm for the Generalized Assignment Problem, http://www.lac.inpe.br/~lorena/gap/CGA-PGA-2000.pdf

Marinov, V. and C. Revelle, Siting Emergency Services, in Facility Location: A Survey of articles, applications and methods, edited by: Drezner, Z, *Springer Series in Operations Research*, pp. 199-222 (1995).

Martello, S.,W. R. Pulleyblank, P. Toth, and D. de Werra, Balanced Optimization Problems, *Operations Research Letters* 3, pp.275-278 (1984).

Nahi, N.E., Freeway Ttraffic Data Processing, *Proceedings of the IEEE,* 61, No. 5, pp. 537-541 (1973).

Narciso, M.G. and L.A.N. Lorena, "Lagrangian/surrogate relaxation for generalized assignment problems", *European Journal of Operational Research*, Vol. 114, No. 1, pp. 165-177 (1999).

Nicholson H. and C. D. Swann, The Prediction of Traffic Flow Volumes Based on Spectral Analysis, *Transportation Research Record*, Vol. 8, pp. 533-538 (1974).

Nulty, W. G. and M. A. Trick, GNO/PC Generalized Network Optimization System, *O.R. Letters,* Vol. 2, pp. 101-112 (1988).

ReVelle, C., and K. Hogan., A Reliability-Constrained Siting Model with Local Estimates of Busy Fractions, *Environment and Planning B: Planning and Design,* 15, pp. 143-152 (1988).

Revelle, C., Extension and Prediction in Emergency Service Siting Models, *European Journal of Operational Research*, Vol. 40, pp. 58-69 (1989).

Revelle, C. , "A Perspective on Location Science, *Location Science,* 5, No.1" pp. 3-13 (1997).

Ross, G. T. and M. S. Soland, A Branch and Bound Algorithm for the Generalized Assignment Problem, *Mathematical Programming* 8, pp. 91-103 (1975).

Savas, E.S., Simulation and Cost-Effectiveness Analysis of New York's Emergency Ambulance Service. *Management Science*, Vol. 15, No. 12, pp. 608-627 (1969).

Schilling, D. A., D. Elzinga, J. Cohon, R. Church and C. Revelle, The TEAM/FLEET Mmodels for Simultaneous Facility and Equipment Siting, *Transportation Science*, 13(2), pp. 163-175 (1979).

Schilling, D. A., J. Vaidyanathan and R. Barkhi, A Review of Covering Problems in Facility Location, *Location Science*, Vol. 1, pp. 25-55 (1993).

Shantikumar, J.G., and R.G. Sargent, A Unifying View of Hhybrid Simulation/Analytic Models and Modeling, *Operations Research*, Vol. 31, No. 6, pp. 1030-1052 (1983).

Smith, B., Demetsky, M., Short-Term Traffic Flow Prediction: Neural Network Approach, *Transportation Research Record*, No. 1453, pp. 98-104 (1995).

Stephanedes, Y. J., P.G. Michalopoulos, and R.A. Plum, Improved Estimation of Traffice Flow for Real-Time Control, *Transportation Research Record*, Vol. 795, pp. 28-39 (1981).

Toregas, C., Swain, R., ReVelle, and C. Bergman, L., The Location of Emergency Service Facilities, *Operations Research*, Vol. 19-6, pp. 1363-1373 (1971).

Toregas, C., Swain, R., ReVelle, and C. Bergman, L., Reply to Rao's Note on the Location of Emergency Service Facilities, *Operations Research*, Vol. 22-6, pp. 1262-1267 (1974).

Trick, M. A., A Linear Relaxation Heuristic for the Generalized Assignment Problem, *Naval Research Logistics*, Vol. 39, pp. 137-152 (1992).

Yang, Saini, Masoud Hamedi and Ali Haghani, An On-line Emergency Vehicle Dispatching and Routing Model with Area Coverage Constraints, *Transportation Research Record,* No. 1923, pp. 1-9 *(2006)*.

Ziliaskopoulos, A., and H. Mahmassani, Time Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications, *Transportation Research Record* 1408, pp. 94-100 (1993).

Zografos, K., Douligeris, C. and C. Lin, A Model for the Optimum Deployment of Emergency Repair Trucks: An Application in the Electric Utility Industry, *Transportation Research Record*, 1358, pp. 88-94 (1992).

Zografos, K., Douligeris, C., and C. Lin, A Simulation Model for Evaluating the Performance of an Emergency Response Fleet, *Transportation Research Record*, 1452, pp. 27-34 (1994)

Chapter 8

VEHICLE ROUTING AND SCHEDULING MODELS, SIMULATION AND CITY LOGISTICS

Jaime Barceló[1], Hanna Grzybowska[1] and Sara Pardo[2]
[1]Dept. of Statistics and Operations Research, Universitat Politècnica de Catalunya, Campus Nord, Mòdul C5, Jordi Girona Salgado 1-3, 08034 Barcelona, Spain; [2]TSS-Transport Simulation Systems, Passeig de Gràcia 12, 3º1ª, 08007 Barcelona, Spain

Abstract: The distribution of goods based on road services in urban areas, usually known as City Logistics, contributes to traffic congestion and is affected by traffic congestion, generates environmental impacts and incurs in high logistics costs. Therefore a holistic approach to the design and evaluation of City Logistics applications requires an integrated framework in which all components could work together that is must be modelled not only in terms of the core models for vehicle routing and fleet management, but also in terms of models able of including the dynamic aspects of traffic on the underlying road network, namely if Information and Communication Technologies (ICT) applications are taken into account. This paper reports on the modelling framework developed in the national projects SADERYL-I and II, sponsored by the Spanish "Dirección General de Ciencia y Tecnología" (DGCYT) and tested in the European Project MEROPE of the INTERREG IIIB Programme. The modelling framework consists of a Decision Support System whose core architecture is composed by a Data Base, to store all the data required by the implied models: location of logistic centres and customers, capacities of warehouses and depots, transportation costs, operational costs, fleet data, etc.; a Database Management System, for the updating of the information stored in the data base; a Model Base, containing the family of models and algorithms to solve the related problems, discrete location, network location, street vehicle routing and scheduling; a Model Base Management System, to update, modify, add or delete models from the Model Base; a GIS based Graphic User Interface supporting the dialogues to define and update data, select the model suitable to the intended problem, generate automatically from the digital map of the road network the input graph for the Network Location and Vehicle Routing models, apply the corresponding algorithm, visualize the problem and the results, etc. To account for the dynamics of urban traffic flows the system includes an underlying dynamic traffic simulation model (AIMSUN in this case) which is able to track individually the fleet vehicles, emulating in this way the monitoring of fleet vehicles in a real time fleet management system, gathering dynamic data (i.e. current position, previous position, current speed, previous speed, etc.) while

following the vehicle, in a similar way as the data that in real life an equipped vehicle could provide. This is the information required by a "Dynamic Router and Scheduler" to determine which vehicle will be assigned to the new service and which will be the new route for the selected vehicle.

Keywords: City Logistics, Dynamic Traffic Simulation, Vehicle Routing

8.1 INTRODUCTION

Logistics, as defined by the Council of Logistics Management, CLM 2001, is the part of the supply chain process that plans, implements, and controls the efficient, effective flow and storage of goods, services, and related information from the point of origin to the point of consumption in order to meet customers' requirements. However, when logistics activities take place in urban areas they show unique characteristics making them different from the general logistics activities, which is the reason why freight transport in urban areas, and specifically the freight flows associated to the supply of city centres with goods, is usually referred to as "city logistics".

Taniguchi *et al.* (2001) define City Logistics as "the process of totally optimising the logistics and transport activities by private companies in urban areas while considering the traffic environment, traffic congestion and energy consumption within the framework of a market economy". Among the special characteristics of urban freight transport there is two of particular relevance: its contribution to the traffic flows, and the subsequent environmental impacts. According to Koriath *et al.* (1998) from the total traffic within urban areas, freight transport (Lorries > 3,5 to) has an average share of about 10 %. If vans and cars, which are currently becoming more important, are included this share would be much higher

The importance of urban freight transport can also be shown by the cost distribution within the freight transport chain. The share of pick-up and delivery operations, which often take place in urban areas, on the total door-to-door cost is in combined transport about 40 % (Inner Urban Freight Transport). The weight of these costs is further increased by the reduction of stocks, the smaller size of consignments and the increase in their number.

From a systems approach City Logistics systems have many components which must be identified and defined, usual approaches to model City Logistics systems identify fleet planning and management as the core components to which modelling of the dynamic impacts of traffic congestion should be added however, the advent of ICT has brought a new dimension to City Logistics applications, namely in what concerns the information systems and e-commerce, which appear as the most influencing ones and this implies that from the operational point of view the type of Vehicle Routing and Scheduling models

that become more relevant in this context are those which account for processes dealing explicitly with time dependent phenomena as in those cases when customers specify a time-window to be visited by the pick-up/delivery trucks or when the vehicle routing and scheduling has to be dynamic based on real time information that changes whilst vehicles are distributing goods and a sequential updating of routes should occur when new information is received. Examples of types of real time information could be: on system performance (i.e. travel times affected by congestions, incidents and breakdowns), changes in service or waiting times; changes on customer demand, calls from new customers, etc.; or vehicle changes as its location or load status.

Therefore models to account for this new dimension of City Logistics must be models that, further than including the main components of City Logistics applications, should be able of including also the dynamic aspects required to model ICT applications. Methodological proposals of this type have been formulated by Taniguchi *et al.* (2000), Taniguchi *et al.* (2001) and Kohler (1997). Figure 8-1, a slightly modified version of figure 7.13 of Taniguchi *et al.* (2001), summarizes the conceptual scheme of the methodology proposed by Taniguchi. The dynamic traffic simulation models emulates the actual traffic conditions, providing at each time interval the estimates of the current travel times, queues, etc. on each link of the road network. This will be the information used by the logistic model (i.e. a fleet management system identifying in real-time the positions of each vehicle in the fleet and its operational conditions - type of load, available capacity, etc.) to determine the optimal routing and scheduling of the vehicle.

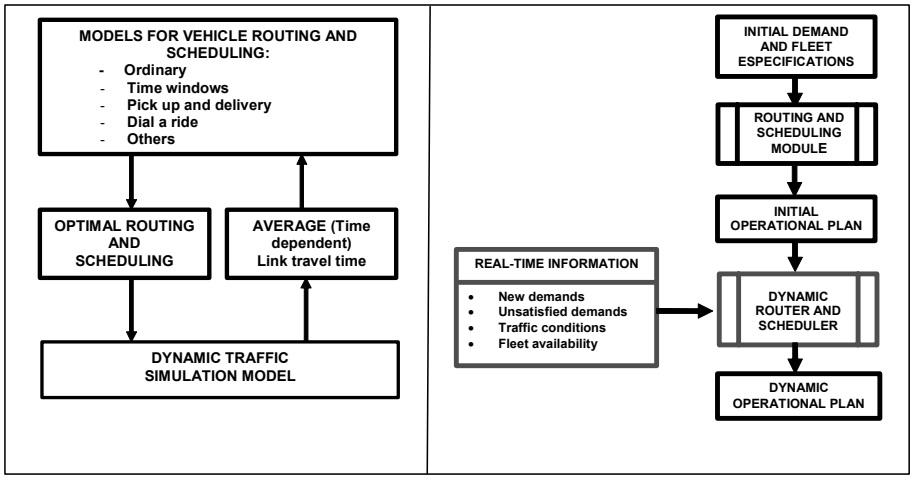


Figure 8-1. On the left conceptual diagram of an integrated "Routing-Simulation" approach for modelling "City Logistic" schemes in presence of ICT; on the right conceptual scheme for the evaluation of real-time fleet management systems.

A further step is the case of dynamic fleet management systems in which carrier fleet operators should be able to respond to changes in demand, driver and vehicle availability taking also into account the time changes in traffic network conditions. These systems are essential to take advantage of real-time information made possible by technological advances in location, communication and geographic information systems. To model properly this new dimension, brought to City Logistics by ICT, dynamic simulation models become a key component of the system (Regan *et al.* 1997; Regan *et al.* 1998). Such conditions can be represented effectively in a computer simulation modelling framework, which provides the requisite flexibility of strategy representation and complex process emulation for the evaluation of dynamic fleet management systems. The approach proposed in this paper, complementary to that of Taniguchi, includes a procedure for generating a set of initial vehicle assignments that would take known and predicted future demands into account, and incorporate strategies for reacting to changes as they occur. The conceptual scheme of the modelling framework is depicted in Figure 8-1. The logic process, depicted in the diagram of Figure 8-1, assumes a partial knowledge of the demand. The service starts at the beginning of the time period (i.e. the day) proposing an initial schedule of the available fleet to service the known demand, this is the initial operational plan which will be modified accordingly later on, when the operation starts, and new information on real time is available; this information can correspond to new demands, unsatisfied demands, changes in the routes due to traffic conditions, changes in the fleet availability (i.e. vehicle breakdowns) and others, which feeds a dynamic router and scheduler that computes a new dynamic operational plan.

To implement computationally the scheme proposed by Taniguchi *et al.* (2001), a key decision is which dynamic traffic simulation approach is the most suitable one. There are two main approaches. Mesoscopic models, describing in a simplified way the flow dynamics and implementing heuristically a dynamic equilibrium, can computationally succeed in the analysis of large networks. DYNASMART (Jayakrisham *et al.* 1994), DYNAMIT (Ben-Akiva, 2002) and more recently DTASQ (Florian *et al.,* 2001; Mahut *et al.*, 2003; Mahut *et al.*, 2004) are good examples of such approach. And microscopic approaches based on the emulation of traffic flows by moving individual vehicles modeling their behavior by means of car-following, lane changing, gap acceptance and other dynamic models consistent with traffic flow theory. AIMSUN, TSS (2006), Paramics, Quadstone (2003) and VISSIM (2000) are good examples of this approach. Each approach has advantages and disadvantages. Mesoscopic approaches are less data intensive and easier to calibrate and are quite useful to provide a dynamic overall view of large networks and the related information as time

dependent path travel times but they do not provide the detailed information required by Intelligent Transport Systems applications, as for instance the emulation of equipped vehicles and Automatic Vehicle Location applications. Microscopic approaches are data intensive and harder to calibrate but are suitable to simulate in detail Intelligent Transport Systems and specially the applications involved in the dynamic fleet management, on the other hand the improvement in hardware and software has speeded up dramatically the computational performance of microscopic simulators making them available for relatively large applications. Consequently the discussion is not whether one approach is better or more appropriate than other, or if there is a unique approach that can replace satisfactorily all others, but which is the most appropriate use of each approach depending on the objectives of the study.

The main objective of this paper is a Decision Support System to assist analysts in the design and evaluation of real time fleet management in urban areas, based on an ad hoc implementation of the referenced Taniguchi's methodological framework, consequently it requires a computational platform able to emulate traffic dynamics and Intelligent Transport System applications, thus our decision was to use a microscopic simulation approach and we selected AIMSUN, the microscopic traffic simulator that we have developed.

8.2 CONCEPTUAL APPROACH TO A DECISION SUPPORT SYSTEM FOR THE DESIGN AND EVALUATION OF CITY LOGISTIC APPLICATIONS

The proposal of a methodology for the design and evaluation of City Logistic systems has to be complemented by the development of a software system, implementing such methodology, in order to be accessible to practitioner. The system can be conceived as a Decision Support System whose conceptual approach is based in the combination of an Operations Research approach and a Computer Science approach. Methodologically Operations Research works with models that formally represent the systems on which the decisions have to make. Valid models of systems provide the support to answer what if questions on the intended system. The main models behind the what if questions that City Logistic Applications have to afford belong to the domain of the Operations Research (i.e. location models, to determine the optimal design of the Public Logistic Terminals – number of Public Transport Terminals to operate in a given city, etc.- determination of the fleet sizes, routing and scheduling of the vehicles, and so on) therefore, it is quite natural to adopt this point of view to address the design of the intended decision support system. A key question for an

efficient use of the Operations Research models concerns the computing environment into which they are embedded, and the friendliness to build the model and determine, and apply, which is the most appropriate algorithm to find the solutions to the model, solutions that will provide the answers to the what if questions. This means that it is not only a problem of an efficient computational implementation of the algorithm, but also of implementing the algorithm as part of a software structure conceived as a Computer Decision Support System, Turban (1993) and based also on Keenan's suggestions (Keenan, 1998), "GIS techniques can contribute to a broad class of routing problems. Standard GIS software may have some features, such as the provision of shortest path algorithms, which can be used as a part of a routing system. However, a general purpose GIS will not allow the decision maker easily interact with the algorithms needed for complex multi-vehicle routing problem. We suggest that a combination of GIS and management science techniques would facilitate decision support for problems with complex path restrictions. The effective combination of GIS and vehicle routing models, to build routing Decision Support Systems, is an area where there are many interesting and relevant research problems that have yet to be fully investigated". In a different context not accounting for the emulation of dynamic conditions in the sense of Taniguchi's framework, Ioannou *et al.* (2002) and Tarantilis *et al.* (2004) have developed similar approaches. Gayialis and Tatsiopoulos (2004), propose a preliminary architecture for decision support systems for vehicle routing and scheduling in presence of information technologies.

The core architecture for the system reported in this paper has been based on an adaptation the conceptual structure proposed by Sprague (1986). This structure, depicted in Figure 8-2, whose preliminary scheme is described in Barcelo (2006a), consists of the following main components:

- A Data Base, to store all the data required by the implied models: locations of logistic centres and customers, capacities of warehouses and depots, transportation costs, operational costs, fleet data, etc.
- A Data Base Management System, for the updating of the information stored in the data base
- A Model Base, containing the family of models and algorithms to solve the related problems, discrete location, network location, vehicle routing, scheduling, etc.
- A Model Base Management System, to update or to modify, add or delete models from the Model Base.
- A Graphic User Interface, GUI, supporting the windows based dialogues to define and update data, select the model suited to the intended problem, apply the corresponding algorithm, visualize the problem and the results, etc.

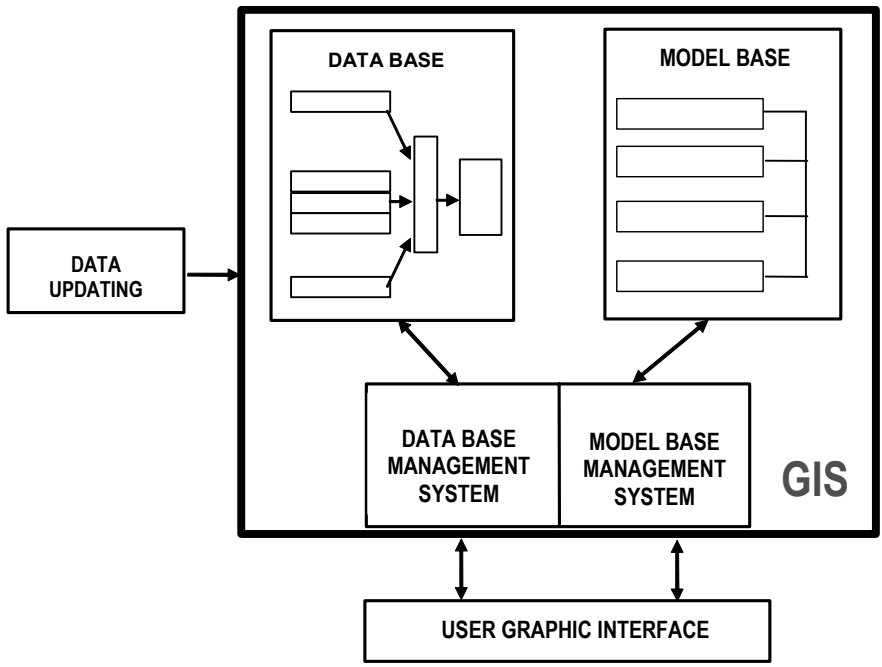


Figure 8-2. Conceptual Structure of a Quantitative Decision Support System.

Taking into account the nature of the problems addressed in City Logistic Application, and their underlying geographic reality, it seems quite natural that the framework in which the GUI should be embedded is that of a Geographic Information System, GIS, or a software platform with the main GIS functions required to support transport applications.

This has been the approach taken in the design and implementation of AIMSUN NG, a software environment consisting of a fully integrated suite of traffic and transportation analysis tools. It can be used for transport planning, microscopic traffic simulation, and demand and traffic data analysis, and provides an integrated platform for both static and dynamic modelling. The conceptual architecture of AIMSUN NG depicted in Figure 8-3 has four main subsystems:

- A set of importers and translators that can import and manipulate GIS data from several sources (ESRI, Tele Atlas, NAVTEQ, etc.). It reads CAD data and bitmaps to simplify editing and model building. It can translate data from other applications (EMME/2, CONTRAM, CUBE and SATURN).

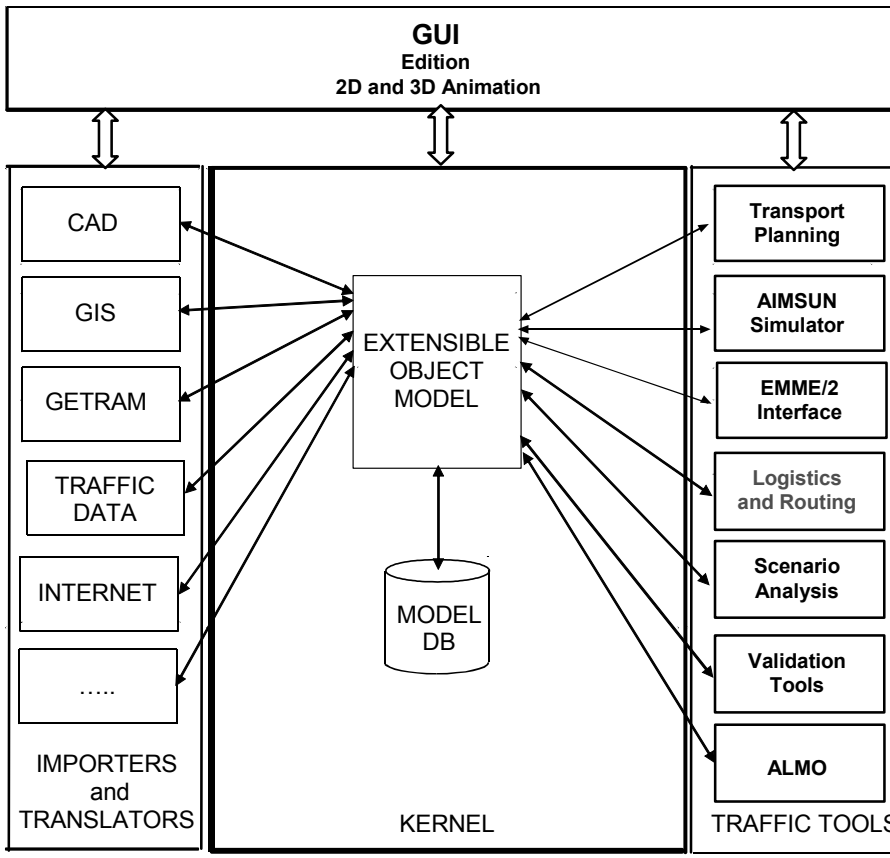


Figure 8-3. Conceptual architecture of AIMSUN NG.

