HANDBOOKS IN
OPERATIONS RESEARCH AND
MANAGEMENT SCIENCE

Volume 13

SIMULATION

S.G. Henderson B.L. Nelson
Editors

NORTH-HOLLAND

HANDBOOKS IN OPERATIONS RESEARCH
AND MANAGEMENT SCIENCE
VOLUME 13

Handbooks in Operations Research and Management Science

Editors

J.K. Lenstra
Centrum voor Wiskunde en
Informatica, Amsterdam

G.L. Nemhauser
Georgia Institute of Technology

Volume 13

Amsterdam – Boston – Heidelberg – London – New York – Oxford
Paris – San Diego – San Francisco – Singapore – Sydney – Tokyo
North-Holland is an imprint of Elsevier

Simulation

Edited by

Shane G. Henderson
Cornell University, Ithaca, NY, USA

Barry L. Nelson
Northwestern University, Evanston, IL, USA

Amsterdam – Boston – Heidelberg – London – New York – Oxford
Paris – San Diego – San Francisco – Singapore – Sydney – Tokyo
North-Holland is an imprint of Elsevier

North-Holland is an imprint of Elsevier
Radarweg 29, PO Box 211, 1000 AE Amsterdam, The Netherlands
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, UK

First edition 2006

Notice
No responsibility is assumed by the publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. Because of rapid advances in the medical sciences, in particular, independent verification of diagnoses and drug dosages should be made

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN-13: 978-0-444-51428-8
ISBN-10: 0-444-51428-7

ISSN: 0927-0507

Printed and bound in The Netherlands

06 07 08 09 10 10 9 8 7 6 5 4 3 2 1

Dedication

On November 16, 2005, our friend and colleague, Perwez Shahabuddin, died tragically. Perwez was a renowned researcher whose contributions will have a lasting impact. He was also a caring and dedicated teacher. As one of his students said, "…he was the definition of a good professor". Perwez had a gentle nature and an infectious laugh that we remember with fondness.

We dedicate this Handbook to his memory.

Shane G. Henderson
Barry L. Nelson

This page intentionally left blank

Contents

Dedication v

CHAPTER 1
Stochastic Computer Simulation
Shane G. Henderson and Barry L. Nelson 1

1 Scope of the Handbook 1
2 Key concepts in stochastic simulation 4
3 Organization of the Handbook 17
Acknowledgements 18
References 18

CHAPTER 2
Mathematics for Simulation
Shane G. Henderson 19

1 Introduction 19
2 Static simulation: Activity networks 21
3 A model of ambulance operations 27
4 Finite-horizon performance 28
5 Steady-state simulation 34
Acknowledgements 50
Appendix: Proof of Proposition 15 50
References 52

CHAPTER 3
Uniform Random Number Generation
Pierre L'Ecuyer 55

1 Introduction 55
2 Uniform random number generators 56
3 Linear recurrences modulo *m* 60
4 Generators based on recurrences modulo 2 69
5 Nonlinear RNGs 75
6 Empirical statistical tests 76
7 Conclusion, future work and open issues 77
Acknowledgements 78
References 78

CHAPTER 4
Nonuniform Random Variate Generation
Luc Devroye 83

1 The main paradigms 83
2 Uniformly bounded times 94
3 Universal generators 97
4 Indirect problems 99
5 Random processes 107
6 Markov chain methodology 108
Acknowledgement 116
References 116

CHAPTER 5
Multivariate Input Processes
Bahar Biller and Soumyadip Ghosh 123

1 Introduction 123
2 Constructing full joint distributions 126
3 Parametric families of joint distributions 133
4 Constructing partially specified joint distributions 136
5 Conclusion 149
References 150

CHAPTER 6
Arrival Processes, Random Lifetimes and Random Objects
Lawrence M. Leemis 155

1 Arrival processes 155
2 Generating random lifetimes 167
3 Generating random objects 172
Acknowledgements 178
References 178

CHAPTER 7
Implementing Representations of Uncertainty
W. David Kelton 181

1 Introduction 181
2 Random-number generation 182
3 Random-structure generation 184
4 Application to variance reduction 186
5 Conclusions and suggestions 189
References 191

CHAPTER 8
Statistical Estimation in Computer Simulation
Christos Alexopoulos 193

1 Introduction 193
2 Background 195

3 Sample averages and time averages 196
4 Stationary processes 199
5 Analyzing data from independent replications 204
6 Density estimation 210
7 Summary 220
Acknowledgements 222
References 222

CHAPTER 9
Subjective Probability and Bayesian Methodology
Stephen E. Chick 225

Introduction 225
1 Main concepts 227
2 Computational issues 237
3 Input distribution and model selection 239
4 Joint input–output models 240
5 Ranking and selection 245
6 Discussion and future directions 252
Acknowledgement 253
References 253

CHAPTER 10
A Hilbert Space Approach to Variance Reduction
Roberto Szechtman 259

1 Introduction 259
2 Problem formulation and basic results 260
3 Hilbert spaces 263
4 A Hilbert space approach to control variates 271
5 Conditional Monte Carlo in Hilbert space 273
6 Control variates and conditional Monte Carlo from a Hilbert space perspective 274
7 Weighted Monte Carlo 275
8 Stratification techniques 278
9 Latin hypercube sampling 281
10 A numerical example 286
11 Conclusions 287
Acknowledgements 288
References 288

CHAPTER 11
Rare-Event Simulation Techniques: An Introduction and Recent Advances
S. Juneja and P. Shahabuddin 291

1 Introduction 291
2 Rare-event simulation and importance sampling 296
3 Rare-event simulation in a Markovian framework 302
4 Large deviations of multidimensional random walks 309

5 Adaptive importance sampling techniques 316
6 Queueing systems 327
7 Heavy-tailed simulations 330
8 Financial engineering applications 335
Acknowledgement 346
References 346

CHAPTER 12
Quasi-Random Number Techniques
C. Lemieux 351

1 Introduction 351
2 An example 356
3 A key concept: Effective dimension 359
4 Constructing quasi-random point sets 364
5 Recurrence-based point sets 369
6 Randomization techniques and variance results 371
7 Combination with other variance reduction techniques 374
8 Future directions 375
Acknowledgements 376
References 376

CHAPTER 13
Analysis for Design
Ward Whitt 381

1 Introduction 381
2 The standard statistical framework 385
3 The asymptotic parameters for a function of a Markov chain 393
4 Birth-and-death examples 397
5 Diffusion processes 401
6 Stochastic-process limits 405
7 Deleting an initial portion of the run to reduce bias 410
8 Directions for further research 411
Acknowledgement 411
References 411

CHAPTER 14
Resampling Methods
R.C.H. Cheng 415

1 Introduction 415
2 The bootstrap 417
3 Quantiles and confidence intervals 422
4 Theory 428
5 Simulation models 436
6 Bootstrap comparisons 443
7 Bayesian models 446

8 Time series output 448
9 Final comments 451
References 451

CHAPTER 15
Correlation-Based Methods for Output Analysis
David Goldsman and Barry L. Nelson 455

1 Introduction 455
2 Motivation 457
3 Estimators using nonoverlapping batches 459
4 Estimators from overlapping batches 468
5 Summary and conclusions 473
Acknowledgement 474
References 474

CHAPTER 16
Simulation Algorithms for Regenerative Processes
Peter W. Glynn 477

1 Introduction 477
2 The steady-state simulation problem 478
3 The regenerative estimator for the TAVC 480
4 Choice of the optimal regeneration state 483
5 The regenerative approach to the initial transient and initial bias problems 484
6 When is a simulation regenerative? 487
7 When is a GSMP regenerative? 489
8 Algorithmic identification of regenerative structure 490
9 A martingale perspective on regeneration 493
10 Efficiency improvement via regeneration: Computing steady-state gradients 496
11 Efficiency improvement via regeneration: Computing infinite horizon discounted reward 498
References 499

CHAPTER 17
Selecting the Best System
Seong-Hee Kim and Barry L. Nelson 501

1 Introduction 501
2 Basics of ranking and selection 502
3 Simulation issues and key results 508
4 Example procedures 517
5 Application 521
6 Asymptotic analysis 522
7 Other formulations 526
8 Future directions 531
Acknowledgements 532
References 532

CHAPTER 18
Metamodel-Based Simulation Optimization
Russell R. Barton and Martin Meckesheimer 535

1 Introduction 535
2 Metamodels and simulation 538
3 Metamodel-based optimization 545
4 Response surface methodology (RSM) 548
5 Global metamodel-based optimization 563
6 Summary 569
Acknowledgements 570
References 570

CHAPTER 19
Gradient Estimation
Michael C. Fu 575

1 Introduction 575
2 Gradient-based simulation optimization 577
3 Indirect gradient estimation 580
4 Direct gradient estimation 583
5 Examples 598
6 Basic theoretical tools 605
7 Simple guidelines for the simulation practitioner 606
8 Applications 607
9 Probing further 609
10 Future research directions 611
Acknowledgements 612
References 612

CHAPTER 20
An Overview of Simulation Optimization via Random Search
Sigrún Andradóttir 617

1 Introduction 617
2 A brief review of random search methods 619
3 Convergence 621
4 Efficiency 624
5 Summary 629
Acknowledgements 630
References 630

CHAPTER 21
Metaheuristics
Sigurdur Ólafsson 633

1 Introduction 633
2 Background to metaheuristics 635
3 Accounting for simulation noise 640

4 Genetic algorithm 643
5 Tabu search 644
6 The nested partition method 646
7 Making convergence statements 647
8 Future directions 652
References 653

Author Index 655

Subject Index 667

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13001-7

Chapter 1
Stochastic Computer Simulation

Shane G. Henderson

School of Operations Research and Industrial Engineering, Cornell University, USA
E-mail: sgh9@cornell.edu

Barry L. Nelson

Department of Industrial Engineering and Management Sciences, Northwestern University, USA
E-mail: nelsonb@northwestern.edu

Abstract

We introduce the topic of this book, explain what we mean by stochastic computer simulation and provide examples of application areas. We motivate the remaining chapters in the book through two in-depth examples. These examples also help clarify several concepts and techniques that are pervasive in simulation theory and practice.

1 Scope of the Handbook

What is "stochastic computer simulation?" Perhaps the most common example in everyday life is an electronic game, such as Solitaire or Yahtzee, that depends on a source of randomness to imitate shuffling cards, rolling dice, etc. The fidelity of the electronic game, which is a simulation of the physical game, depends on a faithful imitation of the physical source of randomness. The electronic game is useless (and no fun) otherwise. Of course, an electronic game usually needs a game player. If you replace the player by an algorithm that plays the game, and you compare different algorithms by playing many sessions of the game, then you have a pretty good representation of what stochastic computer simulation is and how it is used in operations research and the management sciences.

This book is a collection of chapters on key issues in the design and analysis of computer simulation experiments on models of stochastic systems. The chapters are tightly focused and written by experts in each area. For the purposes of this volume, "stochastic computer simulation" (henceforth just "stochastic simulation") refers to the analysis of stochastic processes through the generation of sample paths (realizations) of the processes. We restrict attention to design and analysis issues, and do not address the equally important problems of representations (modeling) and execution (see, for instance, Fishwick, 1995). In broad terms, the goal of this volume is to survey the concepts, principles, tools and techniques that underlie the theory and practice of stochastic simulation design and analysis, emphasizing ideas and methods that are likely to remain an intrinsic part of the foundation of the field for the foreseeable future. The chapters provide an up-to-date reference for both the simulation researcher and the advanced simulation user, but they do not constitute an introductory level "how to" guide. (See, instead, Banks (1998), Banks et al. (2004), Law and Kelton (2000) or Fishman (2001). The latter book is at a slightly more advanced level than the others.)

Computer scientists, financial analysts, industrial engineers, management scientists, operations researchers and many other professionals use stochastic simulation to design, understand and improve communications, financial, manufacturing, logistics and service systems. A theme that runs throughout these diverse applications is the need to evaluate system performance in the face of uncertainty, including uncertainty in user load, interest rates, demand for product, availability of goods, cost of transportation and equipment failures. Much like the electronic game designer, stochastic simulation users develop models that (they hope) faithfully represent the sources of uncertainty in the systems that they simulate. Unlike the game designer, the simulation user also needs to *design the simulation experiment* – decide what cases to simulate, and how much simulation to do – and *analyze the results*.

Later in this chapter we provide two examples of the types of problems that are solved by stochastic simulation. The examples motivate and provide context for the remaining chapters in the book. We do not attempt – either in this chapter or in the remainder of the book – to cover the wide range of applications of simulation. A few important application areas are listed below.

Financial engineering/quantitative finance: The classical problem in this area is valuing a derivative, which is a financial instrument whose value depends on an underlying asset such as a stock or bond. Uncertainty in the value of the asset over time makes simulation a natural tool. Other problems in this vein include estimating the value-at-risk of a portfolio and designing hedging strategies to mitigate risk. See, for instance, Glasserman (2004).

Computer performance modeling: From the micro (chip) level to the macro (network) level, computer systems are subject to unpredictable loads and unexpected failures. Stochastic simulation is a key tool for designing and tuning computer systems, including establishing expected response times from a storage device, evaluating protocols for web servers, and testing the execution of real-time control instructions. See, for instance, Jain (1991).

Service industries: Service industries include call/contact centers, an application in which simulation has been used to design staffing and call-routing policies. Service applications emphasize delivering a specified level of service with a high probability; such issues arise in food delivery, financial services, telecommunications and order fulfillment, for instance, and simulation helps to ensure quality service in the face of uncertainty.

Manufacturing: Stochastic simulation has seen extensive use in manufacturing applications. A few representative contributions include evaluation of production scheduling algorithms; work-center design and layout; estimation of cycle time-throughput curves; and evaluation of equipment replacement and maintenance policies.

Military: Military applications include life-cycle management of defense systems; combat and munitions logistics; crisis communications evaluation; and modeling network architectures for command and control.

Transportation and logistics: Simulation has been used to evaluate the effectiveness of Internet and cell-phone-based traveler information services; to perform benefits analyses for regional Intelligent Transportation Systems; and to design public transportation systems for the elderly and disabled. When the transportation system is part of a supply chain, simulation may be used to determine replenishment policies for manufacturers, distribution centers and retailers, or to estimate the impact of demand uncertainty on supplier inventories.

The *Proceedings* of the annual Winter Simulation Conference is an outstanding source of applications and success stories from these and other areas. The *Proceedings* from 1997 through the present can be found at *http://www.wintersim.org*.

The focus of this volume is narrow, by necessity, because the label “computer simulation” is attached to a number of activities that do not fall under the umbrella of “generating sample paths of stochastic processes”. For instance, there is a vast literature on, and countless applications of, simulation of dynamic systems that are represented by differential and partial differential equations. Such systems may be stochastic, but the approach is to numerically integrate the system of equations through time to determine levels, probabilities, moments etc., rather than generating sample paths and averaging across repetitions as we do. Systems dynamics (see Sterman, 2000) is a popular approach for developing and analyzing differential equation models, and there is a substantial intersection between applications of systems dynamics and applications of stochastic simulation in operations research and the management sciences.

Although closely related, we do not consider issues that arise in person-in-the-loop simulations that are used for, among other things, training of personnel and evaluation of supervisory systems prior to insertion in the field. A key feature of stochastic simulation is that the source of randomness is under the control of the experimenter, which is not the case when a person is incorporated into the experiment. We also do not cover any type of computer animation, up to and including virtual reality, although this is certainly a kind of computer simulation. Stochastic simulation is sometimes a driving process for computer-generated animation, however. Nor do we consider the important area of parallel and distributed simulation; see Fujimoto (1999).

The next section introduces the key concepts we do cover via two examples.

2 Key concepts in stochastic simulation

There are a number of key simulation-related concepts that feature throughout this volume. The goal of this section is to introduce and explain those concepts through the use of two examples, and to link them to subsequent chapters. The reader familiar with stochastic simulation may still find it useful to briefly peruse this section to avoid potential confusion associated with slightly different uses of terminology. In preparing this section we have occasionally relied quite heavily on Nelson (2002, Chapter 4).

Discrete-event systems dominate simulation applications and so it is important to gain an understanding of simulation concepts related to such models. The processing-network example described in Section 2.1 is a special case. Many of the concepts introduced with this model extend to simulations of a broader class of models. The stochastic activity network example described in Section 2.2 reinforces some of the concepts introduced with the processing-network example, and brings out several new ones.

2.1 A processing network problem

A manufacturing system processes two classes of jobs with two processing stations (machines). Class 1 jobs are high-value jobs that need to be completed quickly. Class 2 jobs are of lower value, and while these jobs should be processed quickly, greater delays can be tolerated than with Class 1 jobs. Accordingly, the processing system has been set up as in Figure 1. Station 1 processes only Class 1 jobs. Station 2 is capable of processing both Class 1 and Class 2 jobs, so that it can assist Station 1 to complete the processing of Class 1 jobs. This network was first introduced and analyzed in Harrison (1996) and since then has become a standard example in the study of multiclass processing networks.

A *policy* is a strategy describing how the two stations coordinate their efforts in processing the incoming jobs. A key question is what policy should be used to ensure that Class 1 jobs are delayed for as short a period of time as possible, while ensuring that Class 2 jobs are also completed in a reasonable amount of time. A natural policy one might consider is for Station 1 to process jobs whenever they are available, and for Station 2 to give nonpreemptive priority to Class 1 jobs over Class 2 jobs. One might ask how this policy performs. It turns out that there is a rather surprising answer to this question. We will explain how simulation can be used both to develop an understanding of this system, and to explore a range of operating policies. To perform simulations of the model, we need to further specify its structure and decide how and what to simulate.

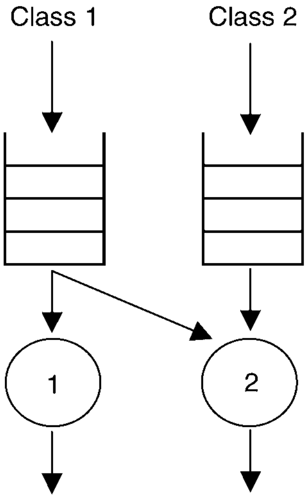


Fig. 1. Processing network.