- The Kernel, consisting of an Extensible Object Model and a Model Data Base shared by all traffic and transport analysis models, including the logistics applications, implemented as plug-ins in AIMSUN NG as depicted in Figure 8-3
- A set of traffic and transportation tools to support various transport analysis approaches, including, among others:
  - A Transport Planning component with:
    - A Static User Equilibrium model, and the corresponding Frank and Wolfe algorithm for transport planning analysis
    - A Demand Analysis component to support all the operations necessary for the calculations with the Origin to Destination matrices as required by the analysis of the demand in transport planning, and to provide a computational platform for the

manipulation of the Origin to Destination matrices to generate the inputs to the microscopic simulation, that is:

- Edition and manipulation of O/D matrices
- Matrix balancing and growth factor models
- Matrix adjustment from link flow counts
- Generation and adjustment of traversal OD matrices for local analysis of large networks

- The AIMSUN microscopic traffic simulator to analyze the dynamic behaviour of traffic flows
- An interface with EMME/2 to replace the transport planning component by the EMME/2 planning software
- A Logistics and Routing component to support:
  - Locational Analysis for decisions on locations of warehouses, logistic centres, hubs, etc.
  - Vehicle Routing and Scheduling models to make decisions on fleet operations
  - **The Dynamic Fleet Management component object of this paper**

- A Graphic User Interface, with the editors for model building and manipulation, and the 2D and 3D animation capabilities for results presentation.

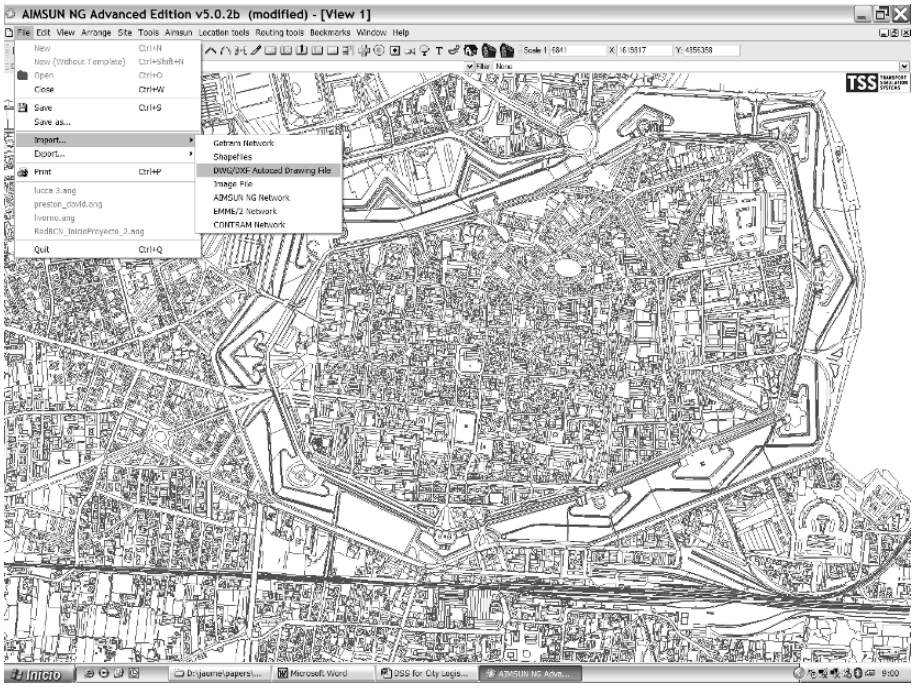


Figure 8-4. Dialogue to import a CAD file into AIMSUN NG and the imported digital map (City of Lucca).

Taking into account the nature of the problems addressed in City Logistic Applications, and their underlying geographic reality, the framework in which the GUI has been embedded is that of a software platform with the main GIS functions required to support transport applications, which imports the digital map of the urban area, and generates automatically the graph its road network to generate the input for the Network Location and Vehicle Routing models. AIMSUN NG becomes in this way a realization of the proposed software platform, whose conceptual architecture is illustrated in Figure 8-2. Figures 8-4 and 8-5 illustrate these GIS like functional capabilities of AIMSUN NG, Figure 8-4 depicts the dialogue to import a CAD file with the digital map of a city, and Figure 8-5 the corresponding dialogue and map in the case of a shape file.

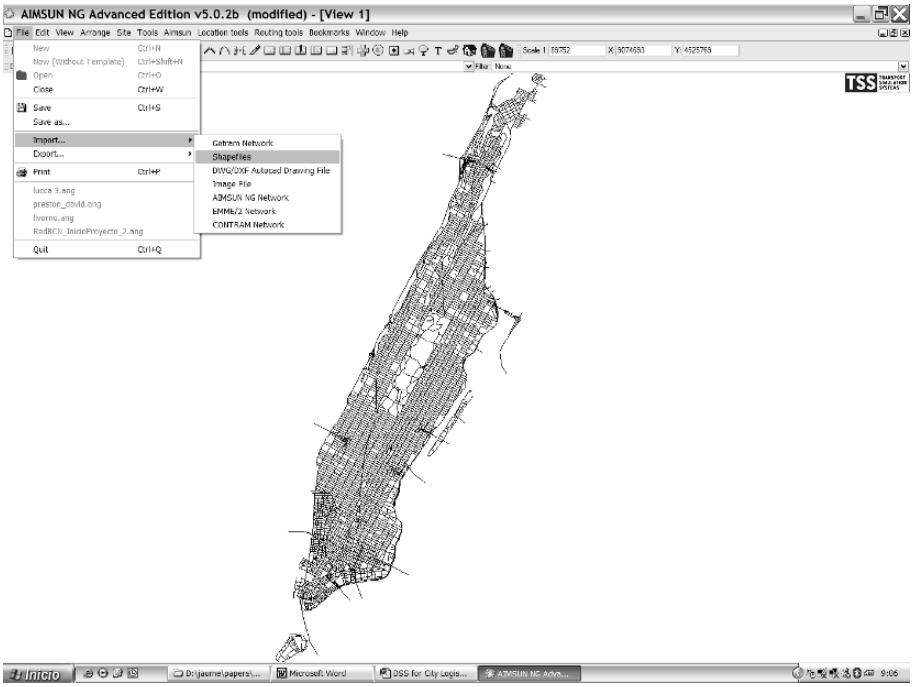


Figure 8-5. Dialogue to import a shape file into AIMSUN NG and the imported digital map (Manhattan).

8.3 AIMSUN MICROSCOPIC TRAFFIC SIMULATOR AND CITY LOGISTICS

In both approaches, for the evaluation of a generic City Logistics application as well as for the evaluation of real time fleet management applications the core models are Vehicle Routing Models able to interact with dynamic simulation models. The simulator used in our project has been AIMSUN (**A**dvanced **I**nteractive **M**icroscopic **S**imulator for **U**rban and **N**on-Urban **N**etworks), in its new version, AIMSUN 5.0, embedded as a component into AIMSUN NG, TSS 2006, as Figure 8-3 shows. AIMSUN captures in very detail the time variability of traffic conditions, accepting as input time sliced Origin-Destination trip matrices. At each time slice, the corresponding number of vehicles starts their trip from their origins to their destinations along the available paths on the network. The paths can be fixed, or traffic conditions dependent and thus timely recomputed, according to a variety of user controlled design factors. At each time slice vehicles are assigned to the available paths according to route choice models. AIMSUN can also account in a very detailed way for junction modeling and the control logic governing the traffic lights at junctions, fixed control plans as well as adaptive real-time control, or pre-emptive signals giving priority to public transport. AIMSUN distinguishes between vehicle classes, and vehicle types within each class, time sliced Origin-destination matrices can be defined by vehicle types provided such detailed information is available. The routes, fixed or time dependent, and the route choice models can also be vehicle type dependent. Dynamically guided vehicles can be allowed to dynamically change the route en route according to the available information.

As a consequence of the ability to reproduce realistically traffic flows on a network by emulating individual vehicles a microscopic simulator can generate many types of dynamic information for any component of the model, including obviously the individual vehicles. The simulator is able to generate information on the time dependent link travel times, relevant in the case of the real time fleet management applications. The graphics in the window in Figure 8-6 depicts the time evolution of the link travel time for the highlighted link in the model, a crucial input to define the link costs for the Dynamic Vehicle Routing Models.

Being based on emulating the movement of individual vehicles through the network a natural function of a proper traffic microscopic simulator is that of tracking individual vehicles, emulating in this way the monitoring of fleet vehicles in a real time fleet management system. The figure 8-7 depicts an example of following a vehicle during the simulation and gathering dynamic data (i.e. current position, previous position, current speed, previous speed, etc.)

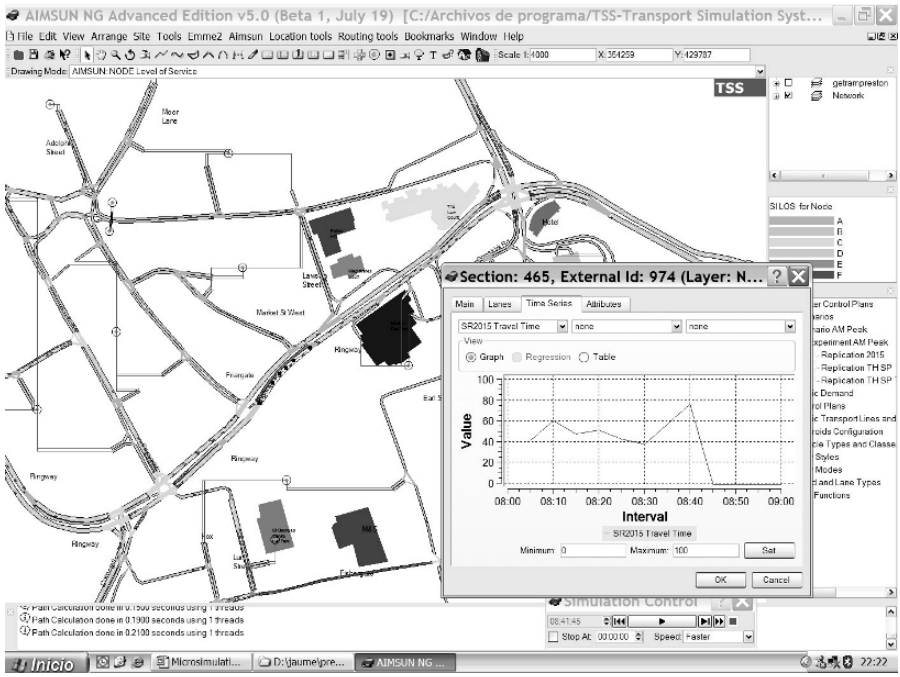


Figure 8-6. Time dependent Link Travel Times on a street section.

while following the vehicle, in a similar way as the data that in real life a vehicle equipped with GPS, or a navigation system, could provide. That is, the dynamic simulation emulates the Floating Car Data (FCD) processes. Other information generated by the microscopic simulator, available at anytime, relevant for the implementation of the real time fleet management decisions, is the identification of the route assigned to a particular vehicle. Route that can be dynamically changed according to specific decision rules. The Figure 8-7 also depicts the route for a selected vehicle and the associated information on route length, route travel time, and route cost when a general cost concept is used.

AIMSUN NG is able to load detection data (both offline from historical databases or online in real time) to be used by any of its components the planner, the visualization module or the AIMSUN microscopic simulator which tracks individually the fleet vehicles, emulating in this way the monitoring of fleet vehicles in a real time fleet management system, gathering dynamic data (i.e. current position, previous position, current speed, previous speed, etc.) while following the vehicle, in a similar way as the data that in real life an equipped vehicle could provide. This is the information required by a Dynamic Router and Scheduler to determine which vehicle will be assigned the new service and which will be the new route for the selected vehicle.

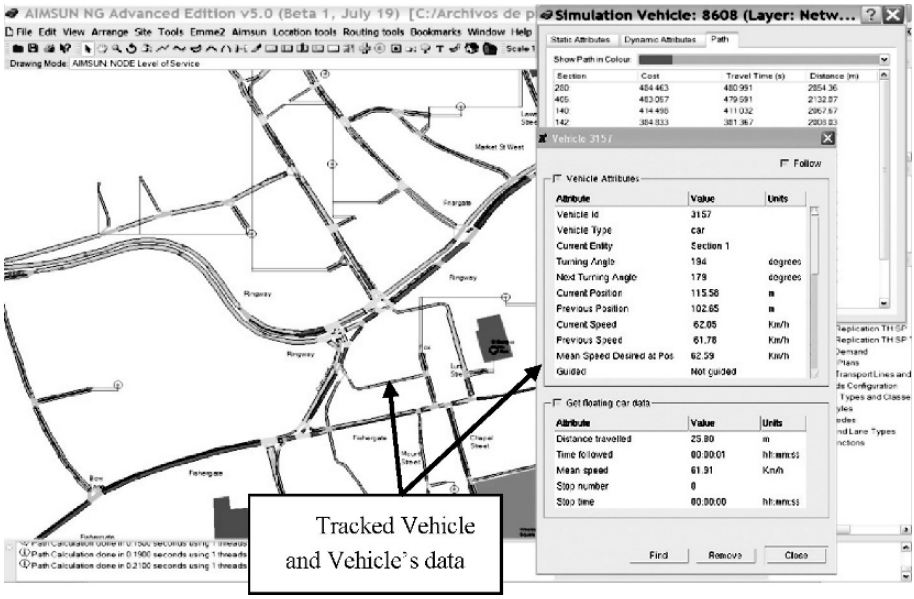


Figure 8-7. Identification of vehicle's paths.

8.4 VEHICLE ROUTING AND SCHEDULING MODELS AND CITY LOGISTICS

According to Taniguchi *et al.* (2001), Vehicle Routing and Scheduling Models provide the core techniques for modelling City Logistics. Once the facilities, or the City Logistics Centres have been located, and the demand nodes have been allocated to each facility, the next step is to decide the efficient use of the fleet of vehicles that must make a number of stops to pick up and/or deliver passengers or products. The problem requires the specification of which customers should be serviced by each vehicle, and in what order, so as to minimize the total cost subject to a variety of constraints such as vehicle capacities, delivery time restrictions, etc. For a comprehensive State-of-the-Art of Vehicle Routing and Scheduling Models see Toth and Vigo (2002).

However, a main difference between vehicle routing algorithms, as they appear in the technical and scientific journals, and real problems in the context of City Logistics, is that the assumption on the symmetry of the costs $c_{ij}=c_{ji}$ no longer holds. In the context of City Logistics Systems it is clear that travels between demand sites, shop retailers, commerce, etc., and facilities, which are the City Logistics Centres, occur only on the street network or, equivalently in terms of the model, on the graph representation of the street network. A more realistic approach for City Logistics applications would be

to obtain travel distances by applying a shortest route algorithm to a computerized model of the road network system, namely when travel times become the relevant data instead of distances. Figure 8-8 depicts an example of a small network with a full representation of its geometry, and its corresponding translations in terms of a directed graph, which is composed of nodes and links. Note that, for each section, a node is created and there is a link for each turning movement. The cost assigned to each arc could be a function of the travel time of the section plus the travel time of the turning movement.

The software architecture rooted on the common extensible Object Models and a Common Data Base, allows a multi-level representation of the Network supporting the automatic translation of the digital map of the road network to a directed graph representation, in terms of links and nodes, for the Vehicle Routing algorithms, this translation must take explicitly into

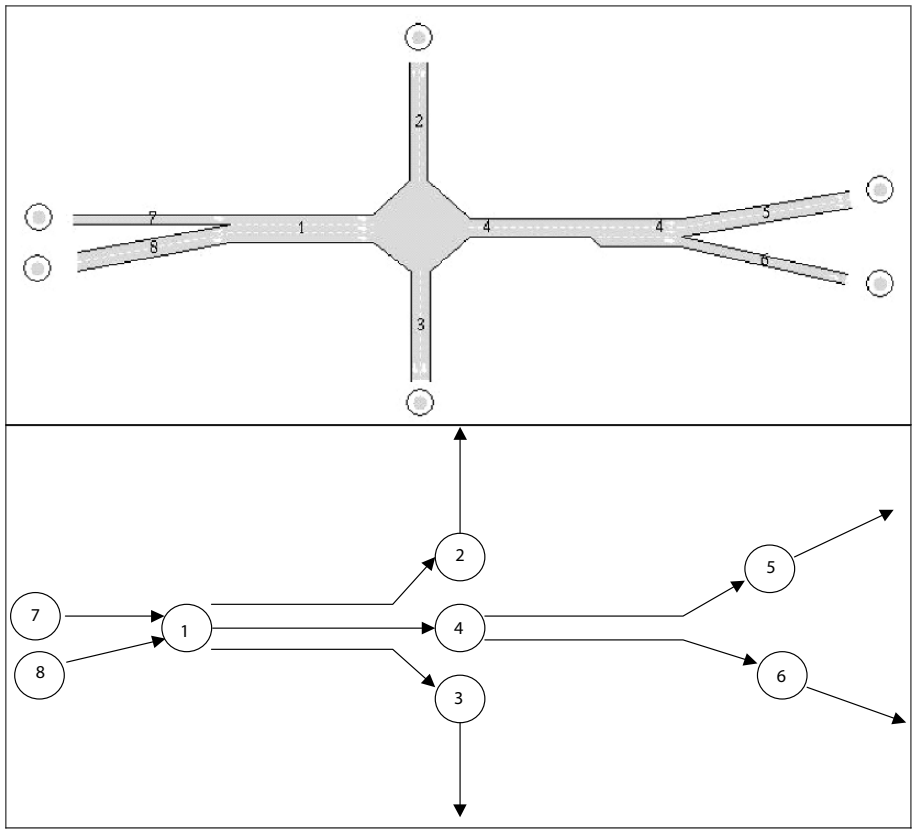


Figure 8-8. Network translation for Vehicle Routing Algorithms.

account turning movements, as they happen in the real network, with their associated costs as resulting from time settings of the traffic control systems at signalized intersections, or delay at intersections governed by give way or stop signals. Figure 8-9 depicts an example of the two level representations for the same network in the AIMSUN NG working area. Figure 8-9 displays on the left the macro level link-node network representation, required by the routing applications, and on the right hand the detailed representation required by other models, including the details of the turning at an intersection.

In the translated graph, the costs are no longer symmetric, as Figure 8-10 illustrates, that is costs are now asymmetric $c_{ij} \neq c_{ji}$. In the example of the Figure 8-10 the digital map of the city of Lucca, one of the test sites in project MEROPE, is depicted. To travel from the Logistic Centre in A, to customer in B, the shortest time travel path is highlighted (Figure 8-10 left), while to travel fro B to A under the same conditions the vehicle should follow a different route (Figure 8-10 right).

This is due to the fact that in a urban environment routes using the streets have to account for one way streets, issues related to penalties at intersections, signalized as well as unsignalized, banned turning movements and/or U-turns, etc. That's the reason why some authors make a distinction between Routing

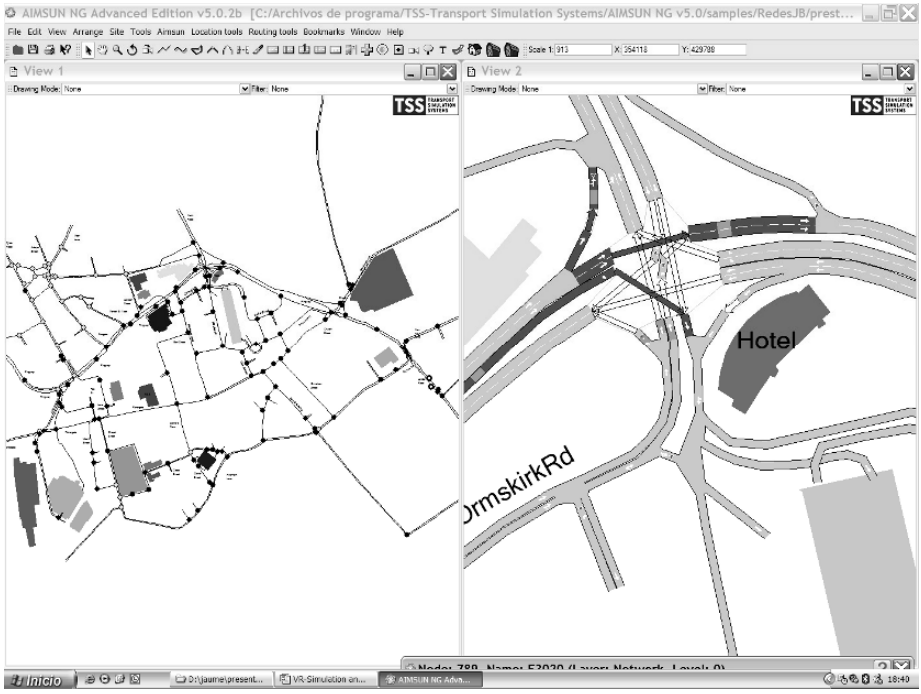


Figure 8-9. Simultaneous multi-level network representation in AIMSUN NG.

Problems and Street Routing Problems (Bodin *et al.,*1983; Bodin *et al.,* 1999; Golden *et al.*, 2002). A relevant aspect of this distinction concerns the concept of link cost for the routing model (a similar comment is also valid for the network location problems in the City Logistics context) and its use in the analytical or heuristic algorithm that will find the solutions to the model.

Assuming in figure 8-11 that the depot is represented by the red square and the customer by the blue one, the cost c_{0i} to travel from the depot, node 0, to the i-th customer, node i, is the cost of the path (highlighted in green in the figure) from the depot to the customer on the urban network. These costs, namely when are expressed in term of travel times, are clearly not symmetric and the underlying graph representing analytically the model is not Euclidean and the triangular property does not hold in it. We shall consequently draw our attention towards the models that deal explicitly with the asymmetry in the travel costs, as the most appropriate for urban routing problems.

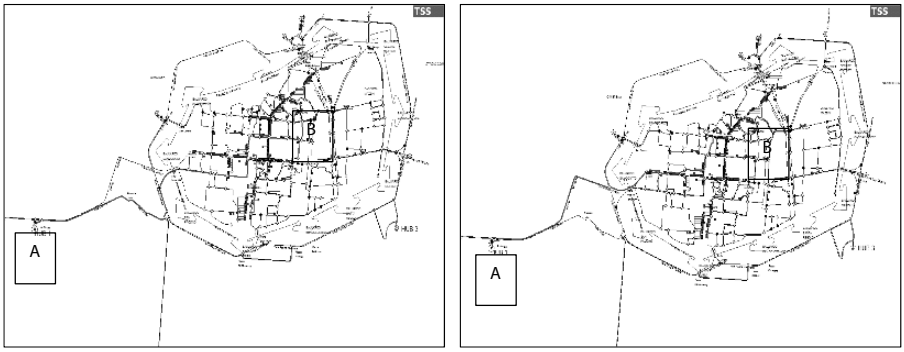


Figure 8-10. Asymmetry of the travel costs in an urban network: from A to B and back.

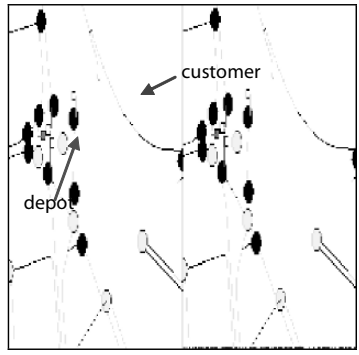


Figure 8-11. Path from the depot to a customer on a street network.

A wide variety of algorithms has been proposed for the Asymmetric Capacitated Vehicle Routing Problem (ACVRP). The algorithmic approaches to solve this problem cover a wide range, from exact algorithms based on branch and bound, branch and cut or lagrangean relaxations, to heuristic and metaheuristic algorithms, among which those based on tabu search, scatter search, genetic and evolutionary algorithms, simulated annealing or ant colonies, appear as the more computationally efficient. For details on algorithms to solve the ACVRP and related models we recommend to see the references Baldacci *et al.* (2004), Cordeau *et al.* (2001), Gendreau *et al.* (1998) or the state of the art review in Toth and Vigo (2002).

8.4.1 Automatic Vehicle Routing Formulation

To assist the analyst in the model building process the proposed Decision Support System translates the digital map of the urban network into a directed graph according to the description in previous section, illustrated by figures 8-8 and 8-9. However this directed graph is not yet the one required by the vehicle routing algorithms. The graph G=(N,A) for VR applications, as depicts Figure 8-12, derived from the Link-Node graph of the road network, consists on the set of nodes N={0,1,2….n} where node 0 is the depot and nodes i, i=1,…,n are the customers, and the set of links A={(i,j)|i,j∈N} corresponds to those feasible paths on the road network connecting node i with node j. The costs c_{ij} associated to links (i,j)∈A in this graph are calculated either as travel times

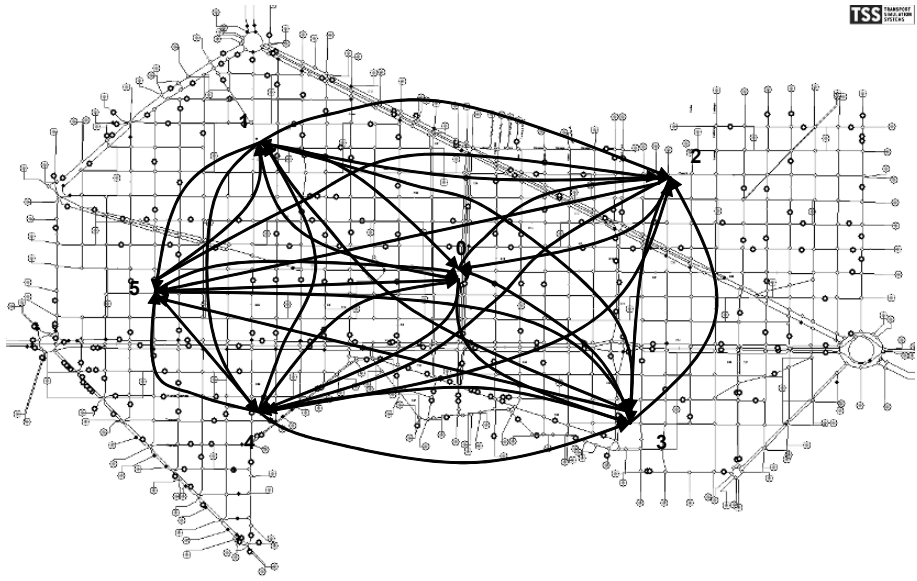


Figure 8-12. Example of graph G=(N,A) for VR applications.

from i to j or as general transportation costs, that is costs calculated as function of travel time, distance, toll prices, and other variables, as corresponds to the street routing problems and has been illustrated in Figure 8-11.

This means that the translation process, as implemented in the Decision Support System based on AIMSUN NG, consists on two steps:

- Translation of the digital map (AIMSUN NG Network model) of the urban network in terms of a link-node network model as a directed graph including the representation as special nodes of depots and customers
- Translation of the link network model in terms of the Vehicle Routing graph whose nodes are depots and customers and whose links are feasible paths in the link-mode directed graph between depot and customers and between customers.

The generic conceptual diagram of an integrated "Routing-Simulation "approach for modelling "City Logistic" schemes in presence of ICT that was illustrated in Figure 8-1, has been implemented in AIMSUN NG in the terms represented by the diagram in Figure 8-13. The process starts by building the AIMSUN NG network model. There are two possibilities: building the model with the assistance of the network editors available in AIMSUN NG on the digital street map imported into the working area (Figure 8-4), or translating automatically the GIS shape files (Figure 8-5) and the associated semantic information into an AIMSUN NG skeleton model. A specific editor and its associated dialogues allow the analyst to complete the model including the other objects of interest as depots and customers and their associated attributes: fleet, capacities, demands, and son on. The AIMSUN NG allows a multilevel network representation as illustrated in Figure 8-8. One of these representations is done in terms of links and nodes as required by the static user equilibrium model, one of the traffic models available in AIMSUN NG (See "Traffic Tools" in Figure 8-3). User equilibrium models (Florian and Hearn, 1995) provide as outcome average link travel times that can be used to compute paths between pairs of points in the network. Alternatively executing a microscopic

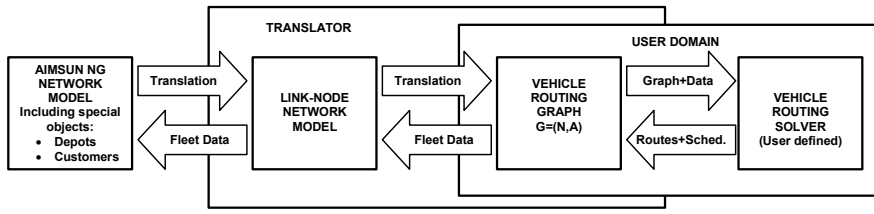


Figure 8-13. Conceptual diagram of an integrated "Routing-Simulation" approach for modelling "City Logistic" schemes in presence of ICT based on AIMSUN NG.

traffic simulation with AIMSUN provides time dependent link travel times, Figure 8-6, to account for a full dynamic context. This link-node network model, the identification of the depot and the customers, and the calculation of the paths connecting them and their associated costs, define the input for the translator that automatically builds the graph G=(N,A) on which the specific vehicle routing problem will be solved in the Vehicle Routing Solver.

The diagram in Figure 8-13 depicts also the flow from the Vehicle Routing Solver to the AIMSUN simulator. Once the fleet routes have been defined in terms of travel times imposed by traffic conditions, the dynamic simulation can continue and fleet vehicle can be tracked during the simulation as illustrated Figure 8-8, completing in this way the methodological scheme proposed by Taniguchi *et al.* (2000).

8.4.2 Dealing with the Appropriate Time Dependent Travel Times

The main interest in combining dynamic traffic simulation with vehicle routing algorithms is to benefit of the emulation of the reality provided by traffic simulator, to supply sound estimates of link and path travel times to the routing algorithms, reproducing in a realistic way the time evolution of travel times on a urban network according to the time evolution of traffic flows. A key question in this approach deals with the reliability and quality of the provided link travel times. The appropriate answer to this question concerns two complementary aspects, the validation of the simulation model and the ability of achieving a dynamic equilibrium with traffic simulation models.

Validation is concerned with determining whether the traffic simulation model is an accurate representation of the traffic system under study; this is a key process to determine the reliability of a traffic simulation model. The possibility of a detailed description is beyond the scope of this paper, for further information the reader can see Barceló and Casas (2004). Validation deals with building the right model and it is established on basis to the comparison analysis between the observed output data from the actual system and the output data provided by the simulation experiments conducted with the computer model. Validation is inherently a statistical analysis process.

The transportation analysis practice has relied since a long time on the user equilibrium paradigm to model drivers' route choices; the dynamic equilibrium paradigm extends this modeling capability to the dynamic case to ensure the consistency of the used routes as identified by a route choice mechanism. Friesz *et al.* (1993), formulated the dynamic user equilibrium model as a generalization of Wardrop's principle (Wardrop, 1952) in terms of what is known as reactive assignment, which can be interpreted in terms

of an approximation to a process by which travelers combine the experienced travel times with conjectures to forecast the temporal variations in flows and travel costs. Florian *et al.* (2001) propose a computational framework for dynamic traffic assignment model consisting of two main components:

1. A method to determining the path dependent flow rates on the paths on the network, and
2. A Dynamic Network Loading method, which determines how these path flows give raise to time-dependent arc volumes, arc travel times and path travel times

We propose a heuristic approach to implement this computational framework (Barceló and Casas, 2006), in which the Dynamic Network Loading mechanism is based on microscopic simulation with AIMSUN, and the Route Choice is performed by a stochastic discrete choice procedure. The computational results presented in Barceló and Casas (2006) verify empirically the dynamic equilibrium assumptions and a consistent estimate of simulated link travel times.

8.5 COMBINING VEHICLE ROUTING WITH AIMSUN SIMULATION

The dynamic information generated by the Microscopic Simulator, described in the previous sections, is the information that will be exchanged between the dynamic traffic simulator and the vehicle routing applications according to the conceptual schemes of Figures 8-1 and 8-13. The Figure 8-14 depicts an example of how the conceptual process described in Figure 8-1 is simulated on basis to the proposed microscopic simulation approach. The various colors identify the initially assigned routes to a set of 5 vehicles and the order in which customers will be served according with the initial schedule. This corresponds to the Initial Operational Plan in Figure 8-1.

The vehicle tracking in the microscopic simulation, as illustrated in Figure 8-14 emulates the real time fleet monitoring in the fleet management system. At time t after the trips have started a new customer calls for an unscheduled service. The simulation process emulates the real time vehicle monitoring, and therefore the positions and availabilities of the fleet vehicles are known. This is the information required "Dynamic Router and Scheduler" in the logic diagram in Figure 8-1, to determine which vehicle will be assigned the new service (vehicles 1 and 2 would be the potential candidates in our example) and which will be the new route for the selected vehicle. Two types of Vehicle Routing Problems have been considered in the prototype Decision Support Systems presented in this paper: a Vehicle Routing

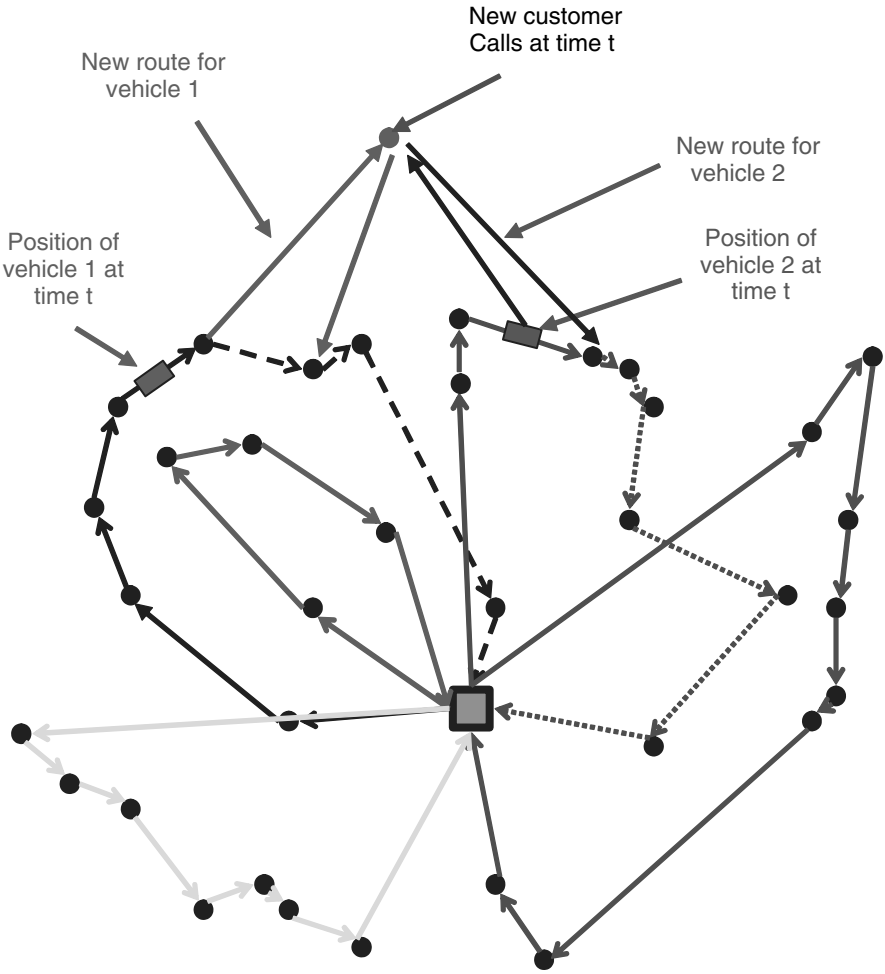


Figure 8-14. Dynamic vehicle rerouting in a real-time fleet management system.

Problem with Time Windows (VRPTW) and a Pickup and Delivery Problem with Time Windows (PDPTW). For the (VRPTW) we have adapted the unified tabu search heuristic proposed by Cordeau *et al*. (2001), a local search meta-heuristic that explores the solution space by moving at each iteration from the current solution s to the best solution in its neighbourhood N(s), including anti-cycling rules to prevent deterioration of the solution, allowing to explore infeasible solutions during the search, and using diversification mechanisms to help the search process to explore a broad portion of the solution space. In the adaptation we have taken explicitly into account the time dependent link travel times along lines inspired in Ichoua *et al.* (2003).

The mechanism described in section 8.4.2 provides a data base with the link travel times for each time interval within the simulation horizon, that means that if the vehicle servicing client i departs at time d_i after completing the service to service customer j, at this departure time the travel time from i to j will be $t_{ij}^{d_i} \neq t_{ij}^{d_j}$ as illustrated in figure 8-15, and this travel time will be different if the vehicle departs a t a different time.

This time dependence of travel times implies that the shortest path problems described in the previous sections have to be time dependent according to the approaches proposed by Chabini (1997) and Ziliaskopoulos and Mahmassani (1993).

For the (PDPTW) the heuristic that we have taken into account in our experiments is a metaheuristic proposed by Li and Lim (2001), which is a Tabu embedded simulated annealing algorithm and has proven its effectiveness according to practical applications of the dial-a-ride problem with time windows.

Depending on the type of routing problem, the re-assignment could be based on the diversion and waiting strategies, proposed by Ichoua *et al.* (2000) and Mitrovic-Minic and Laporte (2004a and 2004b), with the difference with respect to the original algorithms that now the travel times, current and forecasted can be provided by the microscopic simulator.

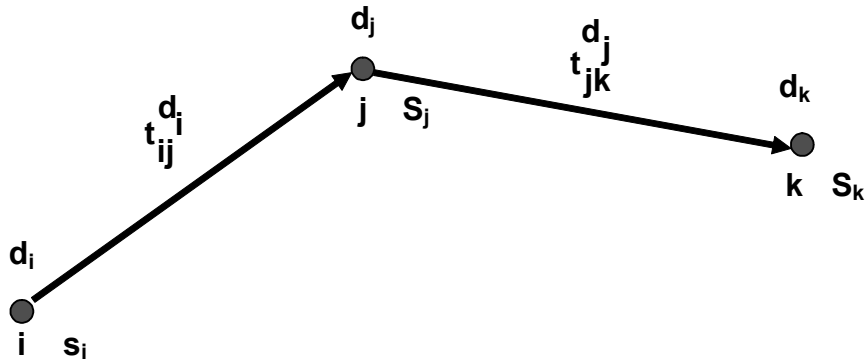


Figure 8-15. Dealing with time dependent travel times, di departure time from client i, si service time for client i travel time from i to j when departing at time di from client i to service client j.

8.6 TWO CASE STUDIES

As mentioned in the introduction the proposed approach was tested in the European Project MEROPE of Program INTERREG IIIB. Two sites were selected for testing purposes, the cities of Lucca and Piacenza in Italy. A summary of the results is presented in this section (for detailed information visit the web page in www.merope.net).

The prototype of the Decision Support System tested in the mentioned projects includes the functions to define the problem in terms of

- The locations and attributes of depots, warehouses or City Logistics Centres: coordinates, capacity, fleets operating from the depot (number of vehicles, vehicle's capacities, etc.)
- The locations and attributes of customers: coordinates, identities, demands, etc.

The automatic formulation of the street routing problems as described conceptually in Figure 8-13.

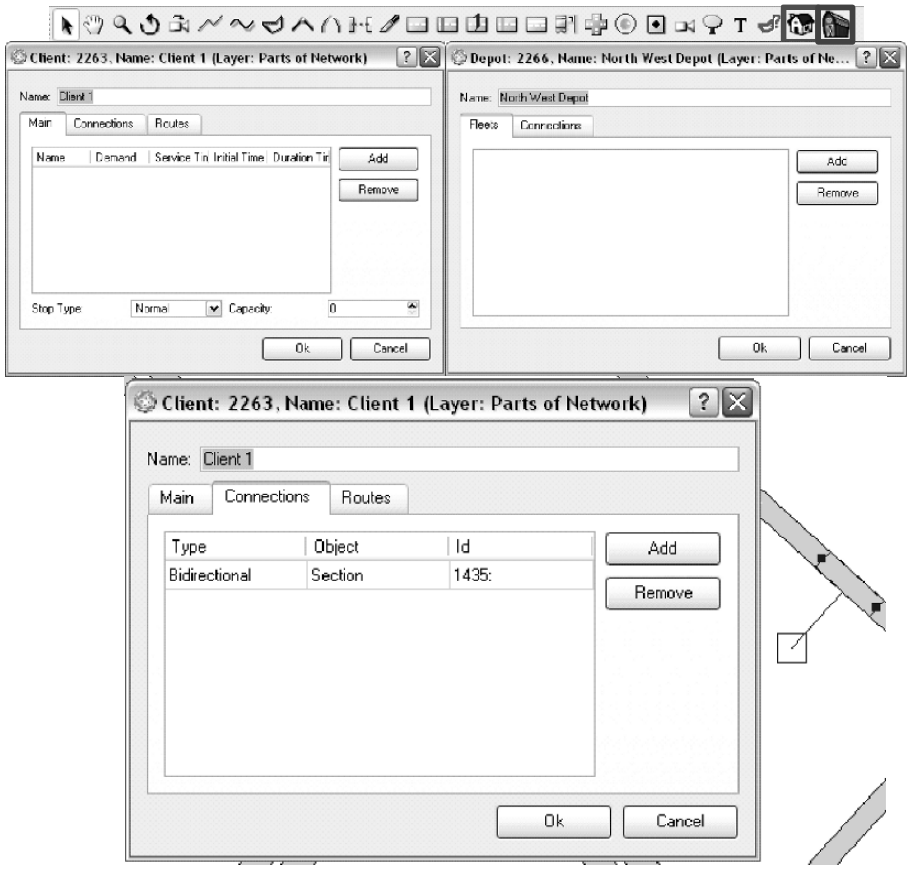


Figure 8-16. Window dialogues and graphic editors to define and locate depots and customers.

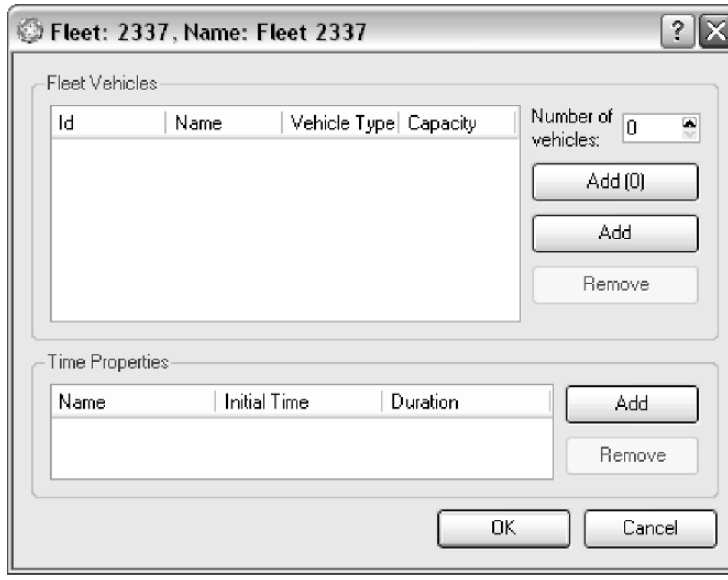


Figure 8-17. Dialogues to define fleet characteristics.