A common approach for specifying simulation models is the *process-interaction view*. In this approach, one specifies the various entities that interact with a system and describes the processes they follow. In our example the entities are jobs of Class 1 and Class 2. Class 1 jobs arrive according to a certain arrival process and potentially wait in a queue of Class 1 jobs. Once they reach the head of the queue, they are served by either Station 1 or Station 2, whichever becomes available first. If both are available then they are served by Station 1. After service, Class 1 jobs depart the system. Class 2 jobs arrive according to a certain arrival process, potentially wait in a queue of Class 2 jobs, and then are served by Station 2 once the queue of Class 1 jobs is empty and Station 2 is available. They then depart the system.

This very natural description of the model helps to explain the popularity of the process-interaction view. Of course, the description of the model is incomplete, since we have not specified the arrival and service processes. Suppose that the times between arrivals of Class 1 jobs are i.i.d. random variables with distribution function F_{A1}. Suppose that, independent of Class 1 arrivals, the sequence of Class 2 interarrival times, the sequence of service times at Station 1, and the sequence of service times at Station 2 are mutually independent i.i.d. sequences of random variables with interevent distribution functions F_{A2}, F_{S1} and F_{S2}, respectively. Notice that Class 1 jobs and Class 2 jobs have the same service time distribution function F_{S2} at Station 2.

The sequence of interarrival times of Class 1 jobs is known as a *stationary* arrival process because all of the interarrival times have the same distribution. This is often a reasonable assumption in manufacturing models where considerable effort is exercised by managers to maintain a smooth flow of work through the system. However, in service systems where "jobs" are, in fact, people, this assumption is often violated. Customer arrivals often exhibit significant "time-of-day" effects where arrivals are more frequent at certain times of the day. Such an arrival process is called "nonstationary", since the distribution of the time between customer arrivals can vary. Chapter 6 describes methods for modeling and simulating nonstationary processes. For now we maintain our assumption that the arrival and service processes consist of i.i.d. sequences of random variables.

The process-interaction view is one example of a *world view*, which is a general approach for specifying models for the purposes of simulation; see, e.g., Banks et al. (2004) and Law and Kelton (2000). The *event-scheduling world view* is another world view that closely mirrors the algorithm underlying most simulation software. We now give an event-scheduling view of the model that is couched in a discussion of how to simulate the model.

The key quantity to track is the vector $\mathbf{X} = (X_1, X_2, J_2)$ consisting of the number of Class 1 and Class 2 jobs in the system, and the class of the job in service at Station 2. If no job is in service at Station 2 then we set $J_2 = 0$. The vector $\mathbf{X}$ is known as the *system state* and it takes on values in the *state space*

$$\{0, 1, \ldots\} \times \{0, 1, \ldots\} \times \{0, 1, 2\}.$$

The state is modified by *events* which are arrivals of Class 1 or Class 2 jobs, and service completions at Station 1 or Station 2. These events occur at discrete epochs (points in time), and between event epochs the system state remains constant. Associated with each event is a *clock* that indicates the time that event is next scheduled to occur. The next event to occur is the one with the smallest clock reading. The simulation jumps immediately to the time of that event and then updates the system state and event clocks accordingly. For example, at the time of a Class 1 arrival, we first increment X_1 and schedule the next Class 1 arrival by generating a sample value from F_{A1}. Furthermore, if Station 1 is available, we start service on the job by setting the clock for service completion at Station 1 to the current time plus a sample from the distribution F_{S1}. If Station 1 is not available and Station 2 is, then we set $J_2 = 1$ and set the clock for a service completion at Station 2 to the current time plus a sample from F_{S2}.

At the time of any service-completion event we follow the logic dictated by the policy to determine which events to schedule next. For example, suppose that Station 2 completes a job of Class 2 (so that $J_2 = 2$). We then decrement X_2 and set J_2 to be either 0, 1 or 2 according to whether there are no jobs of either class waiting, there are Class 1 jobs waiting, or there are no Class 1 jobs waiting and one or more Class 2 jobs waiting. In the second or third case, the clock associated with service completions at Station 2 is set to the current time plus a sample from the distribution F_{S2}. When no job is in service at a station, there should be no service-completion event scheduled at that station. In that case we set the clock reading for the event to ∞. It then plays no role in the simulation until a job enters service at that station, at which time the clock is reset. All that is needed now to set the simulation in motion is a specification of the initial system state and event-clock settings.

Notice that we need to be able to generate samples from a given distribution. This problem is almost invariably solved in two steps. In the first step we generate a sequence of numbers that mimics many of the properties of an i.i.d. sequence $U = (U(n)\colon\ n \geqslant 0)$ of uniformly distributed random numbers on $(0, 1)$; see Chapter 3. We then generate the desired samples assuming the existence of the sequence U using a variety of methods. See the remaining Chapters 4–7 in Part II.

The event-scheduling view of the model is perhaps not as intuitive as the process-interaction view. As noted in Nelson (2002, Section 4.7) the event-scheduling approach "…*defines* system events by describing what happens to the system as it encounters entities. The *process view* of stochastic processes *implies* system events by describing what happens to an entity as it encounters the system". A modeler is often able to visualize the progress of an entity through a system, and so the process-interaction view can be more intuitive than the event-scheduling world view.

The event-scheduling view is very similar to the algorithm used to simulate our model in most discrete-event simulation software. The discrete-event simulation algorithm for a general model can be approximately described as follows.

1. Initialization: Set the simulation clock T to 0. Choose the initial system state X and event clock readings $\{C_i\}$ say.
2. Let $T = \min_i C_i$ be advanced to the time of the next event and let I be the index of the clock reading that achieves this minimum.
3. Execute the logic associated with event I, including updating the system state X and event clocks $\{C_i\}$.
4. Go to Step 2.

Notice that even this simple description of the algorithm embodies several key issues.

- The system is tracked through a system state and clocks indicating the time at which events of different types are scheduled to occur.
- The simulation proceeds from event epoch to event epoch, rather than continuously in time.
- At event epochs the system state and clocks are updated, and any new events are scheduled by generating samples from appropriate distributions.

Our description of discrete-event simulation is somewhat specialized and imprecise, since to give a more general and precise definition would take us too far afield. The interested reader can find more general and precise descriptions in, for example, Shedler (1993), Glynn (1989) and Haas (1999).

The interarrival and service time sequences defined thus far are all *input* stochastic processes, in that they are specified by the simulation user. The logic of the simulation model combines these input stochastic processes to form *output* stochastic processes. Output processes then shed light on the behavior of the system under study. Even for the simple model given above there are a host of stochastic processes one might consider. There are three processes that deserve special attention.

- Let $T(0) = 0$ and let $T(n)$ denote the time of the nth event epoch, $n \geqslant 1$. This value can be read from the simulation clock T just after Step 2 of the algorithm is completed. We call $(T(n)\colon\ n \geqslant 0)$ the *event-epoch* process.
- Let $X(0)$ be the initial system state, and let $X(n)$ be the system state immediately after the nth event epoch. Notice that $X(n)$ is the value of the system state immediately after Step 3 of the algorithm is completed. We call $(X(n)\colon\ n \geqslant 0)$ the *state-change* process.
- Let $Y(t)$ denote the state of the system at simulated-time t, and let $(Y(t)\colon\ t \geqslant 0)$ be the *state* process. We can recover this process from the event-epoch and state-change processes via $Y(t) = X(n)$, where n is chosen so that $T(n) \leqslant t < T(n+1)$.

The state process gives a continuous view of the system state with time. In our problem we are also very interested in the waiting times in queue of the two classes of jobs. In particular, we might also be interested in the stochastic process $(W_1(n)\colon\ n \geqslant 1)$, where $W_1(n)$ is the waiting time in queue of the nth Class 1 job to arrive to the system. It is difficult to construct this stochastic process from the event-epoch and state-change processes. Perhaps the easiest way to do this is to “tag” each Class 1 job as it arrives with the (simulated) time of its arrival. When the job enters service one can then subtract the job’s arrival time from the current simulated time to give the waiting time in the queue. It is therefore straightforward to recover the waiting time sequence. These sorts of calculations are usually performed automatically by discrete-event simulation software.

Many other stochastic processes can be constructed in a similar fashion. The point behind constructing such stochastic processes is to enable the estimation of various *performance measures* that describe some aspect of the simulated system. A performance measure is essentially a summary statistic related to one or more of the stochastic processes described above. The choice of performance measure is invariably related to the questions one has in mind when a simulation model is built. In our example it makes sense to use measures related to the waiting times of jobs, but there are many possible performance measures. For example, we may be interested in any or all of the following.

PM1. For some fixed $t > 0$ let

$$\overline{Y}_1(t) = \frac{1}{t}\int_0^t Y_1(s)\,\mathrm{d}s$$

be the average number of Class 1 jobs in the system over the first t units of simulated time, where $Y_1(s)$ is the number of Class 1 jobs in the system at time s. Notice that $\overline{Y}_1(t)$ is a random variable that depends on the generated sample path. The first performance measure is $\mathrm{E}\overline{Y}_1(t)$.

PM2. For some fixed $n \geqslant 1$ let

$$\overline{W}_1(n) = \frac{1}{n}\sum_{i=1}^{n} W_1(i)$$

be the mean waiting time in queue of the first n Class 1 jobs to arrive to the system. The second performance measure is $\mathrm{E}\overline{W}_1(n)$ for some fixed n.

PM3. Instead of expected values, we may wish to compute a tail probability such as $\Pr(\overline{W}_1(n) > 10)$.

PM4. The pth quantile of a random variable Z can be defined as $\inf\{z\colon \Pr(Z \leqslant z) \geqslant p\}$. We may wish to compute the pth quantile of $\overline{Y}_1(t)$ for some fixed p.

These are all examples of *finite-horizon* performance measures. An alternative term that is often used is *terminating-simulation* performance measures. The distinguishing feature of such measures is that they are related to the distribution of a sample path of finite, but possibly random, length. For example, PM1 requires one to simulate for a fixed length of simulated time, while PM2 requires one to simulate for an amount of simulated time that will vary from sample path to sample path. Both are finite-horizon performance measures. Both PM1 and PM2 are expected values of a particular random variable. One might also be interested in the distributions of these random variables beyond the mean. Both PM3 and PM4 provide such information.

Chapter 8 discusses how to estimate performance measures. Perhaps the most straightforward method is the *replication method*, which works as follows. We have previously described a "recipe" for generating a sample path over a finite interval. If this recipe is repeated n independent times, then one obtains n i.i.d. observations of the sample path. The replication method combines these i.i.d. observations to estimate the performance measure of interest. For example, if $\overline{Y}_1(t; i)$ is the value of $\overline{Y}_1(t)$ on the ith replication, then one can estimate $\mathrm{E}\overline{Y}_1(t)$ via the sample average

$$\frac{1}{n}\sum_{i=1}^{n} \overline{Y}_1(t; i). \tag{1}$$

It is important to distinguish between *within-replication* data and *across-replication* data in the replication method. Each sample path that is simulated consists of within-replication data. These data are often dependent. For example, the number of Class 1 jobs in queue at time 1 and at time 2 are dependent, so that $\overline{Y}_1(t; i)$ is a continuous-time average of a series of dependent observations for any fixed i. On the other hand, across-replication data consists of values from different replications. For example, if, as above, the replications are simulated independently, then $\overline{Y}_1(t; i)$ is independent of $\overline{Y}_1(t; j)$ for any

$i \neq j$, so that the (sample) average (1) is an average of independent observations.

The performance measures PM1–PM4 are all finite-horizon performance measures. One might also be interested in *infinite-horizon* performance measures. These are performance measures that depend on the distribution of a sample path that, in principle, is simulated over the entire interval $[0, \infty)$. One such performance measure is expected discounted performance.

PM5. For our example this might take the form

$$\mathrm{E}\int_0^\infty \mathrm{e}^{-\delta s} Y_1(s)\,\mathrm{d}s,$$

where $\delta > 0$ is a discount factor.

Although there is no need to specify a time horizon t for this performance measure, one must specify the discount factor δ. Estimating PM5 is not straightforward, since it depends on an infinite time horizon. Nevertheless, methods exist for estimating it.

All of the performance measures mentioned so far depend on the initial condition of the simulation. In other words, PM1–PM5 typically depend on the state and clock readings at time 0. While such performance measures are appropriate for many applications, there are some settings where they are not appropriate. For example, in some communication network applications, even a few seconds is a very long time in terms of network evolution. On such time scales, discounting may seem inappropriate, and the initial condition often plays essentially no role. In such settings, one may wish to turn to *steady-state* performance measures.

PM6. Suppose that

$$\overline{Y}_1(t) = \frac{1}{t}\int_0^t Y_1(s)\,\mathrm{d}s \to \alpha$$

as $t \to \infty$ almost surely, where α is a deterministic constant that is the same regardless of the initial condition of the simulation.

The constant α can be interpreted as the long-run average number of Class 1 jobs in the system. The *steady-state estimation problem* is the problem of estimating α. As with PM5, estimation of α poses several challenges to the simulation user. Methods have been developed to address those challenges, and are discussed in Chapters 13, 14, 15 and 16.

The term “steady-state” has been somewhat abused in this definition. The steady-state simulation problem as defined above would perhaps be better named the “time-average estimation problem”, since the problem as stated involves estimating the limiting value of a time average. However, this definition is more easily stated than a more traditional definition involving steady-state distributions of stochastic processes. Furthermore, under certain conditions, limits of time averages like α above and the expected values of steady-state distributions coincide. For our example those conditions are roughly as follows: The stations are capable of keeping up with the incoming work, the input processes are stationary in the sense that they do not change with time, and there are no peculiarities of the input processes that lead to periodic behavior. Under these conditions it is reasonable to expect that the distribution of the number of Class 1 jobs in the system $Y_1(t)$ at time t will depend less and less on t as t increases. Notice that the system state itself will continue to evolve in a random fashion, but the *distribution* of the system state will tend to settle down. Our definition of the steady-state estimation problem avoids the need to make some of these assumptions, which helps to explain our preference for it. See Wolff (1989) for more on this issue.

One of the key challenges in the estimation of any performance measure is error estimation. The term *error* refers to the difference between a deterministic quantity and an estimate of that quantity constructed through simulation. In simulation applications error estimation usually takes the form of a confidence interval for a performance measure. Consider PM1, for example, where we wish to compute the deterministic quantity $\mathrm{E}\overline{Y}_1(t)$ for some fixed t. Assuming that $\overline{Y}_1(t)$ has a finite variance, the central limit theorem applies, and ensures that the sample average (1) is approximately normally distributed for large n. This limit theorem then allows one to report a confidence interval for $\mathrm{E}\overline{Y}_1(t)$ and hence to get a sense of the error in the estimator. This form of error is often called *estimation error* to distinguish it from *modeling error* as discussed below. See Chapters 2, 8 and 9, and Part V for further details on evaluating and controlling error.

A common misconception is to confuse error with *risk*. Continuing with PM1, suppose that the decision maker is worried about excessive queue lengths in the period $[0, t]$. The summary performance measure $\mathrm{E}\overline{Y}_1(t)$ gives some idea of the magnitude one might expect of $\overline{Y}_1(t)$, and so it offers some information on the potential risk involved. A 95% confidence interval for $\mathrm{E}\overline{Y}_1(t)$ might be reported as $(Z - H, Z + H)$ or $Z \pm H$, where Z is the midpoint and H is the halfwidth. This confidence interval is sometimes misinterpreted to mean that 95% of the time the random variable $\overline{Y}_1(t)$ will lie in the interval $(Z - H, Z + H)$. To see why this must be false, notice that the confidence interval width can be made arbitrarily small by increasing the number of simulation replications. The confidence interval will still be a 95% confidence interval, but now will be of the form $Z' \pm H'$, where the interval $(Z' - H', Z' + H') \subset (Z - H, Z + H)$ with high probability. So the level of confidence 95% has everything to do with estimation error, and nothing to do with risk.

A measure of risk for this example may instead be related to the *distribution* of $\overline{Y}_1(t)$. For example, one might estimate the $\Pr(\overline{Y}_1(t) \in (y_\ell, y_u))$, the probability that the (random) average number of Class 1 jobs lies in the deterministic interval (y_ℓ, y_u). This probability can be estimated by simulation, and a confidence interval generated to give some idea of the accuracy of the estimate of the probability. For further discussion of these and related points see Chapters 9 and 14.

We are finally in a position to conduct a simulation of our model. For our specific implementation the interarrival times of Class 1 jobs are exponentially distributed with mean 110 seconds, as are the interarrival times of Class 2 jobs. The processing times at Station 1 are deterministic and equal to 100 seconds, as are the processing times at Station 2. We start the system off empty, with arrival clocks sampled from their respective distributions. We estimate $\mathrm{E}\overline{Y}_1(t)$ and $\mathrm{E}\overline{Y}_2(t)$, where $\overline{Y}_2(t)$ is the average number of Class 2 jobs in the system, up to time $t = 500$ hours. We performed 20 replications, where each replication consisted of running the model for 500 hours.

The 95% confidence interval for $\mathrm{E}\overline{Y}_1(t)$ was 0.9 ± 0.3, so that the average number of Class 1 jobs in the system was around 1! This looks like absolutely wonderful performance. Unfortunately, the 95% confidence interval for $\mathrm{E}\overline{Y}_2(t)$ was 33 ± 4, so that Class 2 jobs fared rather poorly. On a hunch we also ran the model for 20 replications of $t = 2000$ hours each. The confidence interval for $\mathrm{E}\overline{Y}_1(t)$ remained around 1, but the confidence interval for $\mathrm{E}\overline{Y}_2(t)$ was 108 ± 12. The mean number of Class 2 jobs increased by approximately four times when we ran the simulation for four times as long. This suggests that the queue of Class 2 jobs is growing as the simulation proceeds. In other words, the system is unstable.