- Access to various definitions of link and route costs in terms of: Distances, Travel times (Static average travel times provided by the user equilibrium traffic assignment in AIMSUN Planner; Dynamic, time dependent costs varying according to changes in traffic conditions, congestions and so on, provided by microscopic simulation with AIMSUN), general cost functions in terms of distances, travel times, toll prices and in general of any other numerical attribute associated to the links.

A preliminary library of Default Vehicle Routing models and algorithms consisting of

- Asymmetric Traveling Salesman, Glover *et al.* (2001), to compute routes for vehicles with a pre-assigned subset of customers to service based on the selected costs.
- Capacitated Vehicle Routing Problems solved by the Christofides, Mingozzi and Toth heuristic for the Asymmetric Vehicle Routing Problem, Toth and Vigo (2002), when time constraints are not active.
- Solution for each depot and its associated fleet by means of the unified tabu search heuristic for vehicle routing problems with time windows of Cordeau, Laporte and Mercier (2001), when time constraints are active, using travel times as provided by AIMSUN according to the above alternatives.

Other Vehicle Routing Models can be plug-in into the software platform acceding to the automatically built Vehicle Routing Graph. The solution can be visualized by the GUI.

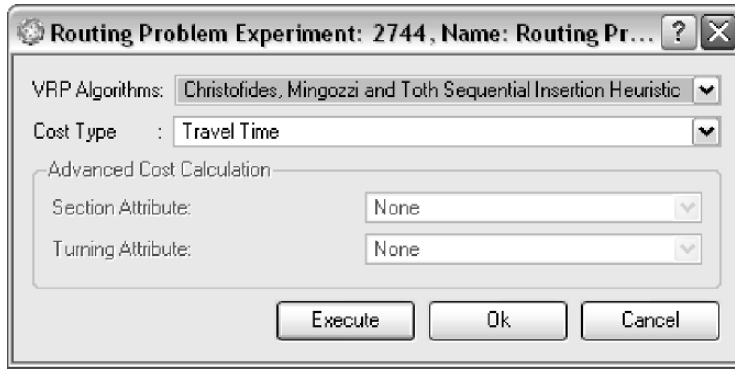


Figure 8-18. Dialogue to select the Vehicle Routing Algorithm for the defined problem

8.6.1 The Lucca Case

The project scope was restricted to the historical centre of the city where the Municipal authorities plan to restrict the distribution of goods to fleets operating from logistic transit points. Four candidate Transit Point locations were proposed by the authorities. Customer's locations were provided by a georeferenced customer's database that used the digital map of the city.

The data provided by the Municipality of Lucca characterized the vehicle fleet and the customers' demand in the following terms

- A fleet of 18 vehicles of capacity 6 cubic metres each
- A total demand of 500 cubic metres per day to be serviced from the selected Transit Point
- The total number of clients is 308, and
- Clients have an average demand of 1 or 2 cubic metres each.

The four candidate locations for Transit Points have been considered. The Figure 8-19 depicts the model without the digital background providing a schematic view of the street network model with the locations of the customers and the facilities (Transit Points). This network model is the one that translated in terms of a directed graph to calculate the routes across the street network, taking into account the special characteristics of graphs for urban applications described in Section 8.3.

Figure 8-20 depicts an example of the solution where the routes from Transit point 1 for three of the vehicles are highlighted. The route lengths, and travel times are the numerical output of the models that are used in the decision making process.

Figure 8-19. AIMSUN NG Model of Lucca with customers and Transit Points (Hubs).

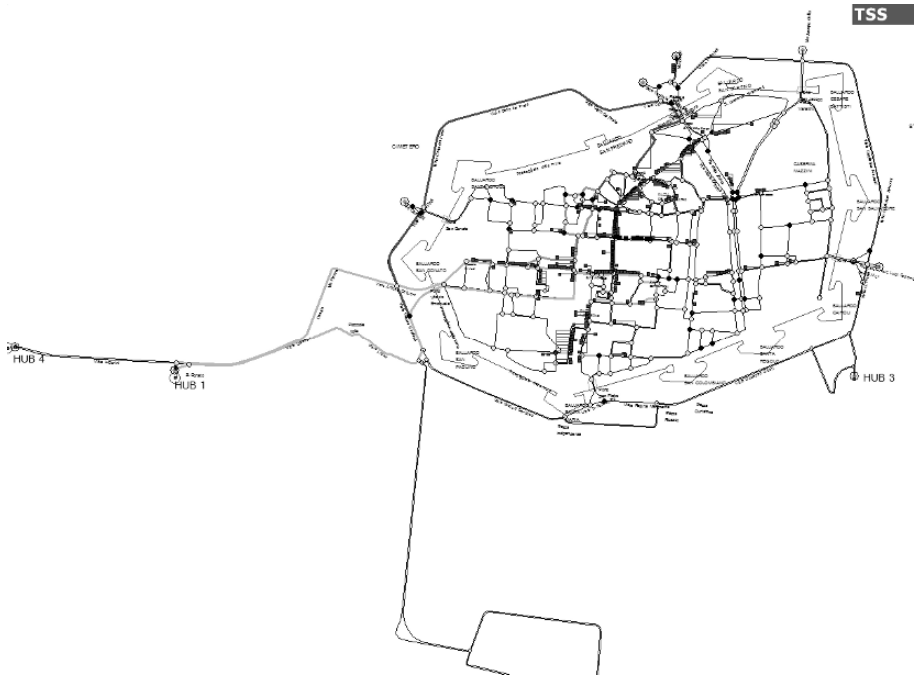


Figure 8-20. Lucca model, routes of tree vehicles from Transit Point HUB 1.

8.6.2 Piacenza Case

The main difference with respect to Lucca was the municipal decision to habilitate special loading-unloading points from where customers have to be serviced. That means that the vehicle routes do not visit directly the customer locations but the loading-unloading points.

Figure 8-21 depicts the dwg file imported into the working area of the Decision Support System for the Design and Evaluation of City Logistics Applications with the locations of the loading-unloading points" provided by the Municipality of Piacenza.

Based on the available information hypothetical scenarios were built according to the following assumptions:

- The demand for individual customers was considered proportional to the size of the shop, distributed between 1 and 4 cubic metres per day.
- The demand for clusters of customers serviced from the loading-unloading points was estimated as the aggregation of the customers closer to the loading-unloading point.
- The vehicle fleet servicing the customers was assumed to be homogeneous, composed of vehicles with average capacities of 6 cubic meters each.
- The upper bound on the size of the fleet necessary for servicing the total demand with these assumptions was estimated in 35, but some of them must repeat between 2 and 4 times part of the route to service the customers in the cluster serviced from the associated loading-unloading point.

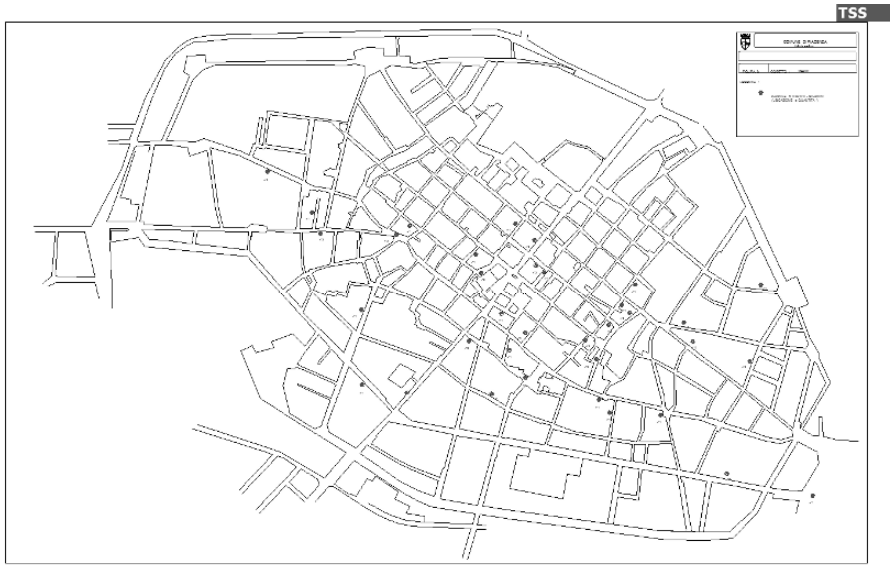


Figure 8-21. Piacenza's loading-unloading points.

Two candidate locations for the Hubs-Transit Points have been considered. Figure 8-22 depicts the routes for three vehicles from Transit Point HUB 1, highlighted in red. The blue squares identify the customers, and squares with a circle inside the loading-unloading points.

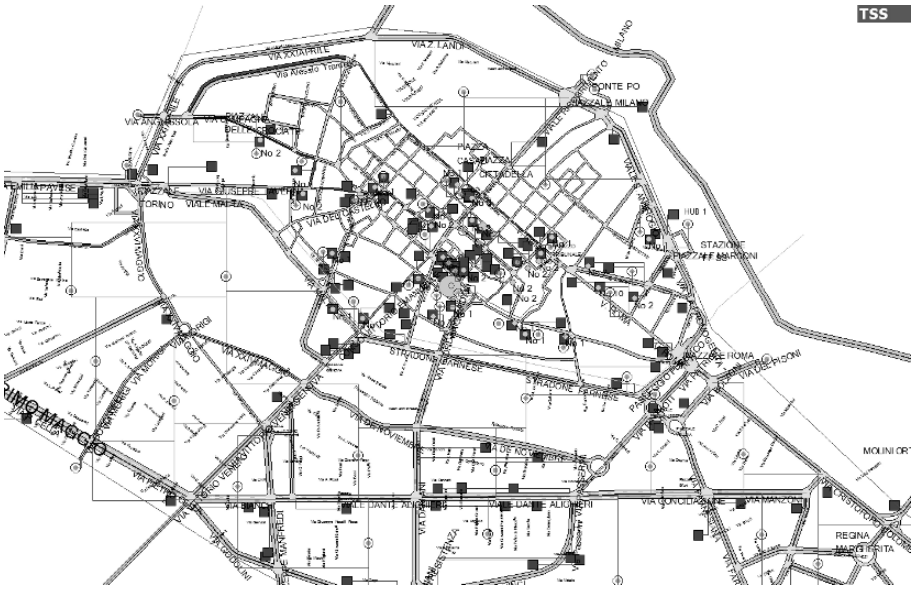


Figure 8-22. Routes for three vehicles from Transit Point HUB 1.

And Figure 8-23 routes from Transit Point Hub 2.

Figure 8-23. Routes for vehicles from Transit Point HUB 2.

Table 8-1 summarizes the results obtained for the loading-unloading scenario operating from HUB2 in terms of the lengths in metres of the routes for each of the 35 vehicles servicing the customers from the loading-unloading points from Transit-Point HUB2, and in terms of travel times, according to the operational hypothesis defined above.

Table 8-1. Simulation results.

	Route/Vehicle	Cost type by distance	Cost typeby time
HUB 2 Loading Unloading	1	5970,200	433,238
	2	5460,480	396,530
	3	5960,280	436,396
	4	5250,700	383,372
	5	5345,580	391,580
	6	5227,180	383,476
	7	5233,480	381,171
	8	5137,840	374,548
	9	5145,320	376,380
	10	5147,240	373,606
	11	5121,600	372,748
	12	5066,220	368,778
	13	4990,780	361,554
	14	5068,580	369,096
	15	5107,160	370,952
	16	5068,500	368,168
	17	5008,500	373,266
	18	5065,280	372,119
	19	5104,740	357,692
	20	4956,020	372,318
	21	5121,800	353,678
	22	4888,060	353,678
	23	4888,060	348,494
	24	4828,260	349,682
	25	4831,220	345,054
	26	4790,140	338,188
	27	4697,720	372,617
	28	5040,800	352,972
	29	5145,740	340,870
	30	4903,040	390,544
	31	4708,160	339,292
	32	5358,360	379,074

Table 8-2. Summary of Piacenza Results.

HUB	by distance	by time
HUB 1 carico-scarico	28263,186	13972,171
HUB 2 carico-scarico	177938,200	11881,131

A similar table summarizes the results for scenarios operating from Transit-Point HUB1. Table 8-2 summarizes the results obtained for the two scenarios and the two variants per scenario using as quantitative evaluation indices the total lengths of the routes of the service vehicles to visit all customers and the total travel times spent by the vehicles to visit either the customers or the loading-unloading points depending on the variant analysed. These results show that, in absence of other criteria, the candidate location 1 for Hub-Transit Point 2, is the best in terms total distance travelled by all vehicles but in terms of average travel times the candidate location 2 is the one resulting in less total travel time.

8.7 CONCLUSIONS

This paper describes the architecture of a Decision Support System to assist the analysts in the design and evaluation of City Logistics applications prior to their implementation. The system is based on an ad hoc implementation of a computational framework based on the methodological concepts of Taniguchi *et al.* (2001). The selected platform is implemented on software that combines the main GIS functions related to transportation models, with the capability to build traffic simulation models and vehicle routing models. The type of fleet management applications addressed in the paper require the ability to reproduce the dynamics of urban traffic and the emulation of fleet management functions as routing and scheduling and Automatic Vehicle Location. This is achieved in the paper resorting to the use of AIMSUN microscopic simulator and its implementation on the AIMSUN NG software environment that provides the required functionalities. The software architecture of the system allows the analyst to replace the default vehicle routing and scheduling algorithms by user defined algorithms. Two examples of application of the Decision Support System to the cities of Lucca and Piacenza are presented to illustrate how the system works. This paper can be considered as a contribution to generalize the seminal ideas of the framework proposed by Taniguchi and opens the path to more complex systems.

8.8 ACKNOWLEDGEMENTS

This research has been partially supported by Spanish DGICYT grant number TIC2000-1750-C06-03.The prototype has been developed as part of the European Project MEROPE, Programme INTERREG IIIB, MEDOC, Axe 3, Measure 4, CODE: 2002-02-3.4-I-091. The new dynamic developments are supported by Spanish DGICYT grant number SADERYL-2 (TIC2003-05982-C05-04)

REFERENCES

Baldacci, R., Mingozzi, A. and Hadjiconstantinou, E., 2004. An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation, *Operations Research*, 52(5), 723-738.

Barcelo, J., 2006a. An overview of models to assist in the design and evaluation of City Logistics projects, in "Systems and Advanced Solutions for City Logistics", Eds. G. Ambrosino, M. Boero, J.D. Nelson and M. Romanazzo, Published by ENEA (Italian National Agency for New Technologies), to appear 2006.

Barceló, J. and Casas, J., 2004. Methodological Notes on the Calibration and Validation of Microscopic Traffic Simulation Models, Paper 04-4975, Transportation Research Board 2004 Annual Meeting, Washington, D.C.

Barceló J. and Casas, J., 2006. Stochastic heuristic dynamic assignment based on AIMSUN Microscopic traffic simulator, Paper #06-3107 presented at 85th Transportation Research Board 2006 Annual Meeting, to appear in Transportation Research Records.

Ben-Akiva M., Bierlaire M., Koutsopoulos H. N. and Mishalani R., 2002, Real-time simulation of traffic demand-supply interactions within DynaMIT, in M. Gendreau and P. Marcotte (eds), Transportation and network analysis: current trends. Miscellanea in honour of Michael Florian, Kluwer Academic Publishers, Boston/Dordrecht/London.

Bodin, L.D., Golden, B.L., Assad A. A. and Ball, M.O., 1983. Routing and Scheduling of Vehicles and Crews: The State of the Art, *Computers and Operation Research* Special Issue, 10, 69-211.

Bodin, L., Maniezzo V. and Mingozzi, A., 1999. Street Routing and Scheduling Problems, in Handbook of Transportation Science, Edited by R. W. Hall, Kluwer.

Chabini, I., 1997. A new algorithm for shortest path in discrete dynamic networks. In Proceedings of the 8th IFAC Symposium on Transportation Systems, Chania, Greece, pp. 551-556.

CLM, 2001, Definition for Logistic, Council of Logistics Management, http://www.clm1.org, Oak Brook.

Cordeau, J-F., Laporte G. and Mercier, A., 2001. A Unified Tabu Search heuristic for Vehicle Routing Problems with Time Windows, *Journal of the Operational Research Society*, 52, 928-936.

Cordeau, J.F. and Laporte, G., 2003. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research* 37B: 579-594.

Florian M. and D. Hearn, 1995. Network Equilibrium Models and Algorithms, Chapter 6 in: M.O. Ball *et al.*, Eds., Handbooks in Operations Research and Management Science, Vol.8, Elsevier Science B.V.

Florian, M., Mahut, M. and Tremblay, N. 2001. A Hybrid Optimization-Mesoscopic Simulation Dynamic Traffic Assignment Model, Proceedings of the 2001 IEEE Intelligent Transport Systems Conference, Oakland, pp. 120-123.

Friesz, T., Bernstein, D., Smith, T., Tobin, R. and Wie, B., 1993. A variational inequality formulation of the dynamic network user equilibrium proble*m. Operations Researc*h, 41,179-191.

Gayialis S.P. and Tatsiopoulos, I.P., 2004. Design of an IT-Driven decision support system for vehicle routing and scheduling. *European Journal of Operational Research*, 152, 382-298.

Gendreau, M., Hertz A. and Laporte, G., 1992. New Insertion and Postoptimization Procedures for the Traveling Salesman Problem, *Operations Research* 40, 1086-1094.

Gendreau, M., Hertz, A., Laporte G. and M. Stan, M., 1998 A generalized insertion heuristic for the traveling salesman problem with time windows, *Operations Research* Vol. 43(3), 330-335.

Glover, F., Gutin, G., Yeo A., and Zverovich, A., 2001. Construction heuristics for the asymmetric TSP. *European Journal of Operations Research* 129, 555-568.

Golden, B.L., Assad A.A. and Wasil, E.A. 2002. Routing Vehicles in the Real World, in The Vehicle Routing Problem, Edited by P. Toth and D. Vigo, SIAM Monographs on Discrete Mathematics and Applications.

Jayakrisham R., Mahmassani H.S. and Yu T.Y., 1994, An Evaluation Tool for Advanced Traffic Information and Management Systems in Urban Networks.Transportation Research C, 2C (3), pp. 129-147.

Ioannou, G., Kritikos, M.N. and Prastacos, G.P., 2002. Map-Route: a GIS-based decision support system for intra-city vehicle routing with time windows. *Journal of the Operational Research Society*, 53, 842-854.

Ichoua, S., Gendreau, M. and Potvin, J.V., 2000. Diversion Issues in Real-Time Vehicle Dispatching, *Transportation Science*, 34(4), 426-438.

Ichoua, S., Gendreau, M. and Potvin, J.-Y., 2003, Vehicle Dispatching with Time-Dependent Travel Times, *European Journal of Operations Research* 144, 370-396.

Inner Urban Freight Transport and City Logistics, 2003, http://www.eu-portal.net.

Jaw, J.J., Odoni, A.R., Psaraftis H.N. and Wilson, N.H.M., 1986. A heuristic algorithm for the multi vehicle advance request dial-a-ride problem with time windows, *Transportation Research Part B 20*, 243-257.

Keenan, P.B., 1998. Spatial decision support systems for vehicle routing. *Decision Support Systems***,** 22: 65-71.

Kohler, U., 1997. An innovating concept for City Logistics, Proceedings of the 4th World Congress on ITS, Berlin.

Li, H. and Lim, A., 2001. Technical Report by Department of Computer Science, National University of Singapore, http://citeseer.ist.psu.edu/.

Koriath H. and Thetrich, W., 1998. Urban Goods Transport; COST 321 – Final report of the action Office for Official Publications of the EC, Bruxelles, Luxembourg.

Mahut M., Florian M. and Tremblay N., 2003, Traffic Simulation and Dynamic Assignment for Off-line Applcations, presented at the 10th World Congress on Intelligent Transportation Systems, Madrid.

Mahut M., Florian M., Tremblay N., Campbell M., Patman D. and McDaniel Z.M. 2004, Calibration and Application of a Simulation based Dynamic Traffic Assignment Model, Proceedings of the 83rd TRB Annual Meeting.

Mitrovic-Minic, S. and Laporte, G., 2004a. Waiting Strategies for the Dynamic Pickup and Delivery problem with Time Windows, *Transportation Research* Part B, 38, 635-655.

Mitrovic-Minic, S., Krishnamurti, R. and Laporte, G., 2004b. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows, *Transportation Research* Part B 38, 669-685.

Quadstone Limited, 2003, Paramics User Guide Version 4.0, Quadstone Limited, Edinburgh, UK.

Regan, A. C., Mahmassani, H.S. and Jaillet, P., 1997. Dynamic decision making for commercial fleet operations using real-time information, *Transportation Research Record*, 1537.

Regan, A.C., Mahmassani, H.S. and Jaillet, P., 1998. Evaluation of dynamic fleet management systems: Simulation Framework, *Transportation Research Record,* 1645.

Tarantilis, C.D., Diakoulaki, D. and Kiranoudis, C.T., 2004. Combination of GIS and efficient routing algorithms for real life distribution-transportation operations. *European Journal of Operational Research,* 152: 437-453.

Taniguchi, E. and R.E.C.M van der Heijden, 2000. An Evaluation Methodology for City Logistics, *Transport Reviews*, 20(1), 65-90.

Taniguchi, E., Thompson, R.G., Yamada T. and Van Duin, R., 2001. City Logistics: Network Modelling and Intelligent Transport Systems, Pergamon.

Toth, P. and Vigo, D., 1997. Heuristic algorithms for the handicapped person transportation problem, *Transportation Science* 31, 60-71.

Toth, P. and Vigo Eds, D., 2002. The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications.