When such behavior is observed, one typically assumes that there is not enough service capacity to keep up with incoming work. But if Class 1 jobs were never routed to Station 2, then we would have two separate single-server queues where the mean interarrival time (110) is greater than the mean service time (100). Such queues are known to be stable. So there is, indeed, enough service capacity. It is just that our priority policy at Station 2 is not a good one.

To diagnose the trouble we explore further. A look at the utilizations of the stations shows that Station 1 is utilized approximately 65% of the time, while Station 2 is utilized 100% of the time. This suggests that Station 1 is being starved of work while Station 2 is overloaded. If one watches an animation of the system (a standard feature in most graphical simulation software), one immediately sees the trouble. When a Class 1 job arrives and Station 1 is busy, the job moves to Station 2 if it can be served immediately. Otherwise it waits in Queue 1. Then, when Station 1 completes its current job, there is often no new Class 1 job to work on. So Station 1 has to wait until the next Class 1 job arrives, and so it is often idle. Meanwhile, Station 2 continues to help out with Class 1 jobs, even when the queue of Class 2 jobs gets long.

There are several ways to modify the policy to improve performance. There is a great deal of theory (see, e.g., Harrison, 1996; Bell and Williams, 1999; Meyn, 2001, 2003) that suggests that a policy that is close to the best that one can do is to instead have Station 2 work on Class 1 jobs only when there are at least x Class 1 jobs in the queue, where x is a *decision variable* that needs to be chosen. If x is large, then Station 2 will only assist Station 1 when the backlog of Class 1 jobs is large. In this case it is highly unlikely that Station 1 will be left with no jobs to process. Indeed, if we take $x = 5$ then we get confidence intervals for $\mathrm{E}\overline{Y}_1(t)$ and $\mathrm{E}\overline{Y}_2(t)$ given by 2.4 ± 0.2 and 8 ± 2, where t is again chosen to be 500 hours and we performed 20 replications. This is far more reasonable! In this setting, x is known as a safety stock. Notice that if x is too large, then we can end up with long waiting times for Class 1 jobs which is undesirable. On the other hand, if x is too small, then the system cannot keep up with the arrival of Class 2 jobs.

So an important question is how to choose the integer x. Suppose we adopt the position that we want to minimize $5\mathrm{E}\overline{Y}_1(t) + \mathrm{E}\overline{Y}_2(t)$. The multiplier 5 is chosen to reflect the importance of completing Class 1 jobs quickly. This is an example of a discrete-optimization problem where the objective function is estimated via simulation. Such problems are the subject of Chapters 20 and 21. Of course, we can only get *estimates* of the objective function, and so with a finite amount of simulation effort we can never be certain that the reported optimal value of x is indeed optimal. Chapter 17 discusses the question of how to report a solution as optimal with a given level of confidence. See Henderson et al. (2003) for more on the safety stock selection problem.

One needs to interpret confidence levels or, more generally, error estimates, with care. An error estimate is a statement about the statistical properties of the model that was implemented. As such it is a statement about *estimation error*. Recall that estimation error is the difference between the *true* value of a performance measure for the simulation model and the *estimate* of that value resulting from running the simulation model. We can typically make estimation error very small by running the simulation for a long time. It is important to be aware that a small estimation error is no guarantee that the simulation results closely mirror reality.

To see why, notice that the implemented model typically contains a number of simplifications of the underlying system that is being studied. (This underlying system could either be *extant*, i.e., currently existing, or *conceptual*, i.e., proposed but not yet existing. We do not distinguish between these two cases.) For example, in our network model we may have ignored a small variation in the interarrival distribution depending on the time of day, or there may be an operator that is required to set the processing in motion before a job can begin service that we have not modeled. Even if the model is a complete representation of reality, the estimation of parameters of distributions (e.g., the mean interarrival time of jobs) can introduce modeling error. The difference between the value of a performance measure for the *actual* system (whether extant or conceptual) and the value of the same performance measure for the corresponding simulation model is known as *modeling error*. One can simultaneously have small estimation error yet large modeling error.

The sum of these two sources of error gives the difference between true system performance and the estimate of performance based on the simulation model. One can often control estimation error by increasing simulation lengths. It is typically much more difficult to get a sense of modeling error. One way is to perform sensitivity analyses of the model, but there are also other methods. The modeling error/estimation error issue is further explored in Chapters 9 and 14.

2.2 *A stochastic activity network problem*

Consider a set of nodes $\mathcal{N}$ and arcs $\mathcal{A}$ such that $(\mathcal{N}, \mathcal{A})$ defines a directed, acyclic network with a single source node $a \in \mathcal{N}$ and sink node $z \in \mathcal{N}$. Figure 2 is an illustration. Associated with each arc $i \in \mathcal{A}$ is a random variable X_i representing the "length" or "duration" of the arc. Here the arcs represent tasks to be completed in a project, and the nodes represent milestones along the way. Let $\mathcal{P}$ be the set of all distinct paths (sequences of arcs) from a to z. In the illustration $\mathcal{P} = \{(1, 4, 6), (1, 3, 5, 6), (2, 5, 6)\}$. There are at least two problems that can be associated with the network $(\mathcal{N}, \mathcal{A})$:

Stochastic longest route problem: If all inbound tasks to each node must complete before any outbound tasks can begin, then the time to complete the project is

$$Y = \max_{P \in \mathcal{P}} |P|, \tag{2}$$

where

$$|P| = \sum_{i \in P} X_i$$

is the duration of the path P, so that Y is simply the duration of the longest path. The random variables

$$C_j = I\big(|P_j| = Y\big), \quad P_j \in \mathcal{P}, \tag{3}$$

indicate which path (or paths) are the longest (most critical). Here, $I(\cdot)$ is the indicator function that is 1 if its argument is true and 0 otherwise.

Stochastic shortest route problem: If the paths $\mathcal{P}$ represent a collection of alternative approaches to the project, then the shortest route is of interest. When all approaches will be attempted simultaneously,

$$Y = \min_{P \in \mathcal{P}} |P|$$

is the time to complete the project. If only a single alternative will be

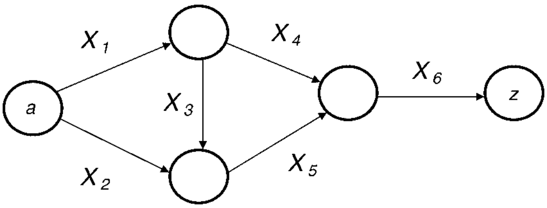


Fig. 2. Stochastic activity network.

chosen, then

$$C_j = I\big(|P_j| = Y\big)$$

is an indicator of whether path P_j is the shortest (optimal) path.

In this section we consider the stochastic longest route problem and focus on the completion time Y. The random variable Y is easier to simulate than the dynamic stochastic process of Section 2.1, since the following algorithm will suffice:

(1) Generate a random sample X_i for all $i \in \mathcal{A}$.
(2) Set $Y = \max_{P \in \mathcal{P}} |P|$.

Notice that the concept of time does not play a central role here as it did in Section 2.1; in fact, the simulation is merely an algorithm for realizing the random variable Y. Simulations of this type are sometimes called *Monte Carlo simulations*, although the term is also used for any simulation that involves random sampling. Placing this problem within the framework of Section 2.1, the stochastic longest route problem is a terminating simulation, and Y represents finite-horizon performance. Therefore, the performance measures PM2–PM4 are still relevant:

PM2. $\mathrm{E}Y$ is the expected value of the time to complete the project.
PM3. $\Pr\{Y > t\}$ is the probability that the project duration exceeds some threshold.
PM4. For given probability p, $y_p = \inf\{y\colon \Pr(Y \leqslant y) \geqslant p\}$ is the pth quantile of project completion, a completion date that can be achieved with probability p.

Let $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ be the random vector of all task durations, with joint cumulative distribution function $F_{\mathbf{X}}$. Then we can write

$$F_Y(t) \equiv \Pr\{Y \leqslant t\} = \int_{\mathbf{x}:|P| \leqslant t, P \in \mathcal{P}} \mathrm{d}F_{\mathbf{X}}(\mathbf{x}). \tag{4}$$

If we could compute F_Y then, in theory, we could also compute $\mathrm{E}Y$ and y_p. However, even in simple networks this can be a difficult calculation, and it is essentially impossible if the network is large and the joint distribution of $\mathbf{X}$ is complex. As a result, simulation becomes competitive because PM2–PM4 can all be estimated by the replication method. For instance, if we repeat the simulation algorithm n times to produce i.i.d. samples of project completion time $Y_1, Y_2, \ldots, Y_n$, then the natural estimator for $\mathrm{E}Y$ is

$$\frac{1}{n}\sum_{i=1}^{n} Y_i, \tag{5}$$

the natural estimator of $\Pr\{Y > t\}$ is

$$\frac{1}{n}\sum_{i=1}^{n} I(Y_i > t) \tag{6}$$

and the natural estimator of y_p is

$$Y_{(\lceil np \rceil)}, \tag{7}$$

where $Y_{(1)} \leqslant Y_{(2)} \leqslant \cdots \leqslant Y_{(n)}$ are the order statistics (sorted values) of the sample.

If the network is large and has many paths, or t and p are extreme, then the estimation error associated with (5)–(7) may be unacceptably large unless n is very big. Part IV describes techniques for reducing the estimation error of the natural estimators without a corresponding increase in computational effort (larger n) or analyst effort (since every minute spent figuring out a more efficient estimator could be spent simulating).

Consider $\mathrm{E}Y$. Because (5) is an unbiased estimator, its mean squared error is

$$\begin{aligned} \mathrm{MSE}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) &= \mathrm{Bias}^2\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) + \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) \\ &= \frac{\mathrm{Var}(Y)}{n}. \end{aligned} \tag{8}$$

Reducing this MSE is the goal of Chapters 10–12, and it is important to consider MSE because some efficiency improvement techniques reduce variance at the cost of introducing some bias. Efficiency improvement often requires that the analyst exploit knowledge beyond the minimum required to build the simulation. For instance, in the network of Figure 2 there is no need to simulate X_6 if the goal is to estimate $\mathrm{E}Y$: The $\mathrm{E}X_6$ can simply be added to the time to reach the penultimate node and a source of variance in the estimator disappears. Rarely is efficiency improvement this easy. Notice, however, that the expected duration of each path $P_j \in \mathcal{P}$ is computable since

$$\mathrm{E}\left(\sum_{i \in P_j} X_i\right) = \sum_{i \in P_j} \mathrm{E}X_i.$$

The actual durations of the paths with the longest expected durations will be correlated – perhaps strongly correlated if there are a few dominant paths – with the project duration Y. A more general efficiency improvement method, called control variates (see Chapter 10), can exploit the correlation between a random variable whose expected value is known and a random variable whose expected value we want to estimate to reduce estimator variance.

In the classical stochastic longest route problem $X_1, X_2, \ldots, X_k$ are independent, but this certainly need not be the case in practice. If the duration of several tasks are affected by weather (as might occur in a construction project), then particularly bad or good weather might influence several tasks, making their durations dependent. One reason that independence is often assumed is that characterizing and simulating a joint distribution, especially when the marginal distributions are from distinct families, is difficult. For instance, specifying the marginal distributions and correlations among $X_1, X_2, \ldots, X_k$ does not, in general, uniquely define their joint distribution. Chapter 5 takes up this problem. Notice that dependence across the activity durations X_i does not interfere with the application of control variates described above, since only the expected path duration is required. On the other hand, the joint distribution could be critical to the definition of a path-based control variate for estimating $\Pr\{Y > t\}$.

Up to this point we have treated the network $(\mathcal{N}, \mathcal{A})$ as fixed. But it may be that, by spending some money, the duration of the project can be shortened. As a simple example, suppose that all of the task durations are exponentially distributed with nominal means $\mu_1, \mu_2, \ldots, \mu_k$, respectively, and that μ_i can be reduced to m_i at a cost of $\$c_i$ per unit reduction. If a total budget of $\$b$ is available, how should it be spent? We now think of the project duration as a random function of the mean durations m_i, say $Y(m_1, m_2, \ldots, m_k)$. (It is random because we have only specified the means of the task durations. The actual task durations are still random.) One way to formulate the decision problem is

$$\begin{aligned}
&\min \mathrm{E}\bigl[Y(m_1, m_2, \ldots, m_k)\bigr] \\
&\text{subject to:} \\
&\quad \sum_{i=1}^{k} c_i(\mu_i - m_i) \leqslant b \\
&\quad 0 \leqslant m_i \leqslant \mu_i, i = 1, 2, \ldots, k.
\end{aligned} \tag{9}$$

This is an example of optimization via simulation, covered in Part VI, and specifically continuous-decision-variable optimization. Chapter 18 describes methods for selecting and fitting a tractable metamodel that represents, say, $\mathrm{E}[Y(m_1, m_2, \ldots, m_k)]$ as a function of the decision variables. Chapter 19 reviews techniques for estimating gradient search directions for continuous-decision-variable optimization. In both approaches the need to handle high-dimensional (large k) problems and sampling variability is central.

3 Organization of the Handbook

The remainder of this volume is organized into parts that contain several chapters each. Part I, including this chapter and the next, provides fundamental background that will be helpful when reading the more specialized chapters that follow. Part II covers random-number generation, random-variate generation, and issues related to the implementation of each. The two chapters in Part III lay the foundation for evaluating system performance via simulation from both a frequentist and Bayesian statistical perspective. Using the material in Part III as the foundation, Part IV describes methods for improving the computational and statistical efficiency of performance estimation, while Part V addresses the need to evaluate and control whatever estimation error remains. Finally, Part VI tackles the goal of optimizing the performance of a simulated system with respect to controllable decision variables.

Acknowledgements

The work of the first author was supported by National Science Foundation Grants DMI-0224884, DMI-0230528 and DMI-0400287. The work of the second author was supported by National Science Foundation Grants DMI-0217690 and DMI-0140385.

References

Banks, J. (Ed.) (1998). *Handbook of Simulation*. Wiley, New York.

Banks, J., Carson II, J.S., Nelson, B.L., Nicol, D.M. (2004). *Discrete-Event System Simulation*, 4th edition. Prentice Hall, Upper Saddle River, NJ.

Bell, S., Williams, R. (1999). Dynamic scheduling of a system with two parallel servers: Asymptotic optimality of a continuous review threshold policy in heavy traffic. In: *Proceedings of the 38th Conference on Decision and Control*. Phoenix, AZ, pp. 1743–1748.

Fishman, G.S. (2001). *Discrete-Event Simulation: Modeling, Programming and Analysis*. *Springer Series in Operations Research*. Springer-Verlag, New York.

Fishwick, P.A. (1995). *Simulation Model Design and Execution: Building Digital Worlds*. Prentice Hall, Englewood Cliffs, NJ.

Fujimoto, R.M. (1999). *Parallel and Distributed Simulation Systems*. Wiley, New York.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glynn, P.W. (1989). A GSMP formalism for discrete event systems. *Proceedings of the IEEE* 77, 14–23.

Haas, P.J. (1999). On simulation output analysis for generalized semi-Markov processes. *Communications in Statistics: Stochastic Models* 15, 53–80.

Harrison, J.M. (1996). The BIGSTEP approach to flow management in stochastic processing networks. In: Kelly, F.P., Zachary, S., Ziedins, I. (Eds.), *Stochastic Networks Theory and Applications*. Clarendon Press, Oxford, UK, pp. 57–89.

Henderson, S.G., Meyn, S.P., Tadić, V. (2003). Performance evaluation and policy selection in multiclass networks. *Discrete Event Dynamic Systems* 13, 149–189. Special issue on learning and optimization methods.

Jain, R.K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Meyn, S.P. (2001). Sequencing and routing in multiclass queueing networks. Part I: Feedback regulation. *SIAM Journal on Control and Optimization* 40 (3), 741–776.

Meyn, S.P. (2003). Sequencing and routing in multiclass queueing networks. Part II: Workload relaxations. *SIAM Journal on Control and Optimization* 42 (1), 178–217. Presented at the 2000 IEEE International Symposium on Information Theory, Sorrento, Italy, June 25–June 30.

Nelson, B.L. (2002). *Stochastic Modeling: Analysis & Simulation*. Dover Publications, Mineola, NY.

Shedler, G.S. (1993). *Regenerative Stochastic Simulation*. Academic Press, Boston, MA.

Sterman, J.D. (2000). *Business Dynamics*. McGraw-Hill, New York.

Wolff, R.W. (1989). *Stochastic Modeling and the Theory of Queues*. Prentice Hall, Englewood Cliffs, NJ.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13002-9

Chapter 2

Mathematics for Simulation

Shane G. Henderson

School of Operations Research and Industrial Engineering, Cornell University, USA
E-mail: sgh9@cornell.edu

Abstract

This chapter surveys certain mathematical results and techniques that are pervasive in the analysis of stochastic simulation. The concepts are introduced through the study of a simple model of ambulance operation to ensure clarity, concreteness and cohesion.

1 Introduction

Stochastic simulation (henceforth just "simulation") is a tremendously broad subject that draws from diverse mathematical fields including, but certainly not limited to, applied probability, statistics, number theory, and mathematical programming. One of the goals of this handbook is to survey stochastic simulation, communicating the key concepts, techniques and results that serve as a foundation for the field. This chapter contributes to that goal by surveying a collection of mathematical techniques and results that pervade simulation analysis.

Given the breadth of the field of simulation, it is necessary to focus the discussion somewhat. This chapter describes a set of mathematical tools and techniques that can be used to explore the estimation of performance measures in both the terminating and steady-state simulation context. The focus throughout is on *large-sample properties* of estimators, i.e., those properties of estimators that persist when simulation runlengths become large. Any practical simulation study works with a finite simulation runlength and so may not reach the regime where large-sample properties emerge. It is therefore of great practical importance to consider the *small-sample properties* of estimators as well. We do not do so here, since to do so would enlarge the scope beyond what is manageable in a single chapter.

A refined statement of the goal of this chapter is then to survey a subset of mathematical techniques and results that are useful in understanding the large-sample behavior of estimators of performance measures. It would be very easy to provide a smorgasbord of such results, but such a chapter would read like a dictionary. Therefore, many of the results are applied to a simple model of ambulance operation that serves to unify the discussion and hopefully make it more interesting.

This chapter is an outgrowth of Henderson (2000) and Henderson (2001), in which there were 2 main topics. First, in the terminating simulation context, performance measures were rigorously defined through the strong law of large numbers for i.i.d. random variables. The performance of estimators of these performance measures was studied via the central limit theorem. Variants of these results were used to study performance measures that, instead of being expectations of random variables, were *functions* of expectations of random variables. Second, in the steady-state context, performance measures were rigorously defined and analyzed by appealing to asymptotic results for general state-space Markov chains. Lyapunov conditions were used to provide sufficient conditions under which the asymptotic results hold.

All of the performance measures described in Henderson (2000) take the form of an expectation of a random variable, or a differentiable function of a finite number of expectations. Such performance measures are particularly useful when the goal is to compare many different stochastic systems, as they provide a concrete basis for the comparison. If instead the goal is to enhance one's *understanding* of a single stochastic system, then it is often more useful to analyze the *distribution* of certain random variables, perhaps through density estimation techniques. This was the focus of Henderson (2001). This chapter combines elements of both of those papers, with a leaning toward density estimation.

In Section 2 we review some approaches to performance-measure estimation in a particularly transparent context, namely that of estimating the density of the completion time in a stochastic activity network. The analysis in this section requires the use of the strong law of large numbers (SLLN) and central limit theorem (CLT). We also review the continuous mapping theorem and converging together lemma.

Section 3 sets the stage for the remainder of the chapter by introducing a simple model of ambulance operation. In Section 4 we specialize this model to the terminating simulation context. Even the *definition* of certain performance measures leads to the use of some interesting techniques and results.

In Section 5 we modify the ambulance model slightly to obtain a steady-state simulation. To rigorously define performance measures for this model, it is necessary to define an appropriate stochastic process with which to work. A great deal is known about the class of Markov processes evolving on general (not necessarily countable) state spaces, and so a general state space Markov chain is defined. To ensure that long-run averages exist, it is necessary to show that this chain is, in a certain sense, positive recurrent.

A very practical approach to establishing that a Markov chain is positive recurrent is to use Lyapunov functions, and this approach is the central mathematical tool illustrated in Section 5. We use Lyapunov theory to show that certain Markov chains are positive recurrent, that our performance measures are well defined, and that certain estimators are consistent and satisfy central limit theorems. An important consideration in the steady-state context is that of initialization bias. We also use Lyapunov theory to characterize this bias. Sometimes a natural choice of Lyapunov function does not work, at least at first sight. Section 5 concludes with a discussion of one approach to dealing with such problems that is especially applicable to queueing examples.

The underlying theme of Section 5 is that Lyapunov functions provide an enormously powerful and easily applied (at least relative to many other methods) approach to establishing results that underlie steady-state simulation methodology. It is fair to say that Lyapunov functions have not been broadly applied in simulation analysis. It is far more common, for example, to see results based on mixing hypotheses for stationary processes. But such hypotheses are difficult to verify in practice, which helps to explain the attention devoted to Lyapunov techniques in this chapter.

Throughout the chapter results are rigorously quoted and references given for the proofs. To simplify the exposition it is often the case that results are quoted using stronger hypotheses than are strictly necessary, but tighter hypotheses can be found in the references provided.

Notation is occasionally reused from section to section, but is consistently applied within each section.

2 Static simulation: Activity networks

Our running example in this section will be that of a stochastic activity network (SAN), as introduced in Chapter 1. A SAN is a directed graph that represents some set of activities that, taken together, can represent some project/undertaking. Each arc in the graph represents a task that needs to be completed, and the (random) length of an arc represents the time required to complete the associated task. Nodes are used to indicate the precedence relationships between tasks. The time required to complete the project is indicated by the longest path between the designated “source” and “sink” nodes.

Example 1. The stochastic activity network in Figure 1 is adapted from Avramidis and Wilson (1996). Nodes a and i are the source and sink nodes respectively. The arcs are labeled for easy identification. For simplicity we assume that the task durations associated with each arc are independent of one another.

Let Y be the (random) network completion time, i.e., the longest path from node a to node i. We are interested in computing both $\mathrm{E}Y$ (assuming it is finite) and the distribution of Y. We first consider $\mathrm{E}Y$.

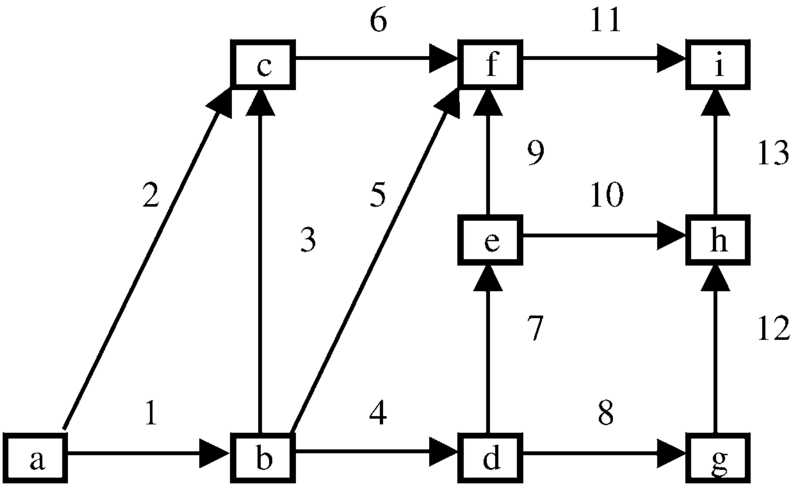


Fig. 1. A stochastic activity network.

As noted in Chapter 1, it is easy to estimate $\mathrm{E}Y$. One simply generates i.i.d. replicates $Y_1, Y_2, \ldots, Y_n$ of Y, and then forms the sample average

$$\overline{Y}_n = \frac{1}{n}\sum_{j=1}^{n} Y_j.$$

The strong law of large numbers ensures that the estimator $\overline{Y}_n$ is a strongly consistent estimator of $\mathrm{E}Y$. (An estimator is *strongly consistent* if it converges almost surely to the appropriate value. It is *consistent* if it converges in probability to the appropriate value. In simulation we don't usually concern ourselves with the distinction between these two concepts since it is impossible to distinguish between the types of convergence based on a finite runlength. However, the difference is important in establishing the validity of sequential stopping (Glynn and Whitt, 1992), and there may be other contexts where the difference plays a role.)

Theorem 1 (SLLN). *If $X_1, X_2, \ldots$ is an i.i.d. sequence of random variables with $\mathrm{E}|X_1| < \infty$, then*

$$\frac{\sum_{i=1}^{n} X_i}{n} \to \mathrm{E}X_1 \quad a.s.$$

as $n \to \infty$.

For a proof, see Billingsley (1986, p. 290).

To apply this result we need to ensure that $\mathrm{E}Y_1 < \infty$. Let T_i be a random variable giving the completion time for task i. Let $\mathcal{A}$ be the set of all arcs. Then

$$Y \leqslant \sum_{i \in \mathcal{A}} T_i, \tag{1}$$

so that a simple sufficient condition for the strong law to hold is that all task durations have finite mean.

Under this condition we know that the estimator $\overline{Y}_n$ converges almost surely to $\mathrm{E}Y$ as $n \to \infty$. But how accurate is it for a finite value of n? The central limit theorem (CLT) provides an answer to this question.

Theorem 2 (CLT). *If $X_1, X_2, \ldots$ is an i.i.d. sequence of random variables with* $\mathrm{E}X_1^2 < \infty$, *then*

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - \mathrm{E}X_1\right) \Rightarrow \sigma\mathrm{N}(0,1)$$

as $n \to \infty$, where $\sigma^2 = \operatorname{var} X_1$, "$\Rightarrow$" denotes convergence in distribution and $\mathrm{N}(0,1)$ *denotes a normal random variable with mean* 0 *and variance* 1.

For a proof, see Billingsley (1986, p. 367).

To apply the CLT we need $\mathrm{E}Y_1^2 < \infty$. From (1) this follows if $\mathrm{E}T_i^2 < \infty$ for all i. If $\mathrm{E}Y_1^2 < \infty$, then the CLT basically establishes that the error in the estimator $\overline{Y}_n$ is asymptotically normally distributed with mean 0 and variance s^2/n, where $s^2 = \operatorname{var} Y_1$, and this is the basis for obtaining confidence intervals for $\mathrm{E}Y$. In particular, an approximate 95% confidence interval for $\mathrm{E}Y_1$ is given by

$$\overline{Y}_n \pm 1.96\frac{s}{\sqrt{n}}. \tag{2}$$

Of course, s^2 must invariably be estimated. The usual estimator is the sample variance

$$s_n^2 = \frac{1}{n-1}\sum_{j=1}^{n}(Y_j - \overline{Y}_n)^2.$$

The confidence interval that is reported is the same as (2) with s replaced by its sample counterpart s_n. But is the modified confidence interval then valid?

If $\mathrm{E}Y_1^2 < \infty$, then the SLLN implies that $s_n^2 \to s^2$ as $n \to \infty$ a.s. Hence, from Billingsley (1986, Exercise 29.4) we have that

$$\begin{pmatrix} n^{1/2}(\overline{Y}_n - \mathrm{E}Y) \\ s_n^2 \end{pmatrix} \Longrightarrow \begin{pmatrix} s\,\mathrm{N}(0,1) \\ s^2 \end{pmatrix}. \tag{3}$$

The joint convergence in (3) is a direct result of the fact that s^2 is a deterministic constant. In general, marginal convergence does not imply joint convergence.

The natural tool to apply at this point is the continuous mapping theorem. For a function $h : \mathbb{R}^d \to \mathbb{R}$, let D_h denote its set of discontinuities (in $\mathbb{R}^d$).

Theorem 3 (Continuous Mapping Theorem). *Let $(X_n \colon n \geqslant 1)$ be a sequence of $\mathbb{R}^d$-valued random variables with $X_n \Rightarrow X$ as $n \to \infty$ and let $h : \mathbb{R}^d \to \mathbb{R}$. If $\mathrm{P}(X \in D_h) = 0$, then $h(X_n) \Rightarrow h(X)$ as $n \to \infty$.*

For a proof, see Billingsley (1986, p. 391).

Define $h(x, y) = x/y^{1/2}$, and then apply the continuous mapping theorem to (3), to obtain

$$\frac{n^{1/2}(\overline{Y}_n - \mathrm{E}Y_1)}{s_n} \Rightarrow \mathrm{N}(0, 1) \tag{4}$$

as $n \to \infty$ when $s^2 > 0$, and so the confidence interval procedure outlined above is indeed valid.

The argument leading to (4) is commonly known as a *converging together argument*. It is based on the *converging together lemma*, sometimes known as *Slutsky's theorem*, which is a direct corollary of the continuous mapping theorem. We have essentially supplied a proof of this result in the argument above.

Corollary 4 (Converging Together Lemma). *If $X_n \Rightarrow X$, $U_n \Rightarrow u$ and $V_n \Rightarrow v$ as $n \to \infty$, where $X \in \mathbb{R}^d$, and u and v are constants, then*

$$U_n X_n + V_n \quad \Longrightarrow \quad uX + v$$

as $n \to \infty$.

Let us now see how these same tools can be used to estimate another performance measure. Recall that we are interested in the distribution of Y beyond its mean. One way to represent the distribution is through a density function. So when does Y have a density?

Before answering this question, we first need to understand what we mean when we say that a random variable has a density. The standard definition is based on the Radon–Nikodym theorem; see Billingsley (1986, pp. 434, 443).

Definition 1. We say that a real-valued random variable X has a density if $\mathrm{P}(X \in A) = 0$ for all Lebesgue-measurable sets A with Lebesgue measure 0. This is equivalent to saying that there exists a nonnegative function f (a density function) with the property that for all $x \in \mathbb{R}$,

$$F(x) \equiv \mathrm{P}(X \leqslant x) = \int_{-\infty}^{x} f(y)\,\mathrm{d}y.$$

The first part of this definition may not be as familiar as the second part. Heuristically speaking, X has a density if the probability that X takes on values in "insignificant" sets is 0. The second part of the definition is perhaps more familiar. We will use the 2 definitions interchangeably in what follows. The proof of the following result demonstrates the use of the definitions.

Proposition 5. *Consider a SAN with a finite number of arcs/tasks, where the individual task durations are independent. Suppose that every path from the source to the sink contains an arc for which the corresponding task duration has a density. Then the time to complete the project has a density.*

Proof. Let P be a path from the source to the sink. The length L of the path P is the sum of the times required to traverse each arc in the path. At least one of these times has a density, and since the task durations are independent, it follows that L has a density. Let $m < \infty$ denote the number of such paths from the source to the sink, and let $L_1, \dots, L_m$ denote the times required to traverse each path. Then $Y = \max\{L_1, \dots, L_m\}$.

Now, the maximum of 2 random variables, X_1 and X_2 say, that have densities, also has a density. To see why, let A denote an arbitrary (measurable) subset of the real line, and let $Z = \max\{X_1, X_2\}$. Then

$$\begin{aligned} \mathrm{P}(Z \in A) &\leqslant \mathrm{P}\big(\{X_1 \in A\} \cup \{X_2 \in A\}\big) \\ &\leqslant \mathrm{P}(X_1 \in A) + \mathrm{P}(X_2 \in A), \end{aligned}$$

where the first inequality follows since $Z \in A$ implies that at least one of X_1 and X_2 must be in A, and the second is Boole's inequality. Now, we know that X_1 and X_2 have densities, so if the Lebesgue measure of A is 0, then

$$\mathrm{P}(X_1 \in A) = \mathrm{P}(X_2 \in A) = 0.$$

Hence $\mathrm{P}(Z \in A) = 0$ and so, by Definition 1, Z has a density.

We can now apply this result inductively to $Y = \max\{L_1, \dots, L_m\}$ to conclude that Y has a density. □

Applying this result to Example 1, we see that the network completion time Y will have a density if T_{11} and T_{13}, the completion times for tasks 11 and 13, have densities. But then, how can we estimate this density?

In general, density estimation is difficult. However, the special structure in this problem allows us to use a simple device. The look-ahead density estimators developed by Henderson and Glynn (2001) are easily analyzed, and have excellent statistical properties. The mathematics of look-ahead density estimation are intimately related to those of gradient estimation via conditional Monte Carlo. See Chapter 19 for more on conditional Monte Carlo and gradient estimation, and Fu and Hu (1997) for a comprehensive account.

Let L_f and L_h be the lengths of the longest paths from the source node to nodes f and h respectively. Recall that T_{11}, T_{13} denote the (random) task durations for tasks 11 and 13. Let F_{11}, F_{13} be the corresponding distribution functions which we assume to be continuously differentiable, and let f_{11}, f_{13} be the corresponding derivatives (and therefore densities). Then

$$\begin{aligned}
&\mathrm{P}(Y \leqslant t) \\
&\quad = \mathrm{E}\,\mathrm{P}(Y \leqslant t | L_f, L_h) \\
&\quad = \mathrm{E}\,\mathrm{P}(T_{11} \leqslant t - L_f, T_{13} \leqslant t - L_h | L_f, L_h) \\
&\quad = \mathrm{E}\bigl[\mathrm{P}(T_{11} \leqslant t - L_f | L_f, L_h)\,\mathrm{P}(T_{13} \leqslant t - L_h | L_f, L_h)\bigr] \qquad (5) \\
&\quad = \mathrm{E}\bigl[F_{11}(t - L_f) F_{13}(t - L_h)\bigr] \\
&\quad = \mathrm{E}\int_{-\infty}^{t} \frac{\mathrm{d}}{\mathrm{d}x}\bigl\{F_{11}(x - L_f) F_{13}(x - L_h)\bigr\}\,\mathrm{d}x \qquad (6) \\
&\quad = \mathrm{E}\int_{-\infty}^{t} \bigl\{F_{11}(x - L_f) f_{13}(x - L_h) + f_{11}(x - L_f) F_{13}(x - L_h)\bigr\}\,\mathrm{d}x \\
&\quad = \int_{-\infty}^{t} \mathrm{E}\bigl\{F_{11}(x - L_f) f_{13}(x - L_h) + f_{11}(x - L_f) F_{13}(x - L_h)\bigr\}\,\mathrm{d}x. \qquad (7)
\end{aligned}$$

Equality (5) follows since all of the task durations are independent, (6) is just the fundamental theorem of calculus and (7) follows since the integrand is nonnegative.

Thus, we can conclude that Y has a density f say, where

$$f(x) = \mathrm{E}\bigl[F_{11}(x - L_f) f_{13}(x - L_h) + f_{11}(x - L_f) F_{13}(x - L_h)\bigr]. \qquad (8)$$

(See Avramidis and Wilson, 1996, Section 4.1, for a related discussion.)

The expression (8) has an intuitive interpretation. The first term in (8) is related to the probability that the longest path from the source to the sink through node f has length at most x and at the same time, the longest path from the source to the sink through node h is exactly of length x. The second term can be interpreted similarly.

The expression (8) immediately suggests a density estimator for f. We generate i.i.d. replicates $L_f(i)$, $L_h(i)$ for $i = 1, \ldots, n$, and estimate $f(x)$ by

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n} \Gamma(i; x),$$

where

$$\begin{aligned}
\Gamma(i; x) = {} & F_{11}\bigl(x - L_f(i)\bigr) f_{13}\bigl(x - L_h(i)\bigr) \\
& + f_{11}\bigl(x - L_f(i)\bigr) F_{13}\bigl(x - L_h(i)\bigr).
\end{aligned}$$

Applying the SLLN, we see that $f_n(x) \to f(x)$ a.s., for all x for which (8) is finite. The set of values x for which this does not hold has Lebesgue measure 0, and so the convergence of f_n to f occurs for almost all x. This is all that can be expected, because densities are only defined up to a set of Lebesgue measure 0.