TSS –Transport Simulation Systems, 2006, AIMSUN NG, User's Manual, http:// www.aimsun.com

Turban, E., 1993. Decision Support and Expert Systems: Management Support Systems, Macmillan Publishing Company, ISBN 0-02-421691-7.

Sprague, R.H. and Watson, H.J., 1986. Decision Support Systems: Putting Theory into Practice, Prentice- Hall.

"VISSIM 3.5, 2000, User Manual", PTV Planung Transport Verkehr AG, Germany.

Wardrop J. G., 1952, Some theoretical aspects of road traffic research, Proceedings, Institution of Civil Engineers II(1), pp. 325-378.

Ziliaskopoulos, A.K. and Mahmassani, H.S., 1993. Time-dependent shortest path algorithm for real-time intelligent vehicle-highway systems. *Transportation Research Record* 1408, pp.94-100.

Chapter 9

DYNAMIC MANAGEMENT OF A DELAYED DELIVERY VEHICLE IN A CITY LOGISTICS ENVIRONMENT

V. Zeimpekis[1], I. Minis[2] , K. Mamassis[2] and G.M. Giaglis[1]
[1]Department of Management Science and Technology, Athens University of Economics and Business, 47A Evelpidon & 33 Lefkados St., GR-11362, Athens, Greece, {vzeimp, giaglis}@aueb.gr; [2]Department of Financial Management Engineering, University of the Aegean, 31 Fostini St., GR-82100, Chios, Greece, {k.mamassis, i.minis}@fme.aegean.gr

Abstract: Distribution schedules designed *a priori* may not cope adequately with unexpected events that occur during the plan execution, such as adverse traffic conditions or vehicle failures. This limitation may lead to delays, higher costs, and inferior customer service. This chapter presents the design and implementation of a real-time fleet management system that handles such unexpected events during urban freight distribution. The system monitors delivery vehicles, detects deviations from the distribution plan using dynamic travel time prediction, and adjusts the schedule accordingly by suggesting effective rerouting interventions. The system has been tested in a Greek 3PL operator and the results show significant improvements in customer service.

Keywords: Urban freight distribution; Real-time vehicle routing; dynamic incident handling.

9.1 DISTRIBUTION IN A CITY LOGISTICS ENVIRONMENT

Freight distribution accounts for a significant portion of the total costs of logistics (Lambert, 1998). Techniques to minimize distribution costs typically focus on the development of near-optimal plans using various types of effective vehicle routing algorithms (Ballou, 2004). Urban distribution, however, is more susceptible to unexpected costs and delays that arise during

the execution of the delivery plan due to unforeseen adverse conditions, such as traffic delays, vehicle breakdowns, road works, customer depot overload, and so on (Regan *et al.,* 1997; Fleischmann, 2004; Zeimpekis, 2005). Table 9-1 presents a typical classification of incidents and their effects on delivery. It is emphasized that the use of an initial distribution plan, although necessary, is by no means sufficient to address these unexpected events that may have adverse effects on the performance of the delivery system.

Existing work that deals with the dynamics of the distribution process includes mainly algorithmic approaches that focus on solving the Dynamic Vehicle Routing Problem (DVRP) (Psaraftis, 1995; Ichoua *et al*., 2003; Haghani, 2004). The dynamics usually stem from client orders that arrive during the execution of the delivery plan and need to be assigned to working vehicles.

Several fleet management systems have been proposed in the literature focusing on this problem. Goetschalckx (1998), Powell (1990), Savelsbergh and Sol (1997), Slater (2002), as well as Gans and Ryzing (1999) proposed decision support systems that cope only with new customer requests arriving dynamically. These systems treat travel times either as constant or use simple procedures to adjust them according to the time of day. Kim *et al*. (2003) introduced real-time traffic information in such systems. Ichoua *et al*. (2003) presented a real-time fleet management model based on time-dependent travel speeds. An experimental evaluation of the proposed model showed that the time-dependent model provides substantial improvements over a model based on fixed travel times. Fleischmann *et al.,* (2004) presents a dynamic routing system that dispatches a fleet of vehicles according to customer orders arriving at random during the planning period. The system also uses online information of travel times from a traffic management centre. Finally, Hanghani and Jung (2004) present a systemic approach to address the

Table 9-1. Dynamic incidents in urban freight distributions.

Source of incident	Incident	Effect in delivery
Road Infrastructure & Environment	Traffic congestion, adverse weather conditions, road construction, flea markets, protests	Increased vehicle travel time
Clients	No available unloading area, problems with the delivered products (e.g. wrong order)	Longer client service times
	New client request (delivery or pickup), amount of request	Vehicle re-routing in real-time/ no service
Delivery Vehicle	Car accident, mechanical failure	Customer Service interception

dynamic vehicle routing problem with time-dependent travel times. They present a genetic algorithm to solve a pick-up or delivery vehicle routing problem with soft time windows and real-time service requests. Dynamic travel times are obtained by on-board terminals.

This paper addresses a different problem of dynamic fleet management, in which the distribution plan needs to be adjusted in real-time to accommodate changes in uncontrollable parameters of the delivery environment (see Table 9-1). Specifically, we consider a traditional distribution setting, in which, each vehicle distributes a pre-specified set of orders along a preplanned route. The latter may be the result of a typical routing process, manual or algorithmic. The execution of this plan, however, may be impeded by various unexpected events, such as traffic congestion, parking space unavailability at a customer site, etc, which may result in significant delays. In this case, the vehicle may no longer be capable to complete its entire route, and, thus, rerouting becomes necessary in order to provide the best customer service possible under these circumstances. The mathematical model related to this problem resembles the so called Orienteering Problem (OP), a variation of the Traveling Salesman Problem (TSP). We propose a real-time fleet management system that continuously monitors the execution of the initial plan, detects significant deviations that require rerouting, solves the related optimization routing problem and transmits the revised plan to the vehicle, all in real time.

The remainder of the Chapter is organised as follows. Section 2 presents significant requirements for dynamic incident handling. Section 3 describes the architecture and the methods used in the proposed real-time fleet management system. Section 4 discusses the testing of the system to a realistic case and the results obtained. Concluding remarks as well as the main benefits from the use of the system are included in Section 5.

9.2 USER REQUIREMENTS AND SYSTEM DESCRIPTION

9.2.1 User Requirements

In order to identify the user requirements for the proposed system, we performed the analysis shown in Figure 9-1. The users targeted by the analysis were third party logistics (3PL) companies as well as manufacturers that distribute their products using private fleets in an urban environment.

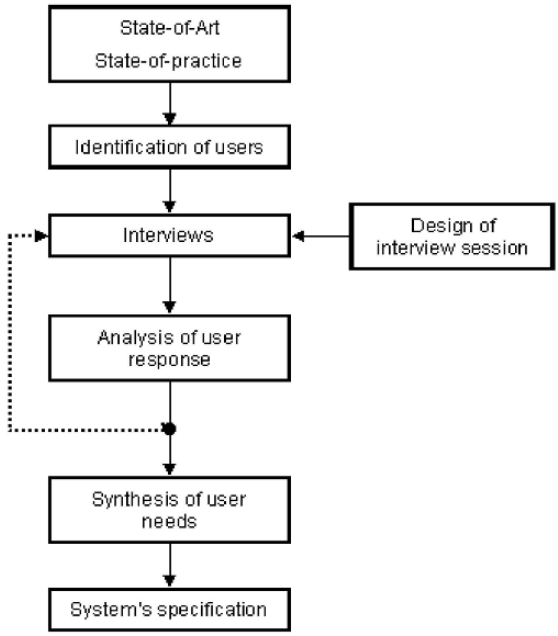


Figure 9-1. Methodological framework for user requirements elicitation.

Table 9-2 summarizes the major system requirements resulting from the above process, and indicates which of these requirements are addressed by current fleet management systems.

Table 9-2. Urban freight distribution requirements.

Requirements for Real-time Fleet Management Systems	**Addressed by current fleet management systems**
Real-time vehicle monitoring	Yes
Vehicle performance reporting	Yes
Proof of delivery	Yes
Dealing with vehicle re-routing	No
Adhering to delivery time windows	No
Dealing with vehicle breakdowns	No

Critical requirements not addressed by such systems include:

- The ability to intelligently reroute a delivery vehicle that has been delayed and, as a result, is no longer capable to serve all scheduled clients.

- The ability to deal with vehicle breakdowns by rerouting nearby vehicles to deliver the load of the immobilized vehicle. If a backup vehicle is not available at the depot, then a vehicle(s) that has both adequate load capacity and time availability should be identified to unload the items from the immobilized vehicle and continue the delivery tasks of the latter.

The aforementioned requirements led to the design of the system architecture presented in the following section.

9.2.2 System Architecture

The proposed system (Figure 9-2) comprises of three subsystems namely back-end, wireless communications, and front-end:

The back-end system incorporates typical components of a fleet monitoring system such as a) a Geographical Information Module (GIM), responsible for managing cartographic information, b) a Data Management Module (DMM) that incorporates data related to clients, vehicles, and distribution schedules, c) a Control Centre User Interface through which the dispatcher can receive the status of the vehicles, proof-of-delivery data, etc, and transmit the revised plan as well as various messages to the drivers, and d) a novel Decision Support Module (DSM), responsible for dynamic incident handling.

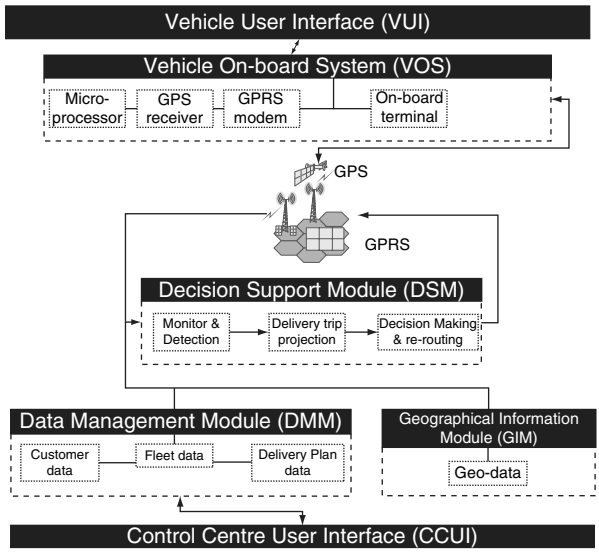


Figure 9-2. System Architecture of the real-time fleet management.

The wireless communication sub-system consists of two parts: a) The *mobile access terrestrial network*, which is responsible for the wireless interconnection of the back-end system with the front-end on-board devices, and b) the *positioning system*, which is responsible for vehicle tracking. We have used the GPRS network for terrestrial data transmission, which supports efficient real-time transfer of data. GPS has been used for vehicle tracking.

The front–end system comprises of the telematic equipment that supports real-time communication and data processing (Vehicle On-Board System), as well as a portable data terminal for the driver.

9.3 MANAGING A DELAYED DISTRIBUTION VEHICLE

The incident handling method, implemented by the system of Section 2.2, is depicted in Figure 9-3 and consists of two stages: a) Monitoring and Detection and b) Decision Making and Rerouting. These stages are described in detail in the following sections.

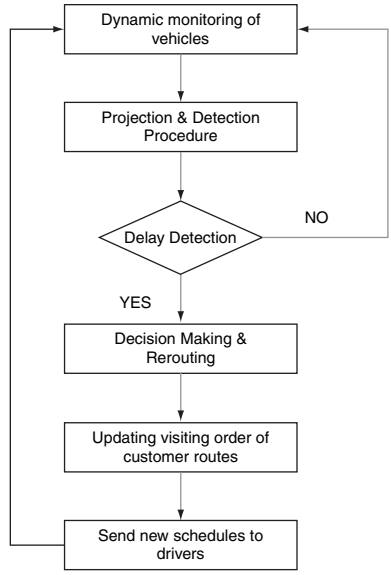


Figure 9-3. The Incident handling method.

9.3.1 Monitoring and Detection

The monitoring mechanism communicates with the vehicle and collects information about its status periodically. This information includes the geo-location of the vehicle, as well as a proof-of-delivery message for each served client. The detection mechanism examines whether the travel time from the current position of the vehicle to each of the remaining non-served clients *i* is less than or equal to the upper limit of the client's time window. This is expressed in Eq. (1), in which *k* and *n* are the next and last clients to be served along the vehicle's route, $k \le i \le n$):

$$t_c + t_{ck} + \sum_{j=k}^{i-1} \bar{t}_{j,j+1} + \sum_{j=k}^{i-1} \bar{t}_{s_j} + \alpha \left(s_{ck}^2 + \sum_{j=k}^{i-1} s_{j,j+1}^2 + \sum_{j=k}^{i-1} s_{s_j}^2 \right)^{1/2} \le t_i^u \quad i \ge k, k+1, \ldots n \quad \text{(l)}$$

In Equation 1, t_c is the current time (point of data collection), t_{ck} is the estimated travel time from current position to the next client k, $\bar{t}_{j,j+1}$ is the mean historical travel time from client *j* to client *j+1*, $\bar{t}_{s_j}$ is the mean historical time for serving client *j*, α is a parameter related to the desired confidence level, s_{ck}^2 is the variance of the estimated travel time from the current position to client *k*, $s_{j,j+1}^2$ is the variance of the historical travel time from client *j* to client *j+1*, $s_{s_j}^2$ is the variance of the historical service time for client *j*, and t_i^u is the upper limit window for client *i*.

The aforementioned equation assumes that travel times $t_{j,j+1}$ between the clients along the route are independent. If this assumption does not hold, then the following equation can be used.

$$\bar{t}_{ci} + \sum_{j=k}^{i-1} \bar{t}_{s_j} + \alpha \left(s_{ci}^2 + \sum_{j=k}^{i-1} s_{s_j}^2 \right)^{1/2} \le t_i^u \quad i \ge k, k+1, \ldots n \qquad (2)$$

This formulation uses the mean historical travel time $\bar{t}_{ci}$ between the current position *c* and client *i*. It does not construct $\bar{t}_{ci}$ from the sum of the travel times $t_{j,j+1}$ of the segments between all intermediate clients from the current position *c* to client *i*. Obviously, in order to apply Eq. (2) historical travel data are required between the current position of the vehicle and each client *i*. This is a limitation in using Eq. (2) instead of Eq. (1), since the latter

allows constructing the path from a sequence of travelled segments (even if the path instance was never travelled before), and obtain the path statistics using the statistics of the segments.

The above method uses historical data from previously executed routes, and gives very accurate results when traffic patterns do not vary significantly over time, but are rather stationary. However, there are cases in which travel times vary significantly over time. Indeed, according to Chung *et al.* (2004), as well as Chien and Kuchipudi (2003), patterns of travel time prediction in urban environments, are not easy to quantify because:

- Travel demand changes everyday due to different activities and different departure times, and this result to travel time fluctuations and road congestion.
- Trip time is affected by incidents, traffic conditions and the weather.
- Urban road networks are complex and an adjacent congested route can affect the subject route.

In such cases, our system uses a second travel prediction method, employs real-time data to compute the network travel times in a dynamic manner. More specifically, as the vehicle is travelling towards its destination, travel time is predicted iteratively by using the average speed achieved by the vehicle in the portion of the travelled route. This method is also used for travel time prediction in links (i.e. routes) where historical data are not available. For further details regarding this method see Zeimpekis (2007).

Both methods are computationally suitable for real time applications, since they include simple calculations. Note that in our system the detection mechanism selects the most suitable method based on traffic patterns and the state of the vehicle.

9.3.2 Decision Making and Rerouting

If the monitoring and detection mechanism detects a significant delay, then the vehicle is rerouted by the decision making subsystem. In this section we describe the case of a delayed delivery vehicle, and methods to address it in an effective manner.

Consider a vehicle assigned to serve clients following a certain sequence. Clients have a predefined time window within which service may begin and are characterized by an individual importance rating. An unexpected event (e.g. heavy traffic) may delay the vehicle significantly so that the remaining time horizon is not adequate to serve all remaining clients and return to the depot by following the initial client sequence.

Thus, the vehicle has to be rerouted in order to serve the most important clients within the available time horizon. There are significant time constraints, e.g. i) the start of serving a client may begin within the client's time window, and ii) the vehicle should return to the depot within the available time horizon.

This problem bears strong similarities with the so-called orienteering problem (OP); the latter concerns a sportsman that visits a set of geographical sites collecting a certain prize from each site. The objective is to maximize the total collected prize; the task is constrained by a time (or distance) upper limit. The first to formulate the OP was Tsiligirides (1984) who proposed two heuristic solution methods, as well as some widely used benchmark problem instances. Golden *et al.* (1987) proved that OP is NP-hard, and Chao *et al.* (1996) proposed an efficient three-step iterative heuristic to solve it.

To formulate the model for the single vehicle re-routing problem, we have used the model of the classical OP enhanced by appropriate constraints representing the client time windows (see e.g. Focacci et al, 2002), and the horizon available to serve all remaining clients (see Appendix A for the mathematical formulation).

Below we present the heuristic proposed to solve this problem. The basic idea is as follows: Consider that the vehicle is departing from client i, and all remaining clients to be served belong to the unvisited client set V_u. Let the departure time from client i be $t_i + s_i$ where t_i is the time when service starts at client i and s_i is the service time at this client.

STEP 1 - For every unvisited client $j \in V_u$, check the feasibility of serving this client j, i.e.

$$r_j \le t_j \le d_j \tag{3}$$

$$t_j + s_j + c_{j0} \le T \tag{4}$$

where t_j is the time that service starts at client j, r_j and d_j represent the beginning (start) and end of the time window of client j, T is the available time horizon and c_{j0} is the travel time from client j to the depot.

STEP 2 - Quantify the desirability of visiting client j. Select the four most desirable clients and among them select one client randomly. The desirability of client j from client i can be quantified by an appropriate metric that takes into account the client's profit P_j, the cost c_{ij} to access client j from client i, and the time S_{ij} (*slack*) available prior to the end of

the time window of client *j*. The random client choice is realized using a roulette wheel function incorporating the normalized client desirability values.

STEP 3 – Repeat the above two steps until no more clients can be feasibly inserted in the route, or until the client set is empty. At this point one feasible solution has been obtained.

STEP 4 – Repeat steps 1 to 3 in order to achieve M feasible solutions.

STEP 5 – Select the best route, out of the M feasible solutions. The best route is the one that gives the maximum cumulative income to the vehicle.

The above algorithm is strongly influenced by the *S-algorithm* of Tsiligirides (1984). However, the introduced time window constraints require modifications to the desirability metric and the stochastic selection of the next client to be inserted in the route.

The desirability metric A_{ij} ,used in our case, takes into account the profit P_j of client *j* in relation to the distance c_{ij} that the vehicle needs to travel to reach that client; *m* is an appropriate integer:

$$A_{ij} = \left(\frac{P_j}{c_{ij}}\right)^m \tag{5}$$

To perform the stochastic client insertion, we used a probability function, which is based on the slacks of the four selected clients. Specifically,

$$\Pi_{ij} = \frac{\frac{1}{S_{ij}}}{\sum_{j \in W} \frac{1}{S_{ij}}} \tag{6}$$

where S_{ij} is the slack of client *j* given by:

$$S_{ij} = d_j - \left(t_i + c_{ij} + s_i\right) \tag{7}$$

Using A_{ij} we rank the remaining clients by favoring those with high profit and low insertion cost. Out of the four most desirables ones, the client with the most "urgent" time window (least slack) will be selected with the highest probability. The random selection is used to avoid local optimum points. It should be emphasized that the combination A_{ij}, Π_{ij} was selected as

the most promising one among several alternative desirability metrics and probability measures that used the client profit P_j the distance (cost) c_{ij} and the slack S_{ij}. This selection was based on extensive experiments with randomly generated problems.

9.4 SYSTEM EVALUATION

9.4.1 System Operation and Case Setting

In a typical operation, the real-time fleet management system monitors continuously the adherence to the initial delivery plan. Figure 9-3 depicts the user interface of the control centre at an initial stage, in which the travel estimation technique has not detected any deviation from the initial plan (the column, which presents the estimated arrival time for each client, is highlighted in green (light grey in Figure 9-4). After a vehicle has served a number of clients (Figure 9-5), the system detects several time window violations for non-served clients (certain cells of the column are highlighted in red-dark grey in Figure 9-5), and proposes a rerouting plan (i.e. a different way for visiting the remaining clients). The new delivery plan is transmitted to the driver through the on-board terminal (Figure 9-6).

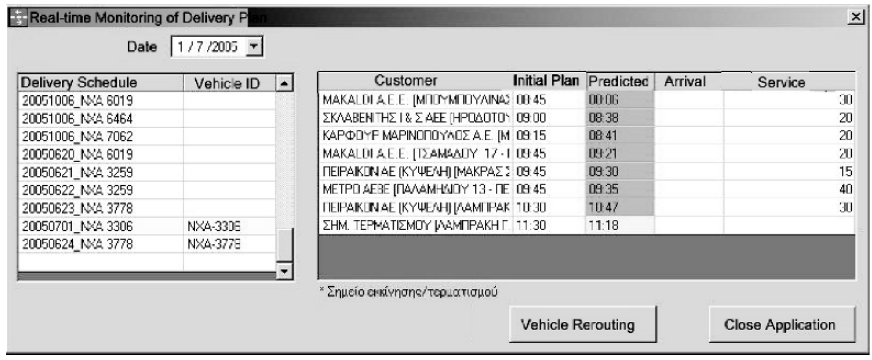


Figure 9-4. Initial monitoring of delivery execution.

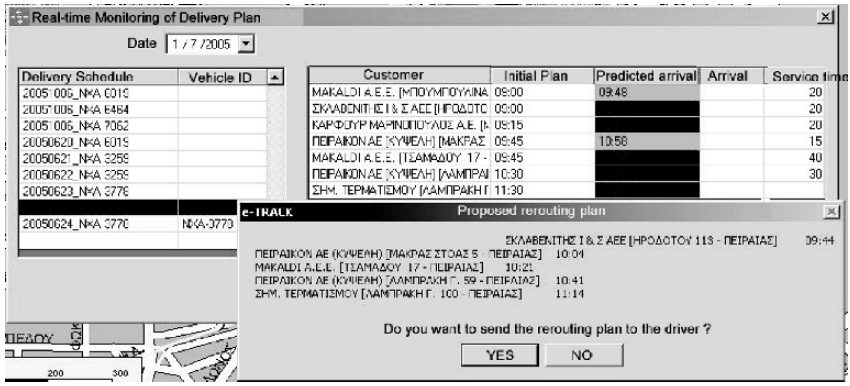


Figure 9-5. Detection of delivery time violation and rerouting plan.