We can apply the CLT to report confidence intervals for $f(x)$ exactly as in the case of estimating $\mathrm{E}Y$ if $\mathrm{E}[\Gamma^2(1; x)] < \infty$.

To this point we have established pointwise consistency of the estimator f_n of f. One typically estimates the density to get a sense of what the *entire* density looks like, and not just the density at a point. To this end, we might attempt to establish *global* measures of convergence such as uniform (in x) convergence. To do so here would carry us too far afield. See Henderson and Glynn (2001) for such results for look-ahead density estimation in the Markov chain context.

The look-ahead density estimator described above has very appealing and easily-derived asymptotic properties. These attractive properties are a result of carefully investigating the model to identify exploitable properties. One might ask if this somewhat specialized density estimation technique can be complemented by a more general-purpose approach that does not require as much tailoring to specific applications. The field of *nonparametric functional estimation* encompasses several such approaches; see, e.g., Prakasa Rao (1983). Wand and Jones (1995) is a very readable introduction to the field of *kernel density estimation*, which is also discussed in Chapter 8. We will not go into this area in any detail because the mathematical techniques used to analyze kernel density estimators are beyond the scope of this chapter.

This section has introduced several mathematical techniques that are repeatedly used in simulation analysis. The SLLN and CLT need no introduction. The continuous mapping theorem and converging together lemma are not as well known, but are also ubiquitous in simulation analysis. Conditional Monte Carlo can also be viewed as a variance reduction technique; see Chapter 10.

3 A model of ambulance operations

We now describe a very simple model that will serve as a vehicle for the concepts to follow. The purpose of the example is simplicity, and certainly not realism, although with a few straightforward extensions, the model could be considered to be quite practical.

Suppose that a single ambulance serves calls in a square region. By translating and rescaling units, we may assume that the square is centered at the origin, with lower left-hand corner at $(-1/2, -1/2)$ and upper right-hand corner at $(1/2, 1/2)$. The combined hospital/ambulance base is located at the origin.

Calls arrive (in time) according to a homogeneous Poisson process with rate λ calls per hour. The location of a call is independent of the arrival process, and uniformly distributed over the square. To serve a call, the ambulance travels at unit speed in a Manhattan fashion (i.e., at any given time, movement is restricted to lie only in the x direction or the y direction) from its present location to the location of the call. For definiteness we assume that travel in the y direction is completed before travel in the x direction. A random amount of time, independent of all else, is then spent at the scene treating the patient and successive scene times are i.i.d. For definiteness we assume that scene times are gamma distributed (see Law and Kelton, 2000, p. 301, for details on this distribution). After the scene time is complete, and independent of all else, with probability p the ambulance is required to transport and admit the patient to the hospital. Hospital admission occurs instantaneously once the ambulance reaches the hospital. If the patient does not require transport to the hospital then the ambulance is immediately freed for other work. It then returns to the hospital/base. If a call requires service before the free ambulance reaches the base, then the ambulance responds to the call from its current location.

4 Finite-horizon performance

In this section, we assume that the ambulance only receives calls from (say) 7 a.m. until 11 p.m. each day. At 11 p.m., the ambulance completes the call that it is currently serving (if any) and returns to base. We will further assume that if the ambulance is engaged with a call when another call is received, then some outside agency, such as another emergency service, handles the other call. Finally, we assume that the random variables associated with each day are independent of those for all other days.

We will be primarily concerned with two performance measures.

- α The long-run fraction of calls attended by the ambulance.
- r The conditional density of the response time to a call given that the ambulance attends the call.

The utilization, or fraction of time that the ambulance is busy, is also of interest but the performance measures α and r are sufficient for our purposes.

We first consider α, the long-run fraction of calls attended by the ambulance. Let N_i denote the total number of calls received on day i, and for $j = 1, \ldots, N_i$, let A_{ij} be 1 if the ambulance is available when the jth call arrives on day i and 0 otherwise. Then the number of calls A_i attended by the ambulance on day i is $\sum_{j=1}^{N_i} A_{ij}$. After n days, the fraction of calls attended by the ambulance is given by

$$\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n N_i}. \tag{9}$$

Dividing both the numerator and denominator of (9) by n, and applying the SLLN separately to both the numerator and denominator, we see that

$$\frac{\sum_{i=1}^n A_i}{\sum_{i=1}^n N_i} \to \alpha = \frac{\mathrm{E}A_1}{\mathrm{E}N_1} \quad \text{a.s.}$$

as $n \to \infty$. But $\mathrm{E}N_1 = 16\lambda$, so we can estimate α by

$$\alpha_n = \frac{1}{n} \sum_{i=1}^{n} \frac{A_i}{16\lambda}.$$

The SLLN establishes that α_n is a consistent estimator of α, and the CLT allows us to construct confidence intervals for α based on α_n.

Notice that α is defined as $\mathrm{E}A_1/\mathrm{E}N_1$ and not as $\mathrm{E}(A_1/N_1)$. The latter quantity is not really defined, since $\mathrm{P}(N_1 = 0) > 0$. Even if we were to define A_1/N_1 to be, say, 1 on the event $N_1 = 0$, the latter quantity is not our desired performance measure. Observe that A_i/N_i ($\mathrm{E}(A_1/N_1)$) gives the actual (expected) fraction of calls on day i that are attended by the ambulance. This expected fraction weights days equally, irrespective of the number of calls received. In contrast, the quantity $\mathrm{E}A_1/\mathrm{E}N_1$ weights days by the number of calls that are received on the day, and should be preferred.

We now develop a look-ahead density estimator for r, the conditional density of the response time given that the ambulance responds to a call. But first let us understand exactly what the density r represents. Let R_{ij} be the response time for the jth call on day i when the ambulance responds to the call ($A_{ij} = 1$) and let $R_{ij} = -1$ if the ambulance does not respond to the jth call ($A_{ij} = 0$). For $t > 0$ define

$$\begin{aligned} R(t) &= \lim_{n\to\infty} \frac{\sum_{i=1}^{n} \sum_{j=1}^{N_i} A_{ij} I(R_{ij} \leqslant t)}{\sum_{i=1}^{n} A_i} \\ &= \frac{\mathrm{E} \sum_{j=1}^{N_1} A_{1j} I(R_{1j} \leqslant t)}{\mathrm{E}A_1} \end{aligned} \tag{10}$$

to be the long-run fraction of calls answered by the ambulance with response time less than or equal to t. The maximum distance the ambulance can drive is 2 units from one corner of the square to the opposite corner, so all response times are bounded by 2, and consequently $R(t)$ is 1 for all $t \geqslant 2$. We define $r(\cdot)$ to be the derivative of $R(\cdot)$ on $(0, 2)$. Of course, we have not yet established that $R(\cdot)$ is differentiable. In the process of establishing this fact we will also arrive at a look-ahead estimator for r. In essence we are again applying conditional Monte Carlo which, in this setting, is called filtering (Glasserman, 1993).

Consider the numerator $R(t)\,\mathrm{E}A_1$ of (10). We can write

$$\begin{aligned} R(t)\,\mathrm{E}A_1 &= \mathrm{E} \sum_{j=1}^{\infty} I(j \leqslant N_1) A_{1j} I(R_{1j} \leqslant t) \\ &= \sum_{j=1}^{\infty} \mathrm{E}\big[I(j \leqslant N_1) A_{1j} I(R_{1j} \leqslant t)\big], \end{aligned} \tag{11}$$

where (11) follows since the summands are nonnegative. For $i \geqslant 1$ and $j = 1, \ldots, N_i$, let B_{ij} (C_{ij}) denote the vector location of the ambulance (new call) at the time at which the jth call on day i is received. Let $d(b, c)$ denote the time required for the ambulance to travel from location b to location c. Also, let

$$g(t, b) = \mathrm{P}\bigl(d(b, C) \leqslant t\bigr) \tag{12}$$

be the probability that the travel time for the ambulance from location b to the random call location C is less than or equal to t. Observe that on the event $A_{ij} = 1$ (the ambulance answers the jth call on day i),

$$\mathrm{P}(R_{ij} \leqslant t | A_{ij}, B_{ij}) = g(t, B_{ij}).$$

So we see that

$$\begin{aligned} &\mathrm{E}\bigl[I(j \leqslant N_1) A_{1j} I(R_{1j} \leqslant t)\bigr] \\ &\quad = \mathrm{E}\bigl\{\mathrm{E}\bigl[I(j \leqslant N_1) A_{1j} I(R_{1j} \leqslant t) | A_{1j}, B_{1j}\bigr]\bigr\} \\ &\quad = \mathrm{E}\bigl\{I(j \leqslant N_1) A_{1j} g(t, B_{1j})\bigr\}. \end{aligned} \tag{13}$$

The final step (13) requires some care, but we omit the details.

Combining (13) with (11) we find that

$$R(t) = \frac{\mathrm{E} \sum_{j=1}^{N_1} A_{1j} g(t, B_{1j})}{\mathrm{E} A_1}. \tag{14}$$

We now wish to differentiate both sides of (14). For each fixed b, $g(\cdot, b)$ is continuously differentiable in t with bounded derivative. To see why, notice that $g(t, b)$ is the area of the intersection of the unit square with a diamond (recall that the ambulance travels in Manhattan fashion) centered at b with "radius" t; see Figure 2. For each fixed b this is a piecewise quadratic with continuous derivative at the breakpoints. So define

$$f(t, b) = \frac{\partial g(t, b)}{\partial t}. \tag{15}$$

For each fixed b, $f(t, b)$ is piecewise linear and continuous in t, as discussed in the caption of Figure 2.

We now see that

$$\begin{aligned} r(t) &= \frac{\mathrm{d}}{\mathrm{d}t} \frac{\mathrm{E} \sum_{j=1}^{N_1} A_{1j} g(t, B_{1j})}{\mathrm{E} A_1} \\ &= \frac{\mathrm{E} \frac{\mathrm{d}}{\mathrm{d}t} \sum_{j=1}^{N_1} A_{1j} g(t, B_{1j})}{\mathrm{E} A_1} \\ &= \frac{\mathrm{E} \sum_{j=1}^{N_1} A_{1j} \frac{\mathrm{d}}{\mathrm{d}t} g(t, B_{1j})}{\mathrm{E} A_1} \end{aligned} \tag{16}$$

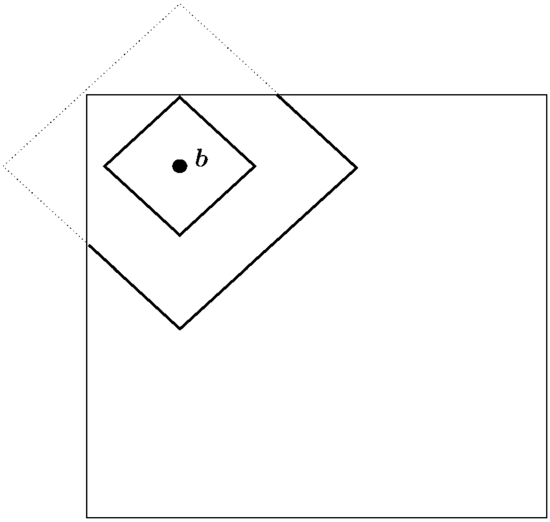


Fig. 2. Each diamond represents the set of points that are a fixed L_1 distance from the point b. The value $f(t, b)$ is proportional, for each fixed t, to the length of the (solid) line segments of the appropriate diamond that fall within the unit square. The value $g(t, b)$ gives the area of the intersection of the unit square with the appropriate diamond, and can also be viewed as an integral of f. Since f is continuous and piecewise linear, it follows that g is continuously differentiable and piecewise quadratic.

$$= \frac{\mathrm{E}\sum_{j=1}^{N_1} A_{1j} f(t, B_{1j})}{\mathrm{E}A_1}$$

$$= \frac{\mathrm{E}Y_1(t)}{\mathrm{E}A_1}, \tag{17}$$

where

$$Y_i(t) \triangleq \sum_{j=1}^{N_i} A_{ij} f(t, B_{ij}).$$

Of course, we need to justify the interchange of expectation and derivative in (16). We use the following result, which is stated in Glasserman (1991, p. 15) and proved in Dieudonné (1960, Section 8.5).

Theorem 6 (Generalized Mean-Value Theorem). *Let h be a continuous real-valued function on the closed interval $[a, b]$ which is differentiable everywhere except possibly on a set D of at most countably many points. Then for all x and $x + \delta$ in $[a, b]$,*

$$\left| \frac{h(x + \delta) - h(x)}{\delta} \right| \leqslant \sup_{y \in [a,b] \setminus D} \left| h'(y) \right|.$$

To justify (16) notice that $g(t, B_{1j})$, when viewed as a function of t, is continuously differentiable with derivative $f(t, B_{1j})$. Now, $0 \leqslant f(\cdot, \cdot) \leqslant 2$. Hence the generalized mean value theorem together with the dominated convergence theorem allow us to conclude that

$$\begin{aligned}
&\mathrm{E}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{j=1}^{N_1} A_{1j}g(t, B_{1j}) \\
&\quad = \mathrm{E}\lim_{\delta\to 0}\frac{\sum_{j=1}^{N_1} A_{1j}g(t+\delta, B_{1j}) - \sum_{j=1}^{N_1} A_{1j}g(t, B_{1j})}{\delta} \\
&\quad = \lim_{\delta\to 0}\mathrm{E}\frac{\sum_{j=1}^{N_1} A_{1j}g(t+\delta, B_{1j}) - \sum_{j=1}^{N_1} A_{1j}g(t, B_{1j})}{\delta} \\
&\quad = \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{E}\sum_{j=1}^{N_1} A_{1j}g(t, B_{1j}),
\end{aligned}$$

and the justification is complete.

Hence, the expression (17) rigorously defines $r(t)$. We now also have a means for estimating $r(t)$ using the estimator

$$r_n(t) = \frac{\sum_{i=1}^{n} Y_i(t)}{\sum_{i=1}^{n} A_i}.$$

So how can we assess the accuracy of the estimator $r_n(t)$? Certainly, the standard central limit theorem cannot be applied, because $r_n(t)$ is a *ratio* of sample means of i.i.d. observations. We first consider a strongly related question, and then return to the problem at hand.

Suppose that $X_1, X_2, \ldots$ is an i.i.d. sequence of random variables with finite mean $\mu = \mathrm{E}X_1$. Let $\overline{X}_n = n^{-1}\sum_{i=1}^{n} X_i$ denote the sample mean. If the real-valued function h is continuous at μ, it follows that $h(\overline{X}_n) \to h(\mu)$ a.s. as $n \to \infty$. So how does the error $h(\overline{X}_n) - h(\mu)$ behave, for large n? For large n, $\overline{X}_n$ will be very close to μ, and so the asymptotic behavior of the error should depend only on the local behavior of h near μ. Indeed, if h is appropriately differentiable, then Taylor's theorem implies that

$$h(\overline{X}_n) - h(\mu) \approx h'(\mu)(\overline{X}_n - \mu),$$

and so if X_1 has finite variance, then

$$\begin{aligned}
n^{1/2}\bigl(h(\overline{X}_n) - h(\mu)\bigr) &\approx h'(\mu)n^{1/2}(\overline{X}_n - \mu) \\
&\Rightarrow \eta\mathrm{N}(0, 1)
\end{aligned}$$

as $n \to \infty$, where $\eta^2 = h'(\mu)^2 \operatorname{var} X_1$.

This argument can be made rigorous and generalized to higher dimensions to obtain the following result, sometimes referred to as the delta method.

Theorem 7. *Suppose that $(X_n\colon n \geqslant 1)$ is an i.i.d. sequence of $\mathbb{R}^d$-valued random variables with $\mathrm{E}\|X_1\|_2^2 < \infty$. Let $\mu = \mathrm{E}X_1$ denote the mean vector and $\Lambda = \operatorname{cov} X_1$ denote the covariance matrix. Let $\overline{X}_n$ denote the sample mean of $X_1, \ldots, X_n$. If $h : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable in a neighborhood of μ with nonzero gradient $g = \nabla h(\mu)$ at μ, then*

$$n^{1/2}\bigl(h(\overline{X}_n) - h(\mu)\bigr) \Rightarrow \sigma \mathrm{N}(0, 1)$$

as $n \to \infty$, where $\sigma^2 = g^\top \Lambda g$.

For a proof, see Serfling (1980, p. 122).

To apply this result in our context, let

$$X_i = \bigl(Y_i(t), A_i\bigr),$$

and define $h(y, a) = y/a$. Theorem 7 then implies that

$$n^{1/2}\bigl(r_n(t) - r(t)\bigr) \Rightarrow \sigma(t)\mathrm{N}(0, 1),$$

where

$$\sigma^2(t) = \frac{\mathrm{E}(Y_1(t) - r(t)A_1)^2}{(\mathrm{E}A_1)^2}.$$

Using the SLLN, one can show that $\sigma^2(t)$ can be consistently estimated by

$$s_n^2(t) = \frac{n^{-1}\sum_{i=1}^n (Y_i(t) - r_n(t)A_i)^2}{(n^{-1}\sum_{i=1}^n A_i)^2},$$

and the same continuous mapping argument discussed earlier establishes that

$$r_n(t) \pm \frac{1.96 s_n(t)}{\sqrt{n}}$$

is an approximate 95% confidence interval for $r(t)$.

The estimator $r_n(t)$, being a ratio estimator, is biased. Taylor's theorem can be used to examine this bias. Reverting to our one-dimensional digression for the moment, Taylor's theorem implies that

$$h(\overline{X}_n) - h(\mu) \approx h'(\mu)(\overline{X}_n - \mu) + \frac{1}{2}h''(\mu)(\overline{X}_n - \mu)^2.$$

Taking expectations, we find that

$$\mathrm{E}h(\overline{X}_n) - h(\mu) \approx \frac{1}{2}h''(\mu)\frac{\operatorname{var} X_1}{n},$$

i.e., we have an explicit expression for the asymptotic bias. This argument can be made rigorous and generalized to higher dimensions.