Figure 9-6. On board terminal.

This system was tested in a Greek 3PL carrier (DIAKINISIS S.A.) in various incident handling cases. The test used a real delivery distribution plan and compared the total profit of clients served by a truck that followed the company's initial delivery plan against the total profit obtained based on the routing directions given by the system.

Diakinisis S.A. is one of the largest third party logistics (3PL) companies in Greece. Its core business focuses on the storage, order management, invoicing, and distribution of goods for a large number of commercial and manufacturing companies. It is situated about 15 km from the Athens and Piraeus city centres where more than 60% of its clients are concentrated. Every day, more than 150,000 kg of goods have to be distributed to an average of 300 clients, located at distances ranging from three (3) to forty (40) km from the company's main warehouse complex.

The orders are delivered using an outsourced fleet of 80 vehicles. The company uses a customised software solution for the routing of these vehicles.

Due to the highly congested urban environment of Athens and Piraeus, the company faces various problems due to unexpected incidents (mainly travel and service time delays), that may adversely affect the delivery process. Currently, when a delay occurs, interventions are performed through voice communication between the driver and the dispatcher. Oftentimes the effectiveness of these interventions is limited, since there is no systemic way of taking into consideration the multitude of parameters involved, such as the importance of the remaining clients, time windows restrictions and so on.

9.4.2 Test Design and Results

We assessed the effectiveness of the proposed system by monitoring certain key performance parameters given in Table 9-3 (see also Lai *et al.,* 2004; Krauth *et al.,* 2005; Regan *et al.,* 1998). Note that a very useful indicator of the customer service achieved by a vehicle k, is the ratio CS_k of the sum of weights of clients served by vehicle k over the total weight of all clients in this vehicle's delivery plan.

The testing process was based on the fractional factorial methodology. Factorial experiments (including fractional factorial ones) are systematic ways to perform an experimental investigation that yields all statistically significant effects of all factors and their interactions (Montgomery, 2001). Since the urban freight distribution process is affected by various uncontrolled factors such as daily traffic, congested unloading ramps at the delivery points, environmental conditions (e.g. rain), we decided to engage two vehicles simultaneously in each scenario that would execute the same delivery plan. The first vehicle would perform the daily schedule according to the current delivery method whereas the second would follow the directions provided by the real-time fleet management system. This

Table 9-3. Test performance parameters.

Customer Service	• Number of clients served • Importance (weight factor) of clients served (1-10 Scale) • Total weight of goods delivered (kg) • Total number of time window violations
Operational Cost	• Total distance traveled (km) • Total travel time (hr) • Total service time (hr)

provides unbiased results, since both vehicles were planned to visit the same delivery points under identical circumstances. A weight (i.e. client importance) was given to each client of the plan. The weight factor (from 1-*less important* to 10-*very important*) was defined by the 3PL company according to the type of the client and its importance. The test cases examined are shown in Table 9-4.

In order to force delays, the delivery period was artificially set to be less than the time usually required for a vehicle to complete the delivery plan. For instance, the delivery period was set between 8:30 – 12:30, whereas the time that a vehicle usually requires to complete the delivery plan is from 8:30 to 14:00. In that way, we forced the incident handling system to reroute the designated vehicle. At the end of each testing period we computed the parameters presented in Table 9-3 above, for each vehicle. The route that achieves the highest CS_j is, in general, the preferred one (note that this is also the objective of the re-routing algorithm). In many cases, this route may contain a lower number of clients, but of high importance. However, a route with a higher CS_j score and a higher number of clients served is clearly superior.

Figure 9-7 shows the customer service (quantified by CS_k) achieved for each test case. For all test cases, the vehicle that followed the new route given by the real-time fleet management system (Vehicle B), provided higher customer service. The average difference in CS between the new plan and the initial one is 25, that is an average improvement of a factor of about 1,39 over the initial plan.

Table 9-4. Test cases.

Test case	Area	Traffic	Type of time windows	Number of time windows	Time window width
1	Suburban	Heavy	Driver's shift	N/A	N/A
2	Suburban	Light	Driver's shift	N/A	N/A
3	Urban	Heavy	Driver's shift	N/A	N/A
4	Urban	Heavy	Client's time window	Low	Relaxed
6	Urban	Heavy	Client's time window	Low	Tight
6	Suburban	Light	Client's time window	High	Relaxed
7	Urban	Heavy	Client's time window	High	Tight
8	Suburban	Light	Client's time window	High	Tight

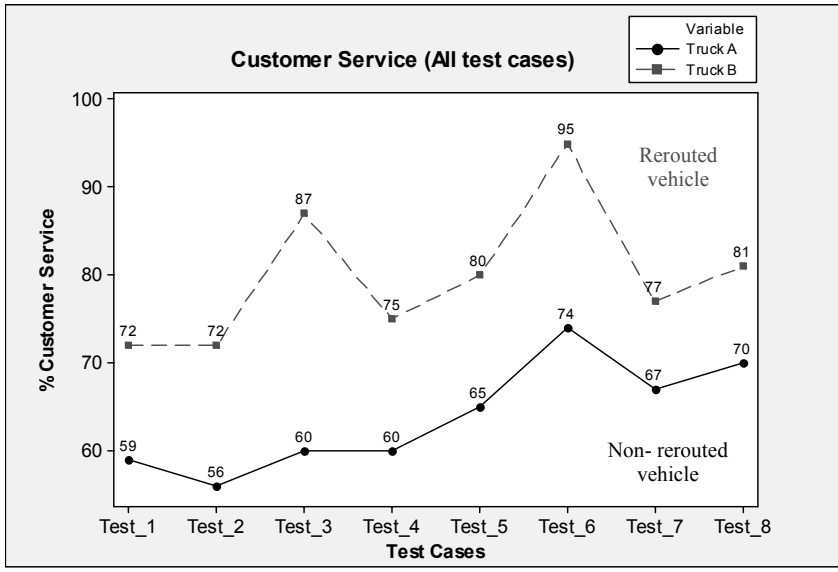


Figure 9-7. Customer Service for all cases in DIAKINISIS S.A.

Table 9-5 summarizes key data for each test case. It provides the initial number of clients, the number of visited clients by each vehicle, the importance of visited clients, the total customer service, as well as the performance achieved in each test case. Thus for Test Case 1, it can be seen that Truck B visited 18 clients of a total importance of 102 points, which led to an improvement in the *CS* by 22.

An important finding from this test has been the impact of time windows on the system's performance. Indeed, in the first three cases (Tests 1, 2 and 3) in which only the restriction of the driver's shift has been applied, the total number of served clients by vehicles A & B was almost equal. In Tests 4, 5, 6, 7 and 8, which included client time windows (in addition to the driver's shift), the system performed better (in terms of clients served) and succeeded in reducing time window violations. The number of clients in the initial plan is another important factor. The higher the number of clients served, the better the performance of the system. This is due to the wider choice of customers to serve in the revised plan. Results from cases 1, 5, and 7 (all of which include a large number of initial clients), show that Truck B visited most of the clients and, in particular, those with higher importance.

Table 9-5. Results from the pilot testing in DIAKINISIS S.A (Truck A follows the predefined delivery schedule whereas Truck B follows the directions provided by the real-time fleet management system).

Test case	**Type of windows**	**Number of scheduled clients**	**Number of visited clients**		**Importance (weight) of visited clients**		**Total distance traveled (km)**		**Total Customer Service (%)**		**Performance Difference**
		Truck A & B	Truck A	Truck B	Truck A	Truck B	Truck A	Truck B	Truck A	Truck B	
1	Driver's Shift	25	14	18	84	102	53,32	55,99	59	72	22
2	Driver's Shift	9	5	6	29	45	36,32	41,99	56	72	28
3	Driver's Shift	9	5	6	33	41	32,14	38,45	60	87	45
4	Client's restrictions	7	4	5	29	36	31,56	37,09	60	75	25
5	Client's restrictions	25	15	22	87	106	59,80	67,90	65	80	23
6	Client's restrictions	6	5	5	29	37	39,30	40,60	74	95	28
7	Client's restrictions	21	13	15	70	81	46,67	43,78	67	77	15
8	Client's restrictions	5	3	4	26	30	27,46	31,76	70	81	15
TOTAL		107	64	81	387	478	326,57	357,56	64	80	25
AVERAGE											

9.5 CONCLUSIONS

In this chapter we described a new system for real-time management of a delayed distribution vehicle. The requirements for dynamic fleet management were elicited through interviews with logistics managers. We presented the architecture of the proposed system, the methodology employed for delay prediction and rerouting, as well as the results from comprehensive tests in a Greek 3PL company. As presented above, the use of the real-time fleet management system increased customer service from 64% to 80%.

Some limitations of the proposed system include:

- The mobile and satellite technology used: Indeed if the vehicle does not have a sky view, the system is unable to track the vehicle, preventing proper estimation of the expected arrival time for the remaining clients. However, as tracking interruptions are usually in the order of a few minutes (i.e. interruptions occur very rarely when the vehicle is in motion) there are no significant practical effects on system performance. As far as possible interruptions of the terrestrial communication (GPRS network) are concerned, there is actually no limitation since information concerning the location of the vehicle as well other collected data are stored in the telematic equipment of the vehicle and are transmitted in the control centre as soon as the network connectivity is restored.
- User acceptance: Acceptance problems were mainly raised by the truck drivers, who were never exposed to a similar IT-assisted way of executing deliveries. Beyond initial difficulties with using the system, the drivers were mainly concerned about the notion that a control centre "spies" over them in real-time.

It is noted that the proposed system may be used for managing other incidents by using appropriate algorithms. We have also developed algorithms and tested the system for the vehicle breakdown case (Zeimpekis, 2007). Finally, the ideas presented here may be extended to emergency services, couriers, rescue and repair services as well as taxi cab services. In each case, the system should address the particular characteristics of the specific environment. Of course, the main concept and architecture of the real-time fleet management system (i.e. incident handling by using dynamic travel prediction methods and rerouting algorithms) will remain applicable.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the Hellenic Ministry of Education (Herakleitos PhD Fellowship) and the Hellenic General Secretariat of Research & Development for partially funding this research under the project entitled Mobile Real-Time Supply Chain Execution (MORSE). We would also like to acknowledge the project partners: Planning S.A, the University of the Aegean, the Athens University of Economics and Business, Emphasis Telematics, Nikas S.A, and Diakinisis S.A.

REFERENCES

Ballou, R. H. "*Business Logistics Management*", 4th International Edition, 2004, Prentice-Hall International Inc., Upper Saddle River, New Jersey

Chao, I.M., Golden, B.L., Wasil, E.A. "A Fast and Effective Heuristic for the Orienteering Problem", *European Journal of Operation Research*, vol. 88, 1996, pp. 475-489

Chien, S. I. J. and Kuchipudi, C. M. (2003) "Dynamic travel time prediction with real-time and historical data", in: Proceedings of the Transportation Research Board 81st Annual Meeting, Washington, DC

Chung, E. Warita, H., Bajwa, S. Kuwahara, M. (2004) "Travel Time Prediction: Issues and Benefits", Proceedings of the 10th World Conference on Transport Research (CD-ROM), Istanbul, Turkey

Fleischmann, B. Gietz, M., Gnutzmann, S. "Time-varying Travel Times in Vehicle Routing", *Transportation Science* 38 (2), 2004, pp.160-174

Focacci, F., Lodi, A., Milano, M. "A hybrid exact algorithm for the TSPTW", INFORMS Journal on Computing 14, 403-417, 2002

Ganz, N., Van Ryzin, G. "Dynamic Vehicle Dispatching: Optimal Heavy Traffic performance and prctical Insights", Operations Research, Vol. 47, 1999, No. 5, pp.675-692

Goetschalckx, M. "A decision support system for dynamic truck dispatching", International Journal of Physical Distribution and Materials Management 14, 1998, pp.34-42

Golden, B.L., Levy, L., Vohra, R. "The Orienteering Problem", *Naval Research Logistics*, vol. 34, 1987, pp. 307-318

Haghani, A. Jung S. "A dynamic vehicle routing problem with time-dependent travel times", *Computers & Operations Research*, Vol.32, No.9, 2005, pp. 2959-2986

Kim, S, Lewis, M.E., White C.C. "Optimal Vehicle Routing with Real-Time Information", Working Paper, 2003, University of Michigan

Krauth, E., Moonen, H., Popova, V., Schut, M.C. "Understanding performance measurement and control in third party logistics", *In the proceedings of the 13th European Conference on Information Systems*, 26-28 May 2005, Regensburg, Germany

Ichoua, S. Gendreau, M. Potvin, J.Y. "Vehicle dispatching with time-dependent travel times", *European Journal of Operational Research* 144, 2003, pp.379-396

Lai, K. H., Ngai, E.W.T., Cheng, T.C.E "An empirical study of supply chain performance in transport logistics", *International Journal of Production Economics*, Vol.87, 2004, pp. 321-331

Lambert, D. M., Cooper, M. C., Pagh, J. D. "Supply Chain Management: Implementation Issues and Research Opportunities", *The International Journal of Logistics Management* 9(2), 1998, pp. 1-19

Lin, S. "Computer solution of the travelling salesman problem", *Bell System Technical Journal*, vol.44, 1965, pp. 2245-2269

Montgomery, D. "*Design and Analysis of Experiments*" 5th edition, John Wiley & Sons, US, 2001

Papachristou, C. "The Orienteering Problem with Time Windows", *Final Year Thesis, Department of Financial and Management Engineering*, University of the Aegean, Greece, 2005

Powell, W.B. "Real-time optimization for truckload motor carriers", OR/MS Today 18, 1990, pp.28-33

Psaraftis, H.N. "Dynamic Vehicle Routing: Status and Prospects", *Annals of Operations Research* 611, 1995, pp. 143-164

Regan, A.C., Mahmassani, H.S., Jaillet, P. "Dynamic decision making for commercial fleet operations using real-time information", *Transportation Research Record*, 1537, 1997, pp.91-97

Regan, A.C., Mahmassani, H.S., Jaillet, P. "Evaluation of dynamic fleet management systems: Simulation Framework", *Transportation Research Record, 1645,* 1998, pp.176-184

Savelsbergh, M.W.P., Sol, M. "Drive: Dynamic routing of independent vehicles", Operations Research 46, 1991, pp. 474-490

Slater, A. "Specification for a dynamic vehicle routing and scheduling system", International Journal of Transportation Management 1, 2002, pp.29-40

Taniguchi, E., Shimamoto, H. "Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times" Transportation research Part C, 2004, Vol. 12, pp.235-250

Tsiligirides, T. "Heuristic Methods Applied to Orienteering", *Journal of Operational Research Society*, vol. 35/9, 1984, pp. 797-809

Zeimpekis, V., Giaglis, G.M., Lekakos, G. "Towards a taxonomy of indoor and outdoor positioning techniques for mobile location–based applications", *Journal of ACM, SIGecom Exchanges*, Vol. 3, No. 4, 2003, pp.19-27

Zeimpekis, V. Giaglis, G.M., Minis, I. "A dynamic real-time fleet management system for incident handling in city logistics" *In the proceedings of 61st IEEE Vehicular Technology Conference, (VTC2005 Spring),* 30 May-1 June, 2005, Stockholm, Sweden

Zeimpekis, V. "Design and evaluation of a real-time fleet management system for dynamic incident handling in urban freight distribution", Athens University of Economics & Business, PhD Thesis, 2007

APPENDIX A

Mathematical Model of the Rerouting Problem

Following the conventional notation of the Travelling Salesman (TS) and the Vehicle Routing Problem(s) (VRP), consider a network *G* comprising a set of vertices $V = \{0, k, 1, ..., n\}$ and a set of arcs A that interconnect these vertices. Vertices in set V represent the depot (0), n clients and the node k (beginning of route). We associate a cost (travel time) c_{ij} to all arcs $x_{ij} \in A$. To each vertex we associate a service cost s_i, which is the time required to serve the corresponding client. We also define the income parameter p_i, which represents a business metric associated with client i, such as the estimated sales volume to this client, or the client importance. A route is a sequence of arcs, which the vehicle will follow originating from the depot, to serve all remaining clients and return to the depot. The following notations are also used in the formulation:

- $\Delta^{+}(i)$ (forward star): is defined as the set of unvisited customers that can be visited directly after customer i
- $\Delta^{-}(i)$ (backward star): is defined as the set of customers that were visited before customer i
- $V_u = V \setminus \{0,k\}$

Let $y_i \in \{0,1\}$ be a binary variable, such that $y_i = 1$ if client $i \in V_u$ is served and $y_i = 0$ otherwise. Also let $x_{ij} \in \{0,1\}$ be another binary variable, such that $x_{ij} = 1$ if arc $x_{ij} \in A_u$ belongs to the new route and $x_{ij} = 0$ otherwise. For each client *i*, *[r(i), d(i)]* defines the time window of *i*. Specifically, *r(i)* indicates the earliest possible arrival and *d(i)* the latest possible arrival of the vehicle. Similarly, *d(0) = T* is the latest possible arrival of the vehicle to the depot. Finally, let the objective be to optimize the total income resulting from serving the clients selected in the new plan. Then, the mathematical program that models the single vehicle re-planning problem with time windows is given below.

$$\max \quad \sum_{i \in V_u}^{n} p_i y_i \qquad \text{(A-1)}$$

s.t.

$$\sum_{j \in \Delta^{+}(i) \setminus \{0\}} x_{ij} = y_i \qquad \forall\, i \in V_u \qquad \text{(A-2)}$$

$$\sum_{j \in \Delta^{-}(i) \setminus \{k\}} x_{ji} = y_i \qquad \forall\, i \in V_u \qquad \text{(A-3)}$$

$$\sum_{j\in\Delta^+(k)} x_{kj} = 1 \tag{A-4}$$

$$\sum_{j\in\Delta^-(k)} x_{jk} = 0 \tag{A-5}$$

$$\sum_{j\in\Delta^+(0)} x_{0j} = 0 \tag{A-6}$$

$$\sum_{j\in\Delta^-(0)} x_{j0} = 1 \tag{A-7}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \le \sum_{i\in S} y_i - 1 \quad \forall \mathrm{S}\subseteq \mathrm{V_u} \tag{A-8}$$

$$\sum_{i\in V_u}\sum_{j\in V} c_{ij}x_{ij} + \sum_{i\in V_u\setminus\{0,k\}} s_i y_i \le T \tag{A-9}$$

$$x_{ij}(t_i + s_i + c_{ij} - t_j) \le 0 \qquad \forall \mathrm{i}, j \in \mathrm{V_u} \tag{A-10}$$

$$r_j \sum_{j\varepsilon\Delta^+(i)} x_{ij} \le t_j \le d_j \sum_{j\varepsilon\Delta^+(i)} x_{ij} \qquad \forall \mathrm{i}, j \in \mathrm{V_u} \tag{A-11}$$

$$x_{ij}, y_i \in \{0,1\} \quad \forall x_{ij} \in A_u, \forall i \in V_u \tag{A-12}$$

Constraints (A-2) and (A-3) ensure that if client i is served s/he will be served exactly once, constraints (A-4) and (A-5) indicate that there is only one arc incident at vertex *k*, constraints (A-6) and (A-7) indicate that there is only one arc incident at the depot, constraint (A-8) ensures that there will be no sub-tours in the new route, and constraint (A-9) indicates that the time required to complete the new route must be lower or equal to the remaining time in the time horizon (s_i is the service time of customer *i*). Constraint (A-10) indicates that the start of service t_j of client *j* is longer or equal to the arrival time in *i* plus the service time in *i* plus the time to travel from $i \to j$. Constraint (A-11) ensures that start of service of client *i* is within the time window of this client. Note that in the above formulation, waiting at a client for its time window to open is not allowed.

Chapter 10

REAL-TIME FLEET MANAGEMENT AT ECOURIER LTD

Andrea Attanasio[1], Jay Bregman[2], Gianpaolo Ghiani[3] and Emanuele Manni[3]
[1]Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, Cosenza, Italy; [2]eCourier Ltd, London, UK; [3]Dipartimento di Ingegneria dell'Innovazione, Università di Lecce, Lecce, Italy

Abstract: In this chapter we describe an innovative real-time fleet management system designed and implemented for eCourier Ltd (London, UK) for which patents are pending in the United States and elsewhere. This paper describes both the business challenges and benefits of the implementation of a real-time fleet management system (with reference to empirical metrics such as courier efficiency, service times, and financial data), as well as the theoretical and implementation challenges of constructing such a system. In short, the system dramatically reduces the requirements of human supervisors for fleet management, improves service and increases courier efficiency. We first illustrate the overall architecture, then depict the main algorithms, including the service territory zoning methodology, the travel time forecasting procedure and the job allocation heuristic.

Keywords: courier industry; same-day courier; global positioning system; real-time fleet management; travel time forecasting; job allocation.

10.1 INTRODUCTION

Same day couriers are utilised by clients who require maximum speed and security for deliveries. Clients usually request couriers with little or no notice, all but eliminating the ability to construct routes or schedules in advance. Once a courier has been assigned a job, he/she proceeds directly to the pickup location, collects the appropriate conveyance, and moves on to the delivery where in return, a signature is obtained. In standard (non-premium or non-solo) service, the courier may "consolidate" deliveries. That is, if a courier is given two or three deliveries at the same time he/she may pick up all three and then deliver all three. Heavy users include law firms,

financial institutions and advertising agencies all of whom send tangible items or require original signatures on documents.

The traditional model of same-day courier service utilises human controllers who communicate with bicycle, motorcycle, car and van couriers via radio or mobile phone. Controllers ask couriers to relay their location information and then assign jobs to the closest and most appropriate courier. This model is not only inefficient, but also suffers from errors inherent with the involvement of a human element. As the number of couriers increases to several hundreds or thousands, informational complexity grows to levels which push the limits of human analysis. At this stage several controllers may be used, increasing costs and requiring constant coordination between controllers in addition to communication with couriers on the street. Human controllers can generally manage a maximum of thirty couriers in a specific area of the city and make job allocation decisions based on incomplete or inaccurate information. This has contributed to the fact that no courier company can gain a very large market share, due to the informational complexity of the allocation problem.

eCourier Ltd commissioned a system whereby courier location information and vehicle type, among several other variables, are available to us in real-time. These information are used by an innovative set of algorithms able to allocate each job to the most appropriate courier on the basis of road congestion and current fleet status as well as individual courier efficiency. Courier location information is provided by GPS devices (Cathey and Dailey, 2003) embedded into palmtop computers which are also used to provide directions to couriers. Traffic patterns on various London streets at certain times of day may also be relevant, especially with regard to cars and vans. That is, the most appropriate courier may not be the closest one, because of congestion and obstacles such as the River Thames. Traffic will vary at different times of days (such as rush hour), days of the week (weekends are generally clearer), and particular times of the year (e.g., holidays).

10.2 THE AUTOMATED INFORMATION-BASED ALLOCATION SYSTEM

This paper describes the forecasting and optimization methodologies that we have implemented and tested in order to develop eCourier's Automated Information-Based Allocation System (AIBA). AIBA is embedded in the system outlined in Figure 10-1 and is made up of two main subsystems: a FORECAST module and an ALLOCATE module.

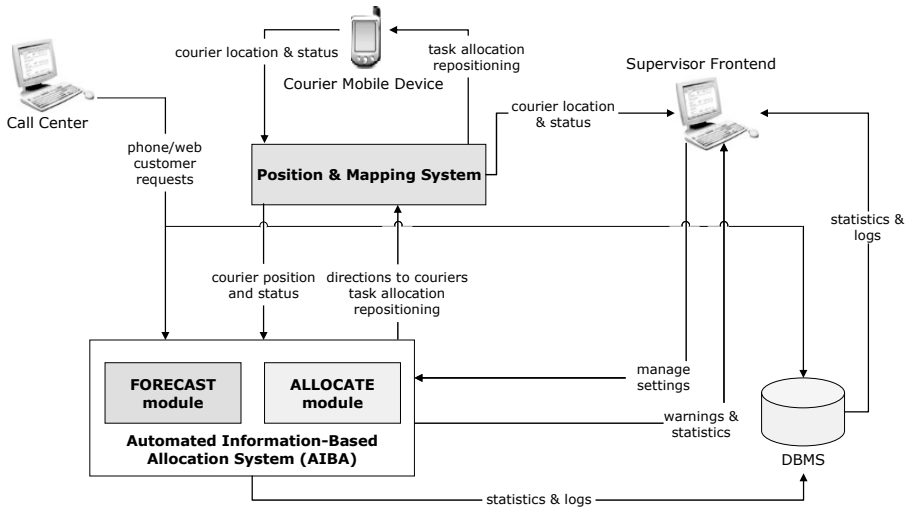


Figure 10-1. eCourier system.

The aim of the forecasting module is to provide reliable near future predictions of customer demand and courier travel times to the allocation procedure. It is worth recalling that demand forecasts are needed in order to reposition idle couriers while travel time forecasts are used mainly for allocating customer requests to couriers. Demand forecasting is based on a time series extrapolation. Instead travel times predictions are based on a time series extrapolation followed by a neural network which takes into account real-time traffic and weather information. Since no time series were available at the beginning of the development, both demand and travel time forecasting methods have been initialized through a multi-stage survey resembling the Delphi method (Montgomery *et al.,* 1990).

10.2.1 Zoning the Service Territory

In order to capture traffic and demand patterns, it is assumed that the service territory is divided into *n zones*. Service territory zoning is done on the basis of ZIP coding in such a way to achieve a suitable compromise between accuracy, computing time and storage requirements. In the UK, zip codes are highly granular, such that full postcodes are generally unique to one or at most a handful of properties. As far as the Greater London area is concerned (Figure 10-2), a zone is characterized by the first letter (or pairs of letters) and the first digit (or pair of digits). This choice has two main advantages: (i) the zoning can be done automatically (i.e., given an address it is immediately known the identifier of its zone); (ii) the distance between any two points in most zones does not exceed 1.2-1.5 km. Let $\sigma(i)$ be the zone including location (address) i.

Figure 10-2. Greater London area zoning.

10.2.2 Time Subdivision

We divided a year into a suitable number of time periods showing typical traffic and demand patterns. We consider n_1 periods. A feasible subdivision is (after each period, we report the associated identifier):

- New Year time (after Christmas day to Mid Feb) = **1**;
- Budget time (Mid Feb to Budget in Mid March) = **2**;
- End of Financial Year time (until April 5) = **3**;
- I part of April (approximately before Easter) = **4**;
- II part of April (approximately after Easter - "May Bank Holiday" period) = **5**;
- Summer period (mid May to mid July) = **6**;

- Holiday Season (end of July to end of August) = **7**;
- Autumn (to Halloween) = **8**;
- Christmas period (until 24th December) = $\boldsymbol{n_1}$ = **9**.

Similarly, we divided a week into n_2 groups of homogeneous days showing typical traffic patterns. In this case a feasible subdivision is:

- Monday = **1**;
- Tuesday = **2**;
- Wednesday = **3**;
- Thursday = **4**;
- Friday = **5**;
- Saturday = **6**;
- Sunday = $\boldsymbol{n_2}$ = **7**.

Finally, a day is divided into n_3 time slots characterized approximately by the same traffic and/or demand patterns (rush hours, etc.). For instance, a feasible subdivision is:

- 07:00 - 08:30 = **1**;
- 08:30 - 10:00 = **2**;
- 10:00 - 11:30 = **3**;
- 11:30 - 13:00 = **4**;
- 13:00 - 14:00 = **5**;
- 14:00 - 15:30 = **6**;
- 15:30 - 16:30 = **7**;
- 16:30 - 18:00 = **8**;
- 18:00 - 19:30 = **9**;
- 19:30 - 07:00 = $\boldsymbol{n_3}$ = **10**.

10.2.3 Forecasting Logistics Requirements

The FORECAST module estimates future travel times between origin-destination points. Such forecasts are based on both historical data and real-time information (weather and traffic conditions). The FORECAST module:

[1] monitors courier travel times from parking areas to pickup points, from pickup points to delivery points, etc.;
[2] uses these real-time data to update traffic patterns (seasonal indices, etc.) as well as to compute near future estimates;
[3] uses recent real-time data to update info provided to customers (expected travel times to pickup points, expected completion times of undergoing tasks, etc.);
[4] provides near future travel time estimates to the ALLOCATE module;
[5] warns system supervisor when a courier shows an undesired behaviour (e.g., if the expected completion time of his/her current task increases or does not decrease, if a user-specifiable number of rejections occurs, etc.).

The forecasting methodology is based on a classical *decomposition technique* (which takes into account the time of day, the day of the week and the particular time of the year) followed by an *artificial neural network* (ANN) which accounts for real-time info.

Let t (≥ 1) be a time period ($t = 1$ is associated to period 07:00-8.30 of a reference day in the past) and let $\alpha(t)$, $\beta(t)$, $\gamma(t)$ be the period of the year, the day of the week and the period of the day characterizing t, respectively. Finally, let $\tau(t)$ be the duration of period t (in minutes).

10.2.3.1 Demand forecasting

In order to compare time periods of different durations, we refer to the *demand rates* (number of service requests per minute) instead of the demands. Let $D_{rs}(t)$ be the number of transportation requests from zone r to zone s during period t. Then the associated demand rate is $d_{rs}(t)=D_{rs}(t)/\tau(t)$. The forecasting model we have used is based on a modification of the classical decomposition approach (Montgomery *et al.,* 1990):

$$d_{rs}(t) = T_{rs}(t) \cdot M_{rs}\left(\alpha(t)\right) \cdot W_{rs}\left(\beta(t)\right) \cdot H_{rs}\left(\gamma(t)\right) \cdot R_{rs}(t) \tag{1}$$

where:

- $T_{rs}(t)$ is the trend;
- $M_{rs}(\cdot)$ is the yearly seasonal effect;
- $W_{rs}(\cdot)$ is the weekly seasonal effect;
- $H_{rs}(\cdot)$ is the daily seasonal effect;
- $R_{rs}(t)$ is a random fluctuation.

$T_{rs}(t)$ is obtained through a quadratic regression. The yearly effect in period of the year a is given by

$$M_{rs}(a) = \frac{\sum_{t:\alpha(t)=a} \frac{d_{rs}(t)}{T_{rs}(t)}}{|\, t:\alpha(t)=a \,|} \quad a = 1,...,n_1 \tag{2}$$

Then, $M_{rs}(\cdot)$ indices are normalized in such a way their average is equal to n_1:

$$M_{rs}(a) = \frac{n_1 M_{rs}(a)}{\sum_{\alpha=1}^{n_1} M_{rs}(\alpha)} \quad a = 1,...,n_1 \tag{3}$$

The weekly effect in day of the week b is given by:

$$W_{rs}(b) = \frac{\sum_{t:\beta(t)=b} \frac{d_{rs}(t)}{T_{rs}(t)M_{rs}(t)}}{|t:\beta(t)=b|} \quad b = 1,...,n_2 \tag{4}$$

$W_{rs}(\cdot)$ indices are normalized. The daily effect in period of the day c is given by:

$$H_{rs}(c) = \frac{\sum_{t:\gamma(t)=c} \frac{d_{rs}(t)}{T_{rs}(t)M_{rs}(t)W_{rs}(t)}}{|t:\gamma(t)=c|} \quad c = 1,...,n_3 \tag{5}$$

Then $H_{rs}(\cdot)$ indices are normalized. The forecast for a future time period t is simply obtained by multiplying the associated trend and seasonal effects:

$$d_{rs}(t) = T_{rs}(t) \cdot M_{rs}\left(\alpha(t)\right) \cdot W_{rs}\left(\beta(t)\right) \cdot H_{rs}\left(\gamma(t)\right) \tag{6}$$

The random fluctuation series is also computed, in order to assess the accuracy of the forecasting method:

$$R_{rs}(t) = \frac{d_{rs}(t)}{T_{rs}(t)M_{rs}(\alpha(t))W_{rs}(\beta(t))H_{rs}(\gamma(t))} \tag{7}$$

Indeed we expect that the sample average of $R_{rs}(t)$ is close to 1 and the Durbin-Watson coefficient is close to 4 (Montgomery *et al.,* 1990). Furthermore, the sample deviation standard of $R_{rs}(t)$ is the forecasting mean squared error (MSE).

10.2.3.2 Travel time forecasting

The methodology used to forecast the travel time $\theta^k_{ij}(t)$ from location (address) i to location (address) j during time period t (using vehicle type k) is similar to the one used for demands except that: the patterns of locations i and j are those of zones $\sigma(i) = p$ and $\sigma(j) = q$, respectively; traffic and weather real-time information (Figures 10-3 and 10-4) are taken into account through suitably trained artificial neural networks (ANNs).

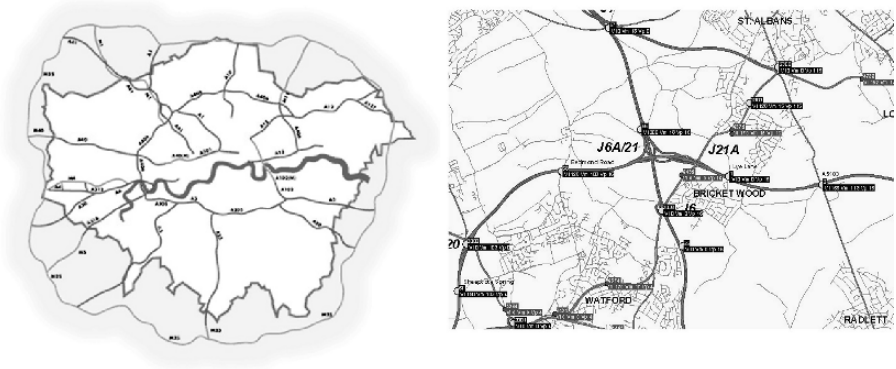


Figure 10-3. Location of traffic sensors in the Greater London area.

Site	Time	Speed
3	16/06/2004 9.29	54
3	16/06/2004 9.33	54
3	16/06/2004 9.37	54
3	16/06/2004 9.41	55
3	16/06/2004 9.45	59
3	16/06/2004 9.49	55
3	16/06/2004 9.53	57
3	16/06/2004 9.57	56
3	16/06/2004 10.01	55
3	16/06/2004 10.05	57
3	16/06/2004 10.09	57
3	16/06/2004 10.13	54
3	16/06/2004 10.17	58

Line	Text (CEN-English, see Explanation Notes)	Code	N	Q	T	D	U	C	R
50	(Q) accident(s). Stationary traffic for 1 km	216		0	D	1	U	1	B1.A101
51	(Q) accident(s). Stationary traffic for 2 km	217		0	D	1	U	1	B1.A102
52	(Q) accident(s). Stationary traffic for 3 km	348		0	D	1	U	1	B1.A103
53	(Q) accident(s). Stationary traffic for 4 km	218		0	D	1	U	1	B1.A104

Figure 10-4. Sample output of the real-time traffic info system.

The forecasting model we used is based on the classical decomposition approach:

$$\theta_{ij}^{k}(t) = t_{ij}^{k} \cdot T_{\sigma(i)\sigma(j)}^{k}(t) \cdot M_{\sigma(i)\sigma(j)}^{k}\left(\alpha(t)\right) \cdot W_{\sigma(i)\sigma(j)}^{k}\left(\beta(t)\right) \cdot H_{\sigma(i)\sigma(j)}^{k}\left(\gamma(t)\right) \cdot R_{\sigma(i)\sigma(j)}^{k}(t) \tag{8}$$

where:

- t^k_{ij} is a reference travel time between locations i and j using vehicle type k;
- $T^k_{\sigma(i)\sigma(j)}(\cdot)$ is the trend;
- $M^k_{\sigma(i)\sigma(j)}(\cdot)$ is the yearly seasonal effect;
- $W^k_{\sigma(i)\sigma(j)}(\cdot)$ is the weekly seasonal effect;
- $H^k_{\sigma(i)\sigma(j)}(\cdot)$ is the daily seasonal effect;
- $R^k_{\sigma(i)\sigma(j)}(\cdot)$ is a random fluctuation.

The travel time of transportation mode k between two nodes i and j, t^k_{ij}, is modelled as follows:

$$t^k_{ij} = l_{ij} \cdot v^k_{\sigma(i)\sigma(j)} \cdot y^k_{\sigma(i)\sigma(j)}(m) \cdot w^k_{\sigma(i)\sigma(j)}(g) \cdot d^k_{\sigma(i)\sigma(j)}(h) \cdot r^k_{\sigma(i)\sigma(j)} \tag{9}$$

where:

- $k = 1$ for bikes, $k = 2$ for motorbikes and $k = 3$ for cars and vans;
- l_{ij} is the distance between nodes i and j provided by the mapping system;
- $v^k_{\sigma(i)\sigma(j)}$ is the average speed of a courier (using transportation mode k) between zones $\sigma(i)$ and $\sigma(j)$ in case of very light traffic (including parking time at destination j);
- $y^k_{\sigma(i)\sigma(j)}(m)$ is the seasonal index corresponding to period of the year m;
- $w^k_{\sigma(i)\sigma(j)}(g)$ is the seasonal index corresponding to period of the weak g;
- $d^k_{\sigma(i)\sigma(j)}(h)$ is the seasonal index corresponding to period of the day h;
- $r^k_{\sigma(i)\sigma(j)}$ is an index taking into account real-time and quasi real-time events like weather conditions, traffic jams, etc.

$T^k_{\sigma(i)\sigma(j)}(t)$ is obtained through a quadratic regression over $\theta^k_{ij}(t)/\ t^k_{ij}$. The yearly effect in period of the year a is given by

$$M^k_{\sigma(i)\sigma(j)}(a) = \frac{\sum_{t:\alpha(t)=a} \frac{d^k_{\sigma(i)\sigma(j)}(t)}{T^k_{\sigma(i)\sigma(j)}(t)}}{|\,t:\alpha(t)=a\,|} \quad a = 1,\ldots,n_1 \tag{10}$$

Then $M^k_{\sigma(i)\sigma(j)}(\cdot)$ indices are normalized. The weekly effect in day of the week b is given by

$$W^k_{\sigma(i)\sigma(j)}(b) = \frac{\sum_{t:\beta(t)=b} \frac{d^k_{\sigma(i)\sigma(j)}(t)}{T^k_{\sigma(i)\sigma(j)}(t) M^k_{\sigma(i)\sigma(j)}(t)}}{|\,t:\beta(t)=b\,|} \quad b = 1,\ldots,n_2 \tag{11}$$

Then $W^k_{\sigma(i)\sigma(j)}(\cdot)$ indices are normalized. The daily effect in period of the day c is given by

$$H^k_{\sigma(i)\sigma(j)}(c) = \frac{\sum_{t:\gamma(t)=c} \frac{d^k_{\sigma(i)\sigma(j)}(t)}{T^k_{\sigma(i)\sigma(j)}(t) M^k_{\sigma(i)\sigma(j)}(t) W^k_{\sigma(i)\sigma(j)}(t)}}{|\, t:\gamma(t)=c \,|} \quad c = 1,\ldots,n_3 \qquad (12)$$

Then $H^k_{\sigma(i)\sigma(j)}(\cdot)$ indices are normalized. The forecast for a future time period t is obtained as follows:

$$\begin{aligned} \theta^k_{ij}(t) = t^k_{ij} \cdot T^k_{\sigma(i)\sigma(j)}(t) \cdot M^k_{\sigma(i)\sigma(j)}\big(\alpha(t)\big) \cdot \\ W^k_{\sigma(i)\sigma(j)}\big(\beta(t)\big) \cdot H^k_{\sigma(i)\sigma(j)}\big(\gamma(t)\big) \end{aligned} \qquad (13)$$

As in the case of demand forecasting, the random fluctuation series is computed in order to assess the accuracy of the forecasting method:

$$R^k_{\sigma(i)\sigma(j)}(t) = \frac{\theta^k_{ij}(t)}{t^k_{ij} T^k_{\sigma(i)\sigma(j)}(t) M^k_{\sigma(i)\sigma(j)}(\alpha(t)) W^k_{\sigma(i)\sigma(j)}(\beta(t)) H^k_{\sigma(i)\sigma(j)}(\gamma(t))} \qquad (14)$$

In order to take into account traffic and weather real-time information, we make use of an ANN. The scheme we have implemented is shown in Figure 10-5.

We have implemented a multilayer feedforward neural network which is commonly recognized to be able to approximate almost any function if there are enough neurons in the hidden layers (Figure 10-6).

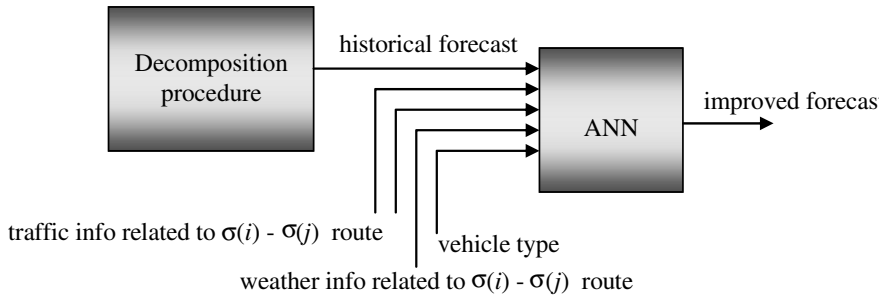


Figure 10-5. Improving the historical forecast of travel time between zones $\sigma(i)$ and $\sigma(j)$ through a neural network.

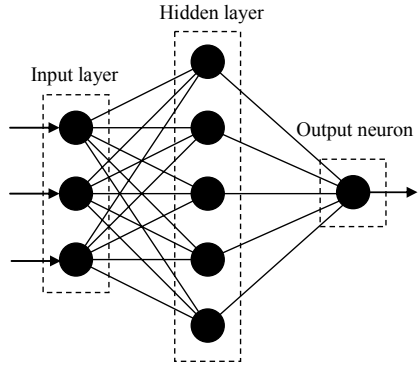


Figure 10-6. Input-output structure of a neural network.

The appropriate number of hidden neurons and layers of neural network depends on the pattern and complexity of the approximated function and the transfer function of the layers. According to previous studies and a preliminary analysis, we guess that one hidden layer is expected to perform well. The input and output neurons are linear while the hidden neuron are sigmoidal. The best number of hidden neurons has to be determined experimentally. On the basis of a preliminary test on dummy data, the number of hidden layers was set equal to 1.3 times the number of inputs. Finally, the ANNs will be suitably trained under supervision. In order to do this, for every prediction we store both the inputs, the ANN forecast and the real travel time evaluated a posteriori.

10.2.4 On Line and Off Line Procedures

In order to avoid overloading the computing system during peak time, both the update of the seasonal indices and the training of the neural network training are performed at night. It is expected they may require up to three hours.

10.2.4.1 Initialization

Since no demand or travel time historical series is available at the moment, both demand and travel time forecasting methods have been initialized through a multi-stage survey resembling the Delphi method (Montgomery et al, 1990). This procedure have allowed us to estimate both trends $T_{rs}(t)$ and $T^k_{\sigma(i)\sigma(j)}(t)$, as well as seasonal indices $M_{rs}(\alpha(t))$, $W_{rs}(\beta(t))$, $H_{rs}(\gamma(t))$, $M^k_{\sigma(i)\sigma(j)}(\alpha(t))$, $W^k_{\sigma(i)\sigma(j)}(\beta(t))$ and $H^k_{\sigma(i)\sigma(j)}(\gamma(t))$. Finally, we have estimated reference travel times t^k_{ij} as follows. In principle this "static" travel time can be computed in real-time by using commercially available routing implementations. However, because of the relatively large time required to

compute a path (about 0.3 seconds on a standard personal computer), a slightly different approach has been used. Firstly, we have divided each zone into a number of microzones using the ZIP code (a microzone is characterized by four ZIP code elements). Then we have computed the shortest route matrix between microzones by using a commercially available routing implementation. This calculation also allowed us to identify the most relevant real-time traffic information sources for each pair of zones.