Theorem 8. *Suppose that* $(X_n\colon n \geqslant 1)$ *is an i.i.d. sequence of* $\mathbb{R}^d$*-valued random variables with* $\mathrm{E}\|X_1\|_2^4 < \infty$*. Let* $\mu = \mathrm{E}X_1$ *denote the mean and* $\Lambda = \operatorname{cov} X_1$ *denote the covariance matrix. Let* $\overline{X}_n$ *denote the sample mean of* $X_1, \ldots, X_n$*. If* $h : \mathbb{R}^d \to \mathbb{R}$ *is such that* $h(\overline{X}_n)$ *is bounded for all n with probability* 1*, and twice continuously differentiable in a neighborhood of* μ*, then*

$$n\bigl(\mathrm{E}h(\overline{X}_n) - h(\mu)\bigr) \to \frac{1}{2}\sum_{i,j=1}^{d} \nabla^2 h(\mu)_{ij}\Lambda_{ij}$$

as $n \to \infty$*.*

The proof is a slight modification of Glynn and Heidelberger (1990, Theorem 7).

One of the hypotheses of Theorem 8 is that $h(\overline{X}_n)$ is bounded for all n a.s. This regularity condition is used in the proof to show that a certain sequence of random variables is uniformly integrable, so that one can pass expectations through to the limit. We did not need this condition in Theorem 7 because that result is about convergence of *distributions* and not about convergence of *expectations*. The boundedness condition can be replaced by less-stringent conditions like moment conditions.

We would like to apply Theorem 8 to the estimator $r_n(t)$. To that end, define $h(y, a) = y/a$. The only condition that is not obviously satisfied is the one that requires that $h(\overline{X}_n)$ be bounded for all n with probability 1. But the function f is bounded by 2, and so $h(\overline{X}_n(t)) = r_n(t)$ is also bounded by 2. We have therefore established that the bias in the estimator $r_n(t)$ is of the order n^{-1}.

It is reasonable to ask whether this bias is sufficient to noticeably affect the performance of the confidence intervals produced earlier for a given runlength n. Recall that the widths of the confidence intervals are of the order $n^{-1/2}$. Thus, the bias decreases at a (much) faster asymptotic rate than the width of the confidence intervals, and so when runlengths are sufficiently large it is reasonable to neglect bias.

In this section we used the delta method, which is very well known and sees a great deal of use. It has been applied in, for example, the regenerative method of steady-state simulation analysis, e.g., Shedler (1987), quantifying the impact of input uncertainty on simulation outputs, e.g., Cheng and Holland (1998) and in analyzing transformation-based simulation metamodels (Irizarry et al., 2003). We also used conditional Monte Carlo to enable the taking of derivatives.

5 Steady-state simulation

We now turn to useful mathematical techniques and results for steady-state simulation analysis. To this end we modify the assumptions of the previous section on the dynamics of the ambulance model. In addition to the assumptions given in Section 3, we assume that the ambulance operates 24 hours a day, 7 days a week. Furthermore, calls that arrive while the ambulance is busy are queued and answered in first-in first-out order. (The assumption that calls queue is not needed to create an interesting steady-state model. We introduce it to allow a nontrivial discussion about stability.) Once the current call is complete, either at the hospital if the patient required hospitalization or at the scene of the call if not, the ambulance then responds to the next call if any. (Recall that a call is completed either at the scene, with probability $1 - p$, or when the ambulance drops the patient off at the hospital, with probability p.) If the patient does not require hospitalization and no calls are queued, then the ambulance returns to the hospital/ambulance base, but can respond to newly-arriving calls before it reaches the hospital/ambulance base.

We are still interested in ambulance utilization and the distribution of response times. But the ambulance now handles all incoming calls, and so the fraction of calls answered by the ambulance is no longer relevant. Our performance measures are the following.

ρ The long-run utilization of the ambulance, i.e., the percentage of time that the ambulance is occupied with a call. The ambulance is not considered to be occupied when returning to the hospital/base without a patient.

r The long-run density of the response time to a call.

Notice that ρ is a deterministic constant, while r is a density function.

In the previous section we attempted to rigorously define the suggested performance measures, and also to derive asymptotic results that lay at the heart of confidence interval methodology for estimating them. We will proceed in a similar fashion in this section. Both performance measures involve the term “long-run”. In order that such long-run measures exist, it is first necessary that the ambulance model be stable, in the sense that calls do not “pile up” indefinitely. In order to be able to make statements about the stability, or lack thereof, of the model it is first necessary to define an appropriate stochastic process from which our performance measures can be derived. Statements about the stability of the model really relate to the stability of the stochastic process.

There are typically a host of stochastic processes that may be defined from the elements of a simulation. The choice of stochastic process depends partly on the performance measures. Given that our measures are related to response time, it is natural to consider a stochastic process that yields information on response times. Furthermore, for mathematical convenience, it is often helpful to ensure that one’s stochastic process is Markov.

For $n \geqslant 1$, let T_n denote the time at which the nth call is received, and define $T_0 = 0$. For $n \geqslant 1$, let W_n be the *residual workload* of the ambulance at time T_n+, i.e., just after the nth call is received. By residual workload at some time t, we mean the amount of time required for the ambulance to complete any current call, along with calls that might also be queued at time t. We assume that the ambulance is idle at the hospital at time $T_0 = 0$, so that $W_0 = 0$.

Unfortunately, $(W_n\colon\ n \geqslant 0)$ is not a Markov process, because the response time for a future call, and hence the workload, depends on the location of the ambulance when the ambulance clears the previous workload. (Here, the ambulance may clear the previous workload either at the location of a call if hospitalization is unnecessary, or at the hospital/base if hospitalization is necessary.) So if we also keep track of the location $\beta_n = (\beta_n(1), \beta_n(2))$ of the ambulance at the instant at which the workload W_n is first cleared, then the resulting process $Z = (Z_n\colon\ n \geqslant 0)$ is Markov, where $Z_n = (W_n, \beta_n)$.

The process Z is a general state space Markov chain, and evolves on the state space

$$S = [0, \infty) \times \left[-\tfrac{1}{2}, \tfrac{1}{2}\right]^2.$$

The first step in ensuring that our long-run performance measures are defined is to establish that Z exhibits some form of positive recurrence. One way to achieve this is to verify that the chain Z satisfies the following condition, which will be explained shortly.

To avoid confusion between general results and those for our particular model, we will state general results in terms of a Markov chain $X = (X_n\colon\ n \geqslant 0)$ evolving on a state space $\mathcal{S}$.

The First Lyapunov Condition (FLC). There exists a nonempty $B \subseteq \mathcal{S}$, positive scalars $a < 1$, b and δ, an integer $m \geqslant 1$, a probability distribution φ on $\mathcal{S}$, and a function $V : \mathcal{S} \to [1, \infty)$ such that

(1) $\mathrm{P}(X_m \in \cdot | X_0 = z) \geqslant \delta\varphi(\cdot)$ for all $z \in B$, and
(2) $\mathrm{E}(V(X_1)|X_0 = z) \leqslant aV(z) + bI(z \in B)$ for all $z \in \mathcal{S}$.

The FLC (sometimes called a Foster–Lyapunov condition) is a stronger Lyapunov condition than we really require, but it simplifies the presentation. The function V is called a Lyapunov (think of energy) function. The second requirement basically states that when the chain X lies outside of the set B, the energy in the system tends to decrease, and when the chain lies inside B, the energy in the system cannot become too big on the next step. This condition implies that the set B gets hit infinitely often. Of course, if one takes $B = \mathcal{S}$, the entire state space, then this requirement is trivially satisfied. The first condition is needed to ensure that the set B is not too “big”.

In any case, the point is that if a chain X satisfies the FLC, then X is appropriately positive recurrent, and in particular has a unique stationary probability distribution. In fact, the FLC is essentially equivalent to a strong form of ergodicity. It is therefore reasonable (but nonstandard) to *define* a chain as being appropriately ergodic if the FLC holds.

Definition 2. We say that a discrete time Markov chain X is V-uniformly ergodic if it satisfies the Lyapunov condition and is aperiodic.

The aperiodicity condition is not strictly necessary for much of what follows, but we impose it to be consistent with the term "ergodic". For more on V-uniform ergodicity and its relationship to the FLC see Meyn and Tweedie (1993, Chapter 16).

Does our chain Z satisfy the FLC? The answer is yes, and it is instructive to go through a proof. However, on a first reading one may skip the following development up to the statement of Proposition 9 without loss of continuity.

For many systems, the function V may be taken to be $\mathrm{e}^{\gamma v}$, where v is some measure of the work in the system. In fact, as we now show, one may take $V(w, b) = \mathrm{e}^{\gamma w}$ for some yet to be determined constant $\gamma > 0$.

Consider what happens on a single transition of the chain Z starting from the point $Z_n = (w, b)$, where $n \geqslant 0$. The workload decreases at unit rate, at least until it hits 0, until the arrival of the next call over an interval of length $\tau_{n+1} = T_{n+1} - T_n$. At time T_{n+1} a new call arrives at location C_{n+1} and adds some work to the workload. In particular, there will be some travel time η_{n+1} to the scene of the call, some time U_{n+1} spent at the scene, and then potentially some travel time ξ_{n+1} to transport the patient to the hospital. If the patient requires transport to the hospital then $\beta_{n+1} = (0, 0)$, which is the location of the hospital. If not, then $\beta_{n+1} = C_{n+1}$, which is the location of the call, and $\xi_{n+1} = 0$. If hospitalization is not necessary and no calls are queued when the ambulance completes service at the scene, then the ambulance returns to the hospital/base, but this travel time is not counted as workload because the ambulance is free to respond to a new call, albeit not necessarily from the hospital/base.

So for $n \geqslant 0$, the new workload W_{n+1} is given by $W_{n+1} = [W_n - \tau_{n+1}]^+ + Q_{n+1}$, where $[x]^+ = \max\{x, 0\}$, and $Q_n = \eta_n + U_n + \xi_n$. Recall that we assume that the scene times $(U_n\colon n \geqslant 1)$ are i.i.d. gamma-distributed random variables, and are independent of all other quantities. We assume that the call location sequence $(C_n\colon n \geqslant 1)$ is i.i.d. and independent of all other quantities.

Equipped with this Lindley-type recursion for the workload, we can now attempt to identify conditions under which the Lyapunov condition will hold. We use the fact that $Q_1 \leqslant 3 + U_1$ because $\eta_1 \leqslant 2$ and $\xi_1 \leqslant 1$ (recall that the ambulance travels distances as measured by the Manhattan metric). If $z = (w, b)$, then $\mathrm{E}[V(Z_1)|Z_0 = z]$ is given by

$$
\begin{aligned}
\mathrm{E}\mathrm{e}^{\gamma([w-\tau_1]^+ + Q_1)} &\leqslant \mathrm{E}\big[\mathrm{e}^{\gamma[w-\tau_1]^+}\mathrm{e}^{\gamma(3+U_1)}\big] \\
&= \mathrm{E}\mathrm{e}^{\gamma[w-\tau_1]^+}\,\mathrm{E}\mathrm{e}^{\gamma(3+U_1)} \\
&\leqslant \big[\mathrm{E}\mathrm{e}^{\gamma(w-\tau_1)} + \mathrm{P}(w - \tau_1 < 0)\big]\,\mathrm{E}\mathrm{e}^{\gamma(3+U_1)} \\
&= \mathrm{e}^{\gamma w}\big[\mathrm{E}\mathrm{e}^{-\gamma\tau_1} + \mathrm{e}^{-(\lambda+\gamma)w}\big]\,\mathrm{E}\mathrm{e}^{\gamma(3+U_1)} && (18) \\
&= \mathrm{e}^{\gamma w}\left[1 + \frac{\lambda+\gamma}{\lambda}\mathrm{e}^{-(\lambda+\gamma)w}\right]\mathrm{E}\mathrm{e}^{\gamma(3+U_1-\tau_1)} && (19) \\
&= V(z)\left[1 + \frac{\lambda+\gamma}{\lambda}\mathrm{e}^{-(\lambda+\gamma)w}\right]\phi(\gamma), && (20)
\end{aligned}
$$

where ϕ is the moment generating function of $3 + U_1 - \tau_1$. Equation (18) uses the fact that $\mathrm{P}(\tau_1 > w) = \mathrm{e}^{-\lambda w}$, while (19) follows since $\mathrm{E}\mathrm{e}^{-\gamma\tau_1} = \lambda/(\lambda + \gamma)$ (when $\gamma > -\lambda$).

Since $\mathrm{E}\mathrm{e}^{tU_1}$ is finite in a neighborhood of 0, i.e., U_1 has a moment generating function defined near 0, we have that $\phi(0) = 1$, and

$$\phi'(0) = \mathrm{E}(U_1 + 3 - \tau_1).$$

So if $\mathrm{E}U_1 + 3 < \mathrm{E}\tau_1$, then $\phi'(0) < 0$, and so $\phi(t) < 1$ for $t > 0$ in some neighborhood of 0. So fix $\gamma > 0$ so that $\phi(\gamma) < 1$.

Now, there is some $K > 0$ such that if $w > K$, then

$$\left[1 + \frac{\lambda + \gamma}{\lambda}\mathrm{e}^{-(\lambda+\gamma)w}\right]\phi(\gamma) < 1. \tag{21}$$

Furthermore, for $w \leqslant K$, we have that

$$\mathrm{E}\big[V(Z_1)|Z_0 = z\big] \leqslant \mathrm{E}\mathrm{e}^{\gamma(K+3+U_1)} < \infty. \tag{22}$$

Thus, if we take $B = [0, K] \times [-\frac{1}{2}, \frac{1}{2}]^2$, then it follows from (20)–(22) that the second requirement in the FLC is met.

It remains to check the first requirement. Suppose that $Z_n = (w, b) \in B$ so that the current workload $w \leqslant K$. If the time τ_{n+1} till the next call is large enough, then irrespective of whether the nth patient requires transport to the hospital or not, the ambulance will have reached the hospital and be available to respond to a new call by the time the $(n+1)$st call arrives. So if $\tau_{n+1} > K+1$, the $(n + 1)$st call will be served immediately by the ambulance from the base. In fact, the chain regenerates at such times. Let

$$\delta = \mathrm{P}(\tau_1 > K + 1) = \mathrm{e}^{-\lambda(K+1)}$$

and φ denote the distribution of $Z_1 = (W_1, B_1)$ assuming that just before time T_1 the ambulance is free and located at the hospital. Then we have that for all $z \in B$,

$$\mathrm{P}(z, \cdot) \geqslant \delta\varphi(\cdot),$$

and the first requirement in the FLC is satisfied.

We have established that Z satisfies the FLC. It is straightforward to show that Z is aperiodic, and so we arrive at the following result.

Proposition 9. *If* $\mathrm{E}U_1 + 3 < \mathrm{E}\tau_1$*, then the chain* Z *is* V*-uniformly ergodic, where* $V(w, b) = \mathrm{e}^{\gamma w}$ *for some* $\gamma > 0$*.*

The stability condition

$$\mathrm{E}U_1 + 3 < \mathrm{E}\tau_1$$

has an appealing interpretation. The left-hand side of the inequality gives an upper bound on the expected amount of work (travel time to the scene + time at the scene + travel time from the scene to the hospital) brought in by an arriving call. We require this to be smaller than the expected amount of time that the ambulance has available between calls to deal with the work. This condition can be weakened by being more careful about defining how much work each call brings to the system, but this is not something that we will pursue further.

The main point is that Proposition 9 gives *easily-verifiable* conditions under which the system is stable. While it may have appeared somewhat difficult to verify the Lyapunov condition, the argument used is actually quite straightforward once one picks an appropriate function V. The difficult part (in general), and the part where one's insight into the underlying process plays a key role, is in choosing an appropriate V. Thankfully, we will see that the payoff from verifying the Lyapunov condition is certainly worth the effort. Based on this result, we can now define our performance measures rigorously, and also construct estimators that are consistent and satisfy central limit theorems.

As in Section 4, the rigorous definition of our performance measures is based on the strong law of large numbers. For simplicity, we state this theorem under stronger hypotheses than are really necessary. Let E_ν denote expectation for the path space of a Markov chain with initial distribution ν.

Theorem 10 (MCSLLN). *Let X be a V-uniformly ergodic Markov chain on state space $\mathcal{S}$ with stationary probability distribution π. Let $h : \mathcal{S} \to \mathbb{R}$ be a real-valued function on $\mathcal{S}$. If $\pi|h| = \mathrm{E}_\pi|h(X_0)| = \int_{\mathcal{S}} |h(x)|\pi(\mathrm{d}x) < \infty$, then*

$$\frac{1}{n}\sum_{i=0}^{n-1} h(X_i) \to \pi h \quad a.s.$$

as $n \to \infty$.

For a proof, see Meyn and Tweedie (1993, Theorem 17.0.1).

Assuming V-uniform ergodicity we see that if h is bounded, then the MCSLLN holds. This will be sufficient for our purposes in this section. However, sometimes one is also interested in unbounded h. So long as $|h(z)| \leqslant cV(z)$ for some $c > 0$ and all z, then $\pi|h| < \infty$; see Meyn and Tweedie (1993, Theorem 14.3.7).

We turn now to the performance measure ρ, the long-run utilization of the ambulance. The actual utilization of the ambulance over the time interval $[0, T_n)$, i.e., up until the time of the nth arrival is

$$\frac{n^{-1}\sum_{i=0}^{n-1}\min\{W_i, \tau_{i+1}\}}{n^{-1}\sum_{i=1}^{n}\tau_i}. \tag{23}$$

Now, the SLLN for i.i.d. random variables implies that the denominator converges to λ^{-1}. We would like to apply the MCSLLN to the numerator, but it is not yet in an appropriate form. There are several ways to proceed here. One way is to enlarge the state space of the chain, and we will demonstrate this method shortly in analyzing the density r. Another approach is to apply filtering; see Glasserman (1993), which can be viewed as a variant of conditional Monte Carlo. We have that

$$\begin{aligned}\mathrm{E}\min\{w, \tau_1\} &= w\,\mathrm{P}(\tau_1 > w) + \mathrm{E}\tau_1 I(\tau_1 \leqslant w)\\ &= \lambda^{-1}\bigl(1 - \mathrm{e}^{-\lambda w}\bigr),\end{aligned}$$

and so we replace (23) by

$$\rho_n = \frac{1}{n}\sum_{i=0}^{n-1}\bigl(1 - \mathrm{e}^{-\lambda W_i}\bigr). \tag{24}$$

Notice that ρ_n is in exactly the form that we need to apply the MCSLLN, with $h(w, b) = 1 - \mathrm{e}^{-\lambda w}$ which is bounded, and so we find that

$$\rho_n \to \rho \quad \text{a.s.}$$

as $n \to \infty$. This then is a rigorous definition of ρ, and also a proof that the estimator ρ_n is strongly consistent.

Let us turn now to r, the steady-state density of the response time to a call. For $n \geqslant 0$, the response time L_{n+1} to the call arriving at time T_{n+1} is the sum of the workload $[W_n - \tau_{n+1}]^+$ just before the arrival of the call and the time η_{n+1} for the ambulance to travel to the location of the new call. The travel time to the new call is given by the distance $d(B_{n+1}, C_{n+1})$ between the call location C_{n+1} and the location B_{n+1} of the ambulance at the time when the ambulance responds to the $(n + 1)$st call. Now, $B_{n+1} = \beta_n$ if the $(n + 1)$st call arrives before the previous workload is cleared, i.e., $W_n \geqslant \tau_{n+1}$. If $W_n < \tau_{n+1}$ then the new call arrives after the ambulance completes the previous workload, so the ambulance may be on its way to the hospital, or at the hospital, when the $(n+1)$st call arrives. In any case, the location B_{n+1} is a deterministic function of $Z_n = (W_n, \beta_n)$ and τ_{n+1}. So the response time L_{n+1} depends not only on Z_n, but also on τ_{n+1} and C_{n+1}.

This dependence of the response time on additional quantities beyond those in the state space of our chain causes some difficulties in our analysis. We could again apply filtering, but let us consider an alternative approach. We expand the state space of the Markov chain so that it is "sufficiently rich" to supply all of the needed information.

Define $\widetilde{Z} = (\widetilde{Z}_n\colon n \geqslant 0)$ where, for $n \geqslant 0$, $\widetilde{Z}_n = (W_n, \beta_n, \tau_{n+1}, C_{n+1})$. Using techniques that are very similar to those used for the chain Z, we can show that $\widetilde{Z}$ is a $\widetilde{V}$-uniformly ergodic chain on the state space

$$\widetilde{S} = [0, \infty) \times \bigl[-\tfrac{1}{2}, \tfrac{1}{2}\bigr]^2 \times [0, \infty) \times \bigl[-\tfrac{1}{2}, \tfrac{1}{2}\bigr]^2,$$

where

$$\widetilde{V}(w, b, t, c) = \mathrm{e}^{\gamma[w-t]^+}$$

for some $\gamma > 0$.

To define the density r we first define the corresponding distribution function R, and then differentiate.

Recall that for $n \geqslant 0$, the response time is given by

$$L_{n+1} = [W_n - \tau_{n+1}]^+ + d(B_{n+1}, C_{n+1}).$$

Consider the *empirical* response time distribution function based on the first n response times

$$\frac{1}{n}\sum_{i=1}^{n} I(L_i \leqslant \cdot).$$

Notice that $I(L_i \leqslant t)$ is a deterministic and bounded function of $\widetilde{Z}_{i-1}$, so we can apply the MCSLLN to assert that

$$\frac{1}{n}\sum_{i=1}^{n} I(L_i \leqslant t) \to R(t)$$

as $n \to \infty$ a.s., for any fixed t, where

$$R(t) = \mathrm{E}_{\tilde{\pi}} I(L_1 \leqslant t).$$

Here $\tilde{\pi}$ refers to the stationary distribution of $\widetilde{Z}$.

It is not yet clear how to obtain an expression for the density r, since the indicator functions that we used to define R are not differentiable. We need to perform some sort of smoothing. We again use conditional Monte Carlo. Notice that

$$\begin{aligned} R(t) &= \mathrm{E}_{\tilde{\pi}} I(L_1 \leqslant t) \\ &= \mathrm{E}_{\tilde{\pi}} \mathrm{P}_{\tilde{\pi}}\bigl([W_0 - \tau_1]^+ + d(B_1, C_1) \leqslant t | W_0, \beta_0, \tau_1\bigr) \\ &= \mathrm{E}_{\tilde{\pi}} \mathrm{P}_{\tilde{\pi}}\bigl(d(B_1, C_1) \leqslant t - [W_0 - \tau_1]^+ | W_0, \beta_0, \tau_1\bigr) \\ &= \mathrm{E}_{\tilde{\pi}} g\bigl(t - [W_0 - \tau_1]^+, B_1\bigr), \end{aligned}$$

where the function $g(\cdot, \cdot)$ was introduced in (12). (Here we extend the definition so that $g(t, b) = 0$ for $t < 0$.) Notice that B_1 can be determined from W_0, β_0 and τ_1. So we can estimate $R(t)$ using

$$R_n(t) = \frac{1}{n}\sum_{i=0}^{n-1} g\bigl(t - [W_i - \tau_{i+1}]^+, B_{i+1}\bigr),$$

and again the MCSLLN shows that $R_n(t) \to R(t)$ as $n \to \infty$ a.s., for each fixed t.

We can now define r, via

$$\begin{aligned} r(t) &= \frac{\mathrm{d}}{\mathrm{d}t}R(t) \\ &= \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{E}_{\tilde{\pi}}g\bigl(t-[W_0-\tau_1]^+, B_1\bigr) \\ &= \mathrm{E}_{\tilde{\pi}}\,\frac{\mathrm{d}}{\mathrm{d}t}g\bigl(t-[W_0-\tau_1]^+, B_1\bigr) \qquad (25) \\ &= \mathrm{E}_{\tilde{\pi}}f\bigl(t-[W_0-\tau_1]^+, B_1\bigr), \qquad (26) \end{aligned}$$

where $f(\cdot,\cdot)$ was defined in (15). Of course, we need to justify the interchange of derivative and expectation in (25). This is virtually identical to the justification given for the interchange (16), and so we omit the details.

Equation (26) defines r, and immediately suggests an estimator for $r(t)$ given by

$$r_n(t) = R_n'(t) = \frac{1}{n}\sum_{i=0}^{n-1} f\bigl(t-[W_i-\tau_{i+1}]^+, B_{i+1}\bigr).$$

The MCSLLN shows that $r_n(t) \to r(t)$ as $n \to \infty$ a.s. for each fixed t. Hence, we have rigorously defined the density r, and established that it can be consistently estimated by r_n.

We now turn to the error in the estimators. As before, error can be assessed through confidence intervals derived from a central limit theorem. In great generality, the error $n^{-1}\sum_{i=0}^{n-1} h(X_i) - \pi h$ is approximately normally distributed with mean 0 and variance σ^2/n, exactly as in the i.i.d. case. The difference here is that we are averaging dependent random variables rather than independent ones, and this difference is exhibited through the variance constant which now includes covariance terms in addition to the variance (under π) of $h(X_0)$.

For simplicity we state the Markov chain central limit theorem under stronger conditions than are strictly necessary.

Theorem 11 (MCCLT). *Suppose that the chain X is V-uniformly ergodic. Then, for any function $h:\mathcal{S} \to \mathbb{R}$ with $h^2(z) \leqslant cV(z)$ for some $c > 0$ and all z,*

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=0}^{n-1} h(X_i) - \pi h\right) \Rightarrow \sigma\mathrm{N}(0,1),$$

where π is the stationary probability distribution of X, and

$$\sigma^2 = \mathrm{var}_\pi\bigl[h(X_0)\bigr] + 2\sum_{k=1}^{\infty}\mathrm{cov}_\pi\bigl[h(X_0), h(X_k)\bigr]. \qquad (27)$$

For a proof, see Meyn and Tweedie (1993, Theorem 17.0.1).

We immediately obtain the following result.

Proposition 12. *Under the conditions of Proposition 9,*

$$\sqrt{n}(\rho_n - \rho) \Rightarrow \sigma_\rho \mathrm{N}(0, 1)$$

as $n \to \infty$, for an appropriately defined constant σ_ρ^2. In addition, for all $t > 0$,

$$\sqrt{n}\big(r_n(t) - r(t)\big) \Rightarrow \sigma(t)\mathrm{N}(0, 1)$$

as $n \to \infty$, for an appropriately defined constant $\sigma^2(t)$.

Thus, just as in the terminating simulation case, the error in the estimators ρ_n and $r_n(t)$ is approximately normally distributed with mean 0 and variance on the order of n^{-1}.

Proposition 12 serves as a foundation for constructing confidence intervals for our performance measures. One approach is to estimate the variance constants directly using the regenerative method (see Chapter 16), which is certainly easily applied to our example. But the method of batch means is, at least currently, more widely applicable and so we instead consider this approach. See Chapter 15 for a more extensive discussion of the method of batch means.

Suppose that we have a sample path $X_0, X_1, \ldots, X_{n-1}$. Divide this sample path into l batches each of size m, where for convenience we assume that $n = ml$, so that the kth batch consists of observations $X_{(k-1)m}, \ldots, X_{km-1}$. (We use "$l$" for the number of batches instead of the traditional "b" since we have already used "b" to describe ambulance locations.) Now, for $k = 1, \ldots, l$, let M_k be the sample mean over the kth batch, i.e.,

$$M_k = \frac{1}{m} \sum_{i=(k-1)m}^{km-1} h(X_i),$$

and let $\overline{M}_l$ denote the sample mean of the l batch means $M_1, \ldots, M_l$. Finally, let

$$s_l^2 = \frac{1}{l-1} \sum_{k=1}^{l} (M_k - \overline{M}_l)^2$$

denote the sample variance of the M_k's. The method of batch means provides a confidence interval for πh of the form $\overline{M}_l \pm t s_l/\sqrt{l}$, for some constant t, and relies on the assumption that for large n, $(\overline{M}_l - \pi h)/(s_l/\sqrt{l}\,)$ is approximately t-distributed, with $l - 1$ degrees of freedom.

The MCCLT above suggests that as $n \to \infty$ with l, the number of batches, held fixed, all of the batch means are asymptotically normally distributed with mean πh and variance $l\sigma^2/n$. If each of the batch means are also asymptotically independent, then a standard result (e.g., Rice, 1988, p. 173) shows that the above confidence interval methodology is valid.

A sufficient condition that supplies both the asymptotic normality and asymptotic independence is that the chain X satisfy a functional central limit theorem; see Schruben (1983) and Glynn and Iglehart (1990), from which much of the following discussion is adapted.

Definition 3. Let X be a Markov chain on state space $\mathcal{S}$, and let $h:\mathcal{S} \to \mathbb{R}$. For $0 \leqslant t \leqslant 1$, let

$$\overline{X}_n(t) = n^{-1} \sum_{k=0}^{\lfloor (n-1)t \rfloor} h(X_k)$$

and set

$$\zeta_n(t) = n^{1/2}\bigl(\overline{X}_n(t) - \kappa t\bigr)$$

for some constant κ. We say that X satisfies a functional central limit theorem (FCLT) if there exists a $\xi > 0$ such that $\zeta_n \Rightarrow \xi B$ as $n \to \infty$, where B denotes a standard Brownian motion.

Observe that if X satisfies an FCLT, then the jth batch mean M_j can be expressed as

$$\begin{aligned} M_j &= l\left[\overline{X}_n\left(\frac{j}{l}\right) - \overline{X}_n\left(\frac{j-1}{l}\right)\right] \\ &= \kappa + n^{-1/2} l\left[\zeta_n\left(\frac{j}{l}\right) - \zeta_n\left(\frac{j-1}{l}\right)\right]. \end{aligned}$$

Since the increments of Brownian motion are independent and normally distributed, the FCLT then implies that the M_j's are asymptotically independent and normally distributed with mean κ and variance $l\xi^2/n$. Thus, under an FCLT assumption, the batch means confidence interval methodology outlined above is asymptotically valid as $n \to \infty$ with l fixed.

So when can we be sure that X satisfies an FCLT? One sufficient condition is the following result.

Theorem 13. *Suppose that X is V-uniformly ergodic and $h^2(z) \leqslant cV(z)$ for all z and some $c > 0$. If the constant σ^2 defined in (27) above is positive, then X satisfies a functional central limit theorem with $\kappa = \pi h$ and $\xi^2 = \sigma^2$.*

For a proof, see Meyn and Tweedie (1993, Theorems 17.4.4, 17.5.3).

Notice that we have already established that the conditions of Theorem 13 hold for our estimators. Thus, we immediately arrive at the conclusion that the method of batch means yields asymptotically valid confidence intervals when used in conjunction with our estimators. In fact, an FCLT is sufficient to ensure that any standardized time series method (batch means with a fixed number of batches is one such method) is asymptotically valid (Schruben, 1983).

As in the terminating simulation case, the performance of confidence interval procedures may be negatively impacted by bias. The bias depends on the initial distribution, μ say, of the chain. The bias in the estimator ρ_n is $\mathrm{E}_\mu \rho_n - \rho$, with a similar expression for the bias in $r_n(t)$ for each $t > 0$.

We give the appropriate calculations for ρ, as those for r are similar. Let $h(w, b) = 1 - \mathrm{e}^{-\lambda w}$. Using a standard technique (e.g., Glynn, 1995), we see that the bias in ρ_n under initial distribution μ is

$$\begin{aligned}
&\mathrm{E}_\mu \frac{1}{n} \sum_{i=0}^{n-1} \bigl[h(Z_i) - \pi h\bigr] \\
&\quad = \frac{1}{n} \sum_{i=0}^{\infty} \bigl[\mathrm{E}_\mu h(Z_i) - \pi h\bigr] - \frac{1}{n} \sum_{i=n}^{\infty} \bigl[\mathrm{E}_\mu h(Z_i) - \pi h\bigr] \\
&\quad = \frac{\nu}{n} + \mathrm{o}\bigl(n^{-1}\bigr),
\end{aligned}$$

where

$$\nu = \sum_{i=0}^{\infty} \bigl[\mathrm{E}_\mu h(Z_i) - \pi h\bigr]$$

provided that

$$\sum_{i=0}^{\infty} \bigl|\mathrm{E}_\mu h(Z_i) - \pi h\bigr| < \infty. \tag{28}$$

So the bias in the estimator ρ_n will be of the order n^{-1} if (28) holds. This result holds in great generality.

Theorem 14. *Suppose that X is V-uniformly ergodic. Let π be the stationary probability distribution of X. If $|h(z)| \leqslant cV(z)$ for all z and some $c < \infty$, and $\mu V < \infty$, then*

$$\sum_{i=0}^{\infty} \bigl|\mathrm{E}_\mu h(X_i) - \pi h\bigr| < \infty$$

and

$$\mathrm{E}_\mu \frac{1}{n} \sum_{i=0}^{n-1} h(X_i) - \pi h = \frac{\nu}{n} + \mathrm{O}\bigl(q^n\bigr)$$

as $n \to \infty$, where $q < 1$ and

$$\nu = \sum_{i=0}^{\infty} \bigl[\mathrm{E}_\mu h(X_i) - \pi h\bigr].$$

This result is a straightforward consequence of Meyn and Tweedie (1993, Theorem 16.0.1).

We can conclude from Theorem 14 that if the initial conditions are chosen appropriately (e.g., if $\widetilde{Z}_0$ is chosen to be deterministic), then the bias of our estimators is of the order n^{-1}.

Recall that the batch means $M_1, \ldots, M_l$ are asymptotically normally distributed with variance $l\chi^2/n$. Their standard deviation is therefore of the order $n^{-1/2}$, and so the width of the batch means confidence interval is also of the order $n^{-1/2}$. The bias in the estimators is of the order n^{-1}, and so it follows that bias will not play a role for large runlengths.

5.1 Multiple ambulances

Sometimes a good choice of Lyapunov function immediately presents itself. In other cases the choice is not so clear, and the process of finding a good function becomes more of an art than a science. Here we consider the case where multiple ambulances operate in the unit square from potentially different bases. We will again look for a good choice of Lyapunov function. Some natural choices do not work, at least at first sight. However, one of those choices *does* work if we use an extension of the FLC.

Suppose now that we have ℓ identical ambulances where ambulance i operates out of a base located at the point $d_i \in [-1/2, 1/2]^2$, $i = 1, \ldots, \ell$. Some ambulances may operate from the same base, in which case some of the d_is take the same value. The dynamics of the system are as follows. Calls are answered in first-in first-out order. When a call is received, a dispatcher assigns the call to the closest available ambulance. If no ambulances are available, then the first one that becomes available is selected. Ties are broken through random uniform selection. This dispatching policy does not necessarily minimize response times because the selected ambulance may be far from the call, and a closer ambulance that will soon be free might get to the call sooner. The details of exactly which ambulance is selected are not too important from our standpoint, so long as a sensible rule is used that spreads the workload among the available ambulances.

After traveling to the call location, the ambulance spends some time at the scene after which, with probability p, the patient is transported to the hospital, which is again at the point $(0, 0)$. In this case, after reaching the hospital the patient is instantaneously dropped off and the ambulance is freed for other work, typically returning to its base. If the patient does not require hospitalization, then after the scene time is complete the ambulance is freed for other work, again typically returning to its base. We allow redirection, where the ambulance may be redirected to a new call before it reaches its base.