10.2.4.2 Allocating couriers

The ALLOCATE module assigns customer requests to available couriers and repositions idle couriers from low demand to high demand zones. When a new request arrives, the algorithm performs a *feasibility check* (FC), i.e., it searches for a feasible solution including the new service request. Once it has been decided whether the new request can be accepted or not, the algorithm performs a "post-optimization" (PO), i.e., it tries to improve the current solution. Post-optimization is run in background. We have implemented a tailored tabu search (TS) for both FC and PO phases. However, TS parameters depend on whether a FC or a PO is being performed. When solving a real-time routing and dispatching problem with several hundred couriers, a parallel implementation is usually needed in order to make route re-optimization computation time acceptable. We use an asynchronous single-point multiple strategy parallelization strategy in which each process is coded in Java.

10.2.4.3 The ALLOCATE module

As said before, the ALLOCATE module assigns customer requests to available couriers and repositions idle couriers from low demand to high demand zones. The objectives to attain are twofold and, more specifically, maximizing customer level (i.e., minimizing the average delay a customer experiences from job booking to delivery) and maximizing the average number of jobs completed by a courier in a working day.

The constraints to which we are subject are the compatibility between jobs and vehicles, pickup deadlines for some customers (*service level agreement*), pickup time windows for jobs booked in advance and vehicle capacity. Time windows for each customer are determined according to the class the customer belongs to. We identified three classes of customers, and precisely:

- class 1 customers, who have an account with eCourier and have also signed a Service Level Agreement (SLA). Time windows for these customers are determined according to the SLA; they are usually very narrow and have a duration of 20 minutes starting from the instant the customer books the job;

- class 2 customers, who have an account with eCourier, but have not signed an SLA. Time windows for this class have a duration of 60 minutes starting from the instant the customer books the job.
- class 3 customers (also called "spot customers") who book their jobs via the web or via phone and pay by credit card each time they book a job. Time windows for these customers are usually quite large and have a duration of 90 minutes starting from the instant the customer books the job.

If the time windows are too tight and make it not possible to have a feasible solution, the supervisor have the possibility to modify them, only for customers belonging to classes 2 and 3, through some interfaces of the application (figure 10-7), aiming at obtaining a feasible solution.

An important aspect is that job queuing and job consolidation (*courier diversion*) are allowed. In particular, job queuing (figure 10-8a) means that a courier can take on more than one job before effecting the first delivery, while job consolidation (figure 10-8b) occurs when a courier is en route to a pickup (or delivery) and another pickup with a similar delivery location is allocated to the courier "on the fly".

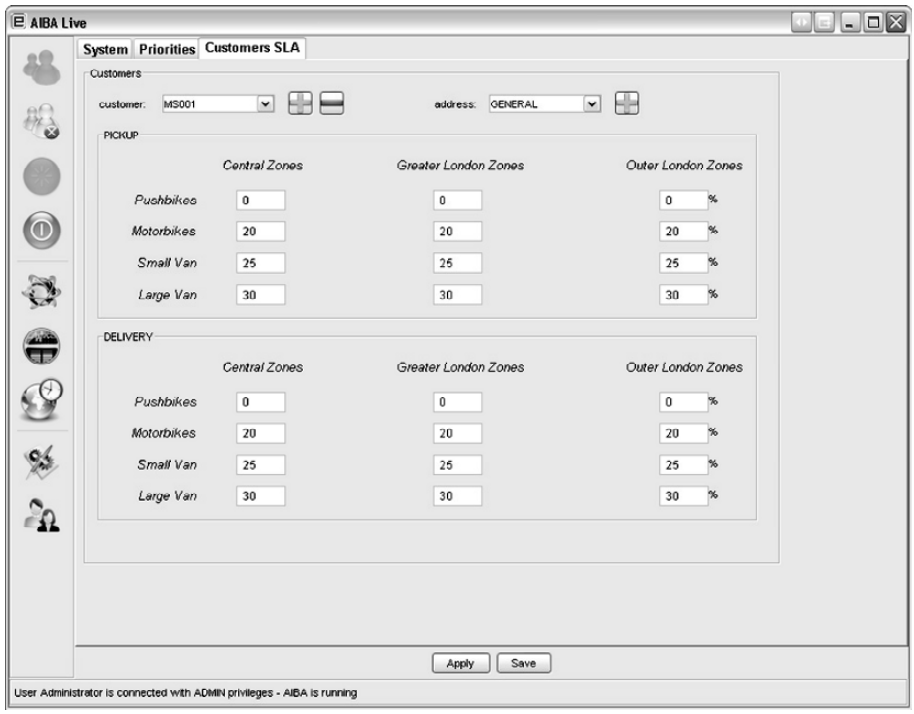


Figure 10-7. A sample of interface used by the supervisor to modify the time windows.

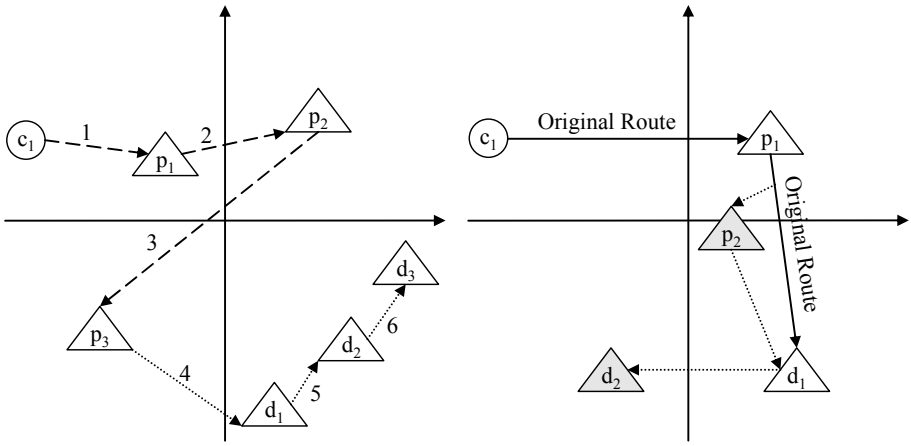


Figure 10-8. Job queuing and job consolidation.

A peculiar feature of this application is that couriers have the ability to accept or reject jobs. As such, the allocation module offers an incoming job to a set of k appropriate couriers (k to be determined). Another important feature of the application is that may exist jobs with a single pickup and multiple deliveries (sequence given).

10.2.4.4 The allocation procedure

At any time the jobs allocated by AIBA are divided into two groups (figure 10-9): jobs which are known by couriers and jobs which are (temporarily) allocated but not known by couriers yet. This allows to keep the solution more flexible by modifying the allocation of the latter requests in case new requests arrive or other event occurs.

The allocation procedure performs a neighborhood search, in which the objective function is

$$\alpha_1 \sum_{k \in R} \max\left(0, t_k - d_k\right) + \alpha_2 \sum_{k \in R} T_k + \alpha_3 \psi \tag{15}$$

where:

- $\alpha_1+\alpha_2=1$, $\alpha_1 \geq 0$, $\alpha_2 \geq 0$;
- R is the set of requests not yet serviced;
- t_k is the expected arrival time of a courier at pickup location k;

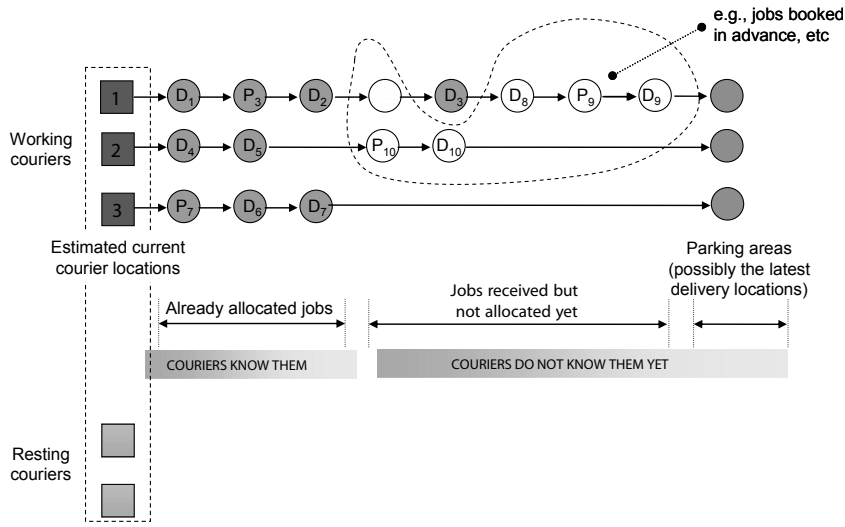


Figure 10-9. AIBA current solution structure.

- d_k is the pickup deadline for request k;
- T_k is the "service time" of request k (time from request to pickup plus travel time between pickup and delivery locations).

The procedure can be outlined as follows:

while "no event"
  run a BACKGROUND OPTIMIZATION procedure;
if an "event" occurs **then**
  (a) stop the BACKGROUND OPTIMIZATION procedure;
  (b) **if** the event is the "occurrence of a new request" **then** run a fast INSERTION procedure;
  (c) **if** the event is the "arrival of a courier at a pick-up or delivery point", **then** offer this courier a request (if available) or REPOSITION the courier;
  (d) if the event is a significant travel time modification, update the data structure of the INSERTION and BACKGROUND OPTIMIZATION procedure.

10.2.4.5 The fast insertion procedure

The fast insertion procedure tries to include the new request in both parts of the current solution (Solomon, 1987). If some of the k best insertions are in the first part of the current solution, the job is offered immediately to the corresponding couriers; if one of these couriers accepts, the request is allocated immediately. Otherwise, the new request will be managed by the background optimization procedure.

10.2.4.6 The background optimization procedure

The background optimization procedure is performed through a Tabu Search working on the second part of the current solution and on the location of the "parking areas". In the second part of the current solution, the one involving jobs not known by couriers yet, we consider a neighborhood made up of the solutions which can be obtained by removing a request from a route and inserting it into the same route in another position or into another route. A request moved at iteration i is tabu until iteration $i+\theta$, where θ is a random number ~ $[\theta_1, \theta_2]$ (θ_1=5, θ_2=12). Moreover, a continuous diversification scheme is used in order to discourage frequently moving the same request. As for the procedure for locating the "parking areas", it is inspired by the "stochastic median policy" which is optimal under light traffic. In this procedure, given the best current solution found so far, including both its first and second part, we determine parking areas by using a heuristic for the p-median problem and we then allocate the p couriers to the p-medians by solving an "assignment" problem. It is worth saying that, in most cases, the relocation will not be implemented because of the arrival of new requests.

10.2.5 Parallelization Strategy

As said before, when solving a real-time routing and dispatching problem with several hundred couriers, a parallel implementation is needed in order to make route re-optimization computation time acceptable. A taxonomy of various parallel tabu search algorithms can be found in Crainic *et al.* (1997). We use an asynchronous single-point multiple strategy parallelization strategy in which each process is coded in Java.

10.3 LITERATURE REVIEW

In the broad category of real-time fleet management vehicle routes are built in an on-going fashion as customer requests, vehicle locations and travel times are revealed over the planning horizon. A large part of the current literature is characterized by algorithms reacting to new requests only once they have occurred, while neglecting available stochastic information. Overviews of these problems can be found in Powell, Jaillet and Odoni (1995), Psaraftis (1988, 1995), Gendreau and Potvin (1998), and Ghiani et al (2003). In the article by Mitrović-Minić and Laporte (2004) four waiting strategies are examined for the dynamic *Pickup and Delivery Problem with Time Windows* (PDPTW). In the dynamic PDPTW, the presence of time windows allows the vehicles to wait at various locations along their routes. The authors show that an adequate distribution of this waiting time may

affect the planner's ability to make good decisions at a later stage. More recently, Branke *et al.* (2005) have analyzed waiting strategies in dynamic vehicle routing problems without time windows in contexts where the objective is to maximize the probability that an additional customer can be integrated into a fixed tour without violating time constraints. The authors have proposed several waiting strategies as well as an evolutionary algorithm to optimize the selected waiting strategy. Computational results have shown that an appropriate waiting strategy can both increase the probability of being able to serve an additional customer and decrease the average length of detours.

Another line of research examines dispatching and routing policies whose performance can be determined analytically if specific assumptions are satisfied. See, e.g., Bertsimas and van Ryzin (1991, 1993), where demands are distributed in a bounded area in the plane and arrival times are modeled as a Poisson process. The authors identify optimal policies both in light and heavy traffic cases. Papastavrou (1996) describes a routing policy that performs well both in light and heavy traffic, while Swihart and Papastavrou (1999) examine a dynamic pickup and delivery extension.

A related area of research, often referred to as *Stochastic Vehicle Routing*, examines problems in which demand becomes known at the beginning of each day. In this context the problem is to determine, on the basis of a probabilistic characterization of random data, a solution of least expected cost in which the order of customers is fixed regardless of the demand realization for a particular day (an *a priori* solution). Jaillet (1988) introduced the *Probabilistic Traveling Salesman Problem*, Jaillet and Odoni (1988) examined the capacitated case, while Bertsimas (1992) introduced the multi-vehicle stochastic vehicle routing problem. A survey of the research in this area can be found in Powell, Jaillet and Odoni (1995), Bertsimas and Simchi-Levi (1996), and Gendreau, Laporte and Séguin (1996).

Finally, we note the existence of a relatively recent line of research, known as *anticipatory routing*, in which *general probability distributions* are used in order to devise exact or heuristic policies. To our knowledge, four papers take this approach. Powell *et al.* (1988) introduce a truckload dispatching problem, and Powell (1996) provides formulations, solution methods, as well as numerical results. In these papers, future demand forecasts are used to determine which loads should be assigned to the vehicles in a truckload environment to account for forecasted capacity needs in the next period. In Thomas and White (2004) a vehicle may serve several requests at a time and may wait for future demand both at a customer and non-customer locations. Not all requests have to be serviced and the objective function to be minimized is the expected value of a combination of travel costs, terminal costs, and revenue generated from a pickup. Bent and

Van Hentenryck (2004) consider a vehicle routing problem where customer locations and service times are random variables which are realized dynamically during plan execution. They develop a multiple scenario approach which continuously generates plans consistent with past decisions and anticipating future requests.

10.4 CONCLUSIONS

In this chapter we have described an innovative real-time fleet management system designed and implemented for eCourier Ltd (London, UK). We have described both the business challenges and benefits of the implementation of a real-time fleet management system (with reference to empirical metrics such as courier efficiency, service times, and financial data), as well as the theoretical and implementation challenges of constructing such a system. The use of our system has allowed the company to reduce the requirements of human supervisors for fleet management, to improve service and to increase courier efficiency. In particular, if we compare eCourier to its competitors in terms of administration cost per courier per year (figure 10-10), we can observe that for conventional couriers growth is not beneficial, because the smaller the fleet, the more profitable the company, whereas for eCourier growth improves operating margin with every delivery, obtaining a sustainable growth which is near exponential.

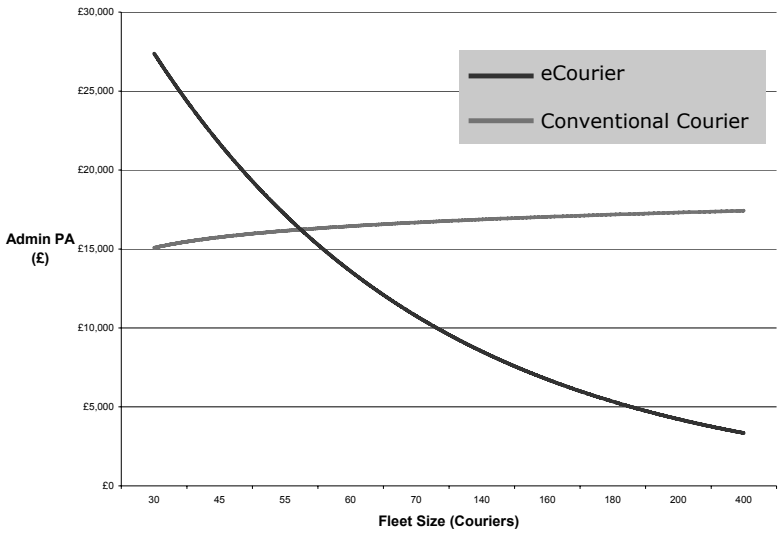


Figure 10-10. Administration cost per courier per year.

REFERENCES

Bent, R. W., and Van Hentenryck, P., 2004, Scenario-based planning for partially dynamic vehicle routing with stochastic customers, *Operations Research*. **52**:977-987.

Bertsekas, D. P., 1995, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, MA.

Bertsimas, D. J., 1992, A vehicle routing problem with stochastic demand, *Operations Research*. **40**:574–585.

Bertsimas, D. J., and van Ryzin, G., 1991, A stochastic and dynamic vehicle routing problem in the Euclidean plane, *Operations Research*. **39**:601–615.

Bertsimas, D. J., and van Ryzin, G., 1993, Stochastic and dynamic vehicle routing problem in the Euclidean plane with multiple capacitated vehicles, *Operations Research*. **41**:60–76.

Bertsimas, D. J., and Simchi-Levi, D., 1996, A new generation of vehicle routing research: robust algorithms, addressing uncertainty, *Operations Research*. **44**:286–303.

Birge, J. R., and Louveaux, F. V., 1997, *Introduction to Stochastic Programming*, Springer-Verlag, New York.

Branke, J., Middendorf, M., Noeth, G., and Dessouky, M., 2005, Waiting strategies for dynamic vehicle routing, *Transportation Science*. **39**:298–312.

Cathey, F. W., and Dailey, D. J., 2003, A prescription for transit arrival/departure prediction using automatic vehicle location data, *Transportation Research Part C: Emerging Technologies*. **11**:241–264.

Chen, Z. L., and Xu, H., 2006, Dynamic column generation for dynamic vehicle routing with time windows, *Transportation Science*. **40**(1):74–88.

Crainic, T. G., Toulouse, M., and Gendreau, M., 1997, Towards a taxonomy of parallel tabu search algorithm, *INFORMS Journal on Computing*. **9**:61–72.

Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F., 2001, VRP with pickup and delivery, in: *The vehicle routing problem*, P. Toth and D. Vigo eds., SIAM Monographs on Discrete Mathematics and Applications, SIAM Publishing, Philadelphia, pp. 225-242.

Fabri, A., and Recht, P., 2006, On dynamic pickup and delivery vehicle routing with several time windows and waiting times, *Transportation Research Part B: Methodological*. **40**(4):335–350.

Gendreau, M., Laporte, G., and Séguin, R., 1996, Stochastic vehicle routing, *European Journal of Operational Research*. **88**:3–12.

Gendreau, M., and Potvin, J. Y., 1998, Dynamic routing and dispatching, in: *Fleet Management and logistics*, T. G. Crainic and G. Laporte eds., Kluwer, Boston, pp. 115-126.

Ghiani, G., Guerriero, F., Laporte, G., and Musmanno, R., 2003, Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies, *European Journal of Operational Research*. **151**:1-11.

Hertz, A., Laporte, G., and Nanchen-Hugo, P., 1999, Improvement procedures for the undirected rural postman problem, *INFORMS Journal on Computing*. **11**:53–62.

Jaillet, P., 1988, A priori solution of the traveling salesman problem in which a random subset of customers are visited, *Operations Research*. **36**:929–936.

Jaillet, P., and Odoni, A. R., 1988, The probabilistic vehicle routing problem, in: *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad eds., North-Holland, Amsterdam, The Netherlands, pp. 293–318.

Mitrović-Minić, S., and Laporte, G., 2004, Waiting strategies for the dynamic pickup and delivery problem with time windows, *Transportation Research Part B: Methodological*. **38**:635-655.

Montgomery, D. C., Johnson, L. A., and Gardiner, J. S., 1990, *Forecasting and Time Series Analysis*, 2nd ed., McGraw-Hill, New York.

Papastavrou, J. D., 1996, A stochastic and dynamic routing policy using branching processes with state dependent migration, *European Journal of Operational Research*. **95**:167–177.

Powell, W. B., Sheffi, Y., Nickerson, K. S., Butterbaugh, K., and Atherton, S., 1988, Maximizing profits for North American Van Lines' truckload division: a new framework for pricing and operations, *Interfaces*. **18**(1):21–41.

Powell, W. B., Jaillet, P., and Odoni, A. R., 1995, Stochastic and dynamic networks and routing, in: *Handbook in OR and MS, Volume 8: Network Routing*, M. Ball, T. Magnanti, C. Monma, and G. Nemhauser eds., Elsevier Science, Amsterdam, pp. 141–295.

Powell, W. B., 1996, A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers, *Transportation Science*. **30**:195–219.

Psaraftis, H. N., 1988, Dynamic vehicle routing problems, in: *Vehicle Routing: Methods and Studies*, B. L. Golden and A. A. Assad eds., North-Holland, Amsterdam, pp. 223–248.

Psaraftis, H. N, 1995, Dynamic vehicle routing: status and prospects, *Annals of Operations Research*. **61**:143–164.

Puterman, M. L., 1994, *Markov Decision Processes*, Wiley, New York.

Resende, M., Pardalos, P., and Eksioglu, S., 1999, Parallel metaheuristics for combinatorial optimization, *Journal of Heuristics*. **3**:44–62.

Solomon, M. M., 1987, Algorithms for the vehicle-routing and scheduling problems with time window constraints, *Operations Research*. **35**(2):254–265.

Sennott, L. I., 1999, *Stochastic Dynamic Programming and the Control of Queueing Systems*, Wiley, New York.

Swihart, M. R., and Papastavrou, J. D., 1999, A stochastic and dynamic model for the single-vehicle pick-up and delivery problem, *European Journal of Operational Research*. **114**:447–464.

Thomas, B. W., and White III, C. C., 2004, Anticipatory route selection, *Transportation Science*. **38**:473-487.

INDEX