A natural Markov chain that models this process is $Z = (Z_n\colon n \geqslant 0)$, where $Z_n = (W_n(i), \beta_n(i)\colon i = 1, \ldots, \ell)$. Here $W_n(i)$ gives the workload for ambulance i associated with all calls that have been received up to, and including, call n. Notice that the workload will be associated with only a subset of the first

n calls since there are multiple ambulances. The vector $\beta_n(i) \in [-1/2, 1/2]^2$ gives the location in the unit square where ambulance i will be located when the workload $W_n(i)$ is first cleared. This is the hospital if its last patient is hospitalized, and the location of its last call if not.

Under what conditions is the chain stable? Consider the work associated with a single call. The ambulance first needs to travel to the call from its current location, taking at most 2 time units. It then tends to the patient at the scene, taking, on average, $\mathrm{E}U$ time units. It may then transport the patient to the hospital, taking at most 1 time unit. Therefore, a bound on the expected amount of work brought in by each call is again $3 + \mathrm{E}U$. Since there are ℓ ambulances we expect that the system will be stable if

$$3 + \mathrm{E}U < \ell \mathrm{E}\tau,$$

where τ is a random variable representing the time between calls. To verify this belief we can appeal to the FLC.

As already mentioned, the FLC is actually stronger than required to establish stability. For example, we have seen that it is also useful for proving that certain steady-state expectations are finite. A "tighter" condition is the following one.

The Second Lyapunov Condition (SLC). There exists a nonempty $B \subseteq \mathcal{S}$, positive scalars ε, b and δ, an integer $m \geqslant 1$, a probability distribution φ on $\mathcal{S}$, and a function $V : \mathcal{S} \to [0, \infty)$ such that

(1) $\mathrm{P}(X_m \in \cdot | X_0 = z) \geqslant \delta\varphi(\cdot)$ for all $z \in B$, and
(2) $\mathrm{E}(V(X_1)|X_0 = z) \leqslant V(z) - \varepsilon + bI(z \in B)$ for all $z \in \mathcal{S}$.

The only change in this definition from the previous one is requirement (2). Here the nonnegative function V again represents energy, and requirement (2) states that the energy tends to decrease when the chain is outside the set B. If the Markov chain satisfies the SLC, then it is again positive recurrent in a certain precise sense; see Meyn and Tweedie (1993, Theorem 13.0.1).

The FLC implies the SLC. To see why, notice that if the chain satisfies requirement (2) of the FLC, then

$$\begin{aligned} \mathrm{E}\bigl(V(X_1)|X_0 = z\bigr) &\leqslant aV(z) + bI(z \in B) \\ &= V(z) - (1-a)V(z) + bI(z \in B) \\ &\leqslant V(z) - (1-a) + bI(z \in B), \end{aligned}$$

where the final inequality follows since $V(z) \geqslant 1$ in the FLC.

So now let us turn to finding a function V that satisfies the SLC for the multiple-ambulance case.

First consider requirement (1). Suppose the workloads $w(i)$, $i = 1, \ldots, \ell$, are all at most K, say. If $\tau_1 > K + 2$, then when the next call is received, all of the ambulances will be at their bases. Therefore, when $\tau_1 > K + 2$, Z_1 has a certain distribution φ and is independent of Z_0. (In fact, the chain regenerates at such times.) Now τ_1 is exponentially distributed, and therefore $P(\tau_1 > K + 2) > 0$. Hence, requirement (1) of the SLC is satisfied when $B = \{z\colon w(i) \leqslant K\}$ for any $K > 0$ where $z = (w(i), \beta(i)\colon 1 \leqslant i \leqslant \ell)$.

Next we turn to requirement (2). It is natural to try $V(z) = w(1) + \cdots + w(\ell)$, the sum of the workloads of the ambulances. Consider what happens on a single step of the Markov chain. Let D_i be the index of the ambulance that responds to the ith call, and Q_i denote the time required for this ambulance to travel to the scene, treat the patient at the scene and, if necessary, transport the patient to the hospital. Then

$$\begin{aligned} \mathrm{E}\bigl[V(Z_1)|Z_0 = z\bigr] &= \mathrm{E}\sum_{i=1}^{\ell}\bigl(\bigl[w(i) - \tau_1\bigr]^+ + Q_1 I(D_1 = i)\bigr) \\ &= \mathrm{E}Q_1 + \sum_{i=1}^{\ell}\mathrm{E}\bigl[w(i) - \tau_1\bigr]^+. \end{aligned} \tag{29}$$

If all of the $w(i)$s are large, then $\mathrm{E}[w(i) - \tau_1]^+ \approx w(i) - \mathrm{E}\tau_1$ for each i. So then

$$\mathrm{E}\bigl[V(Z_1)|Z_0 = z\bigr] - V(z) \approx \mathrm{E}Q_1 - \ell\,\mathrm{E}\tau_1$$

which is negative under our conjectured stability condition as desired. But when one or more of the $w(i)$s is "small", this heuristic argument breaks down. In fact, what can happen in this case is that the overall workload as measured by V increases! The problem is that while the overall work in the system may be high, the work is not evenly shared by the ambulances, and so some may be idle when they are greatly needed.

Perhaps we chose a poor Lyapunov function V? An alternative is $V(z) = \max_i w(i)$, the maximum workload of an ambulance. One finds that it works well when one of the workloads is large and the rest are small, but it fails when all of the workloads are large and roughly equal.

Both of our choices of Lyapunov function have failed. However, both *can* be made to work. We will show how with our first choice, the sum of the workloads of the ambulances. In order to gain some insight it is helpful to consider the drift of the Markov chain. To visualize this consider the case $\ell = 2$, so there are only 2 ambulances. Consider the workloads of the ambulances as a point in the plane, and compute the expected change in workload levels in a single transition as a vector. This can be visualized as in Figure 3. The drift arrows have been normalized to ensure that they do not cross, to ensure that the figure remains uncluttered. The diagonal lines represent lines of constant workload.

Notice that away from the boundary where both workloads are large, the drift is toward lower workload, as desired. But near the boundaries, the drift is toward higher workload, at least for a short time. This plot suggests that the one-step drift of the sum of the workloads is negative "away from the boundary", but not so near the boundary. It also suggests that if we are near the

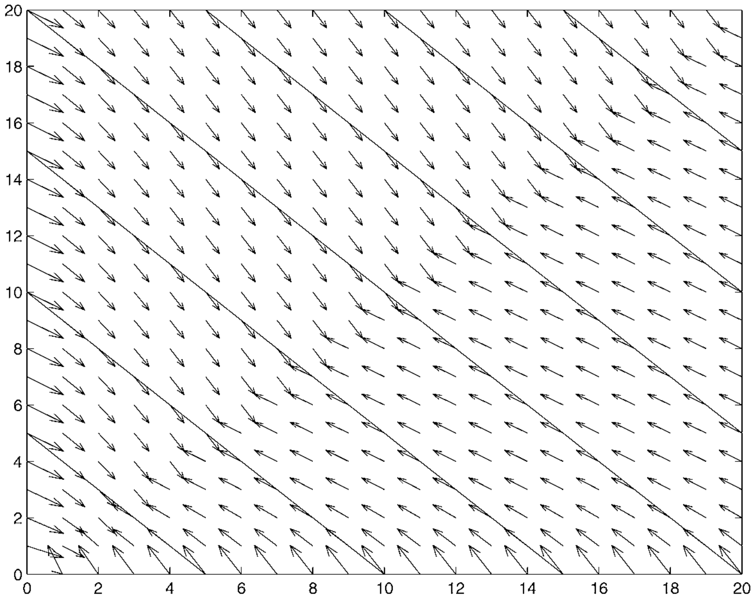


Fig. 3. The workload drift.

boundary, we should wait for a few transitions to see negative drift. The appropriate concept here is "state-dependent drift". A state-dependent version of the SLC (Meyn and Tweedie, 1993, Theorem 19.1.2) is also sufficient to ensure stability, and can be stated as follows.

The Third Lyapunov Condition (TLC). There exists a nonempty $B \subseteq \mathcal{S}$, positive scalars ε, b and δ, an integer $m \geqslant 1$, a probability distribution φ on $\mathcal{S}$, and a function $V : \mathcal{S} \to [0, \infty)$ such that

(1) $\mathrm{P}(X_m \in \cdot | X_0 = z) \geqslant \delta\varphi(\cdot)$ for all $z \in B$, and
(2) $\mathrm{E}(V(X_{n(z)})|X_0 = z) \leqslant V(z) - n(z)\varepsilon + bI(z \in B)$ for some integer $n(z) \geqslant 1$, for all $z \in \mathcal{S}$.

We are now in a position to state and prove a stability result. (The proof given here is more involved than I would prefer, and so is deferred to the Appendix. There may be a simpler proof but I could not find it.)

Proposition 15. *Suppose that* $\mathrm{E}Q_1 < 2\mathrm{E}\tau_1$. *Then V satisfies the TLC, and so the two-ambulance model is stable.*

So under a natural condition the two-ambulance model is stable. The proof uses the TLC, which is a state-dependent generalization of the SLC. There is also a state-dependent version of the FLC that allows us to prove that certain expectations are finite (Meyn and Tweedie, 1993, Theorem 19.1.3), but space reasons prevent us from discussing that further.

In this section we have tried to demonstrate the power of Lyapunov function methods in simulation analysis. There is some art involved in finding a Lyapunov function and proving that it works, but the benefits are great from a theoretical point of view. These techniques have been used in, for example, establishing asymptotic properties of gradient estimators (Glynn and L'Ecuyer, 1995), proving that standardized time series procedures are valid in discrete-event simulation (Haas, 1999), analyzing stochastic approximation algorithms (Bhatnagar et al., 2001), showing that certain variance reduction methods are valid in Markov chain simulations (Henderson and Glynn, 2002; Henderson and Meyn, 2003), and establishing moment conditions used to establish consistency of quantile estimators in Markov chain simulations (Henderson and Glynn, 2004).

Acknowledgements

This work was partially supported by National Science Foundation Grants DMI 0230528, DMI 0224884 and DMI 0400287.

Appendix Proof of Proposition 15

We showed above that if $B = \{z\colon w(i) \leqslant K\}$, where $z = (w(i), \beta(i)\colon i = 1, 2)$ and $K > 0$ is arbitrary, then requirement (1) of the TLC was satisfied. It remains to establish requirement (2).

First consider the case where $\mathrm{E}Q_1 < \mathrm{E}\tau_1$. Let $\varepsilon_1 > 0$ be such that $\mathrm{E}Q_1 < \mathrm{E}\tau_1 - \varepsilon_1$. Now, $[x - \tau_1]^+ - x \to -\tau_1$ a.s. as $x \to \infty$. Furthermore, $|[x - \tau_1]^+ - x| = \min(\tau_1, x)^+ \leqslant \tau_1$ and $\mathrm{E}\tau_1 < \infty$. Therefore, by dominated convergence, there exists an $x^* > 0$ such that $\mathrm{E}[x - \tau_1]^+ - x \leqslant -\mathrm{E}\tau_1 + \varepsilon_1$ for $x > x^*$. Take $K = x^*$. Then for $z \notin B$, at least one of $w(1)$ and $w(2)$ exceeds K. Suppose, without loss of generality, that $w(1) > K$. From (29),

$$\begin{aligned}
&\mathrm{E}\bigl[V(Z_1)|Z_0 = z\bigr] - V(z)\\
&\quad = \mathrm{E}Q_1 + \sum_{i=1}^{2}\bigl(\mathrm{E}\bigl[w(i) - \tau_1\bigr]^+ - w(i)\bigr)\\
&\quad \leqslant \mathrm{E}Q_1 + (-\mathrm{E}\tau_1 + \varepsilon_1) + \bigl(\mathrm{E}\bigl[w(2) - \tau_1\bigr]^+ - w(2)\bigr)\\
&\quad \leqslant \mathrm{E}Q_1 - \mathrm{E}\tau_1 + \varepsilon
\end{aligned}$$

which is the required negative drift for requirement (2) of the TLC.

So now suppose that $\mathrm{E}\tau_1 \leqslant \mathrm{E}Q_1 < 2\mathrm{E}\tau_1$. Let $\tau_i' = \tau_i \wedge K_2$ for some $K_2 > 0$ chosen so that $\mathrm{E}\tau_1' < \mathrm{E}Q_1 < 2\mathrm{E}\tau_1'$. Suppose that we replace the interarrival times τ_i by their truncated versions τ_i' for all i. If we show that the system with truncated interarrival times has negative drift, then so does the system with the untruncated interarrival times. So let us now assume that $\tau_i \leqslant K_2 < \infty$ and $\mathrm{E}\tau_1 < \mathrm{E}Q_1 < 2\mathrm{E}\tau_1$ for all i.

Let B be of the form specified above. We will specify the constant $K > K_2$ soon. Fix $z \notin B$ and assume, without loss of generality, that $w(1) > K$. If $w(2) > K_2$ then the one-step drift is exactly $\mathrm{E}Q_1 - 2\mathrm{E}\tau_1 < 0$ as required, since $\tau_1 \leqslant K_2$. So assume that $w(2) \leqslant K_2$, so that it is "near the boundary". The remainder of the proof is essentially a formalization of the following observation about the dynamics of the chain. For K large enough, and as long as the incoming jobs do not require a huge amount of work, all of the incoming work will be assigned to the second ambulance for some time. The workload of the second ambulance will increase, and after a while it will be far enough from the boundary that its mean drift will be $\mathrm{E}Q_1 - \mathrm{E}\tau_1 > 0$ on each step. Meanwhile, the workload of the first ambulance decreases at rate $\mathrm{E}\tau_1$, so that the overall workload decreases once the workload of the second ambulance is large enough. This will then supply the needed negative drift. Our last few definitions will appear somewhat cryptic, but hopefully their choice will make more sense shortly. Select $k \geqslant 1$ so that $k(\mathrm{E}Q_1 - 2\mathrm{E}\tau_1) + C < 0$, where the constant C does not depend on z, and will be specified below. Choose $K_3 > 0$ so that

$$k\bigl[\mathrm{E}Q_1 - \mathrm{E}\tau_1\bigl(1 + \mathrm{P}(Q_1 \leqslant K_3)^k\bigr)\bigr] + C < 0.$$

Finally, choose K large enough that $K - kK_2 > K_2 + kK_3$. We now show that after k transitions, the expected change in workload is negative.

Over the first k transitions, the total inflow of work is $Q_1 + \cdots + Q_k$. Furthermore, the workload of ambulance 1 is so large that it decreases by $\tau_1 + \cdots + \tau_k$. It may also increase if some of the Q_is are very large. Let $\mathcal{E}$ be the event that $Q_i \leqslant K_3$ for all $i = 1, \ldots, k$, and let $\mathcal{E}^c$ denote its complement. The event $\mathcal{E}$ occurs if the first k jobs are all not too large. On the event $\mathcal{E}$, the first k jobs are all assigned to the second ambulance, and so the second ambulance's workload follows a Lindley recursion, as described next.

Let Y_i denote the waiting time in the queue (not counting service) for the ith job in a single-server queue with interarrival times $(\tau_j\colon j \geqslant 1)$ and service times $(Q_j\colon j \geqslant 1)$. Then $Y_1 = [w(2) - \tau_1]^+$ and for $i \geqslant 1$, $Y_{i+1} = [Y_i + Q_i - \tau_{i+1}]^+$. For $i \geqslant 2$ define $S_i = \sum_{j=2}^{i}(Q_{j-1} - \tau_j)$. Then (Asmussen, 2003, p. 94), for $i \geqslant 2$,

$$Y_i = S_i - \min\{-Y_1, S_2, S_3, \ldots, S_i\}.$$

So we can now write

$$\begin{aligned} &V(Z_k) - V(z) \\ &\quad \leqslant -\sum_{i=1}^{k} \tau_i + I\bigl(\mathcal{E}^c\bigr)\sum_{i=1}^{k} Q_i + \bigl(W_k(2) - w(2)\bigr)I(\mathcal{E}). \end{aligned} \tag{30}$$

The last term in (30) can be written as

$$\begin{aligned}
&\big[Y_k + Q_k - w(2)\big] I(\mathcal{E}) \\
&\quad = \big[S_k - \min\{-Y_1, S_2, S_3, \ldots, S_k\} + Q_k - w(2)\big] I(\mathcal{E}) \\
&\quad = I(\mathcal{E}) \sum_{i=1}^{k} Q_i - I(\mathcal{E}) \sum_{i=2}^{k} \tau_i - w(2) I(\mathcal{E}) \\
&\qquad - I(\mathcal{E}) \min\{-Y_1, S_2, S_3, \ldots, S_k\} \\
&\quad \leqslant I(\mathcal{E}) \sum_{i=1}^{k} Q_i - I(\mathcal{E}) \sum_{i=2}^{k} \tau_i - I(\mathcal{E}) \min\{-Y_1, S_2, S_3, \ldots, S_k\}.
\end{aligned} \tag{31}$$

From (30) and (31) we see that

$$\begin{aligned}
&\mathrm{E}\big[V(Z_k) | Z_0 = z\big] - V(z) \\
&\quad \leqslant -k\, \mathrm{E}\tau_1 + k\, \mathrm{E}Q_1 \\
&\qquad - \mathrm{E} I(\mathcal{E}) \sum_{i=2}^{k} \tau_i - \mathrm{E}\big[I(\mathcal{E}) \min\{-Y_1, S_2, S_3, \ldots, S_k\} | Z_0 = z\big] \\
&\quad = k(\mathrm{E}Q_1 - \mathrm{E}\tau_1) - \mathrm{P}(Q_1 \leqslant K_3)^k (k-1)\, \mathrm{E}\tau_1 \\
&\qquad - \mathrm{E}\big[I(\mathcal{E}) \min\{-Y_1, S_2, S_3, \ldots, S_k\} | Z_0 = z\big] \\
&\quad \leqslant k\big(\mathrm{E}Q_1 - \mathrm{E}\tau_1\big(1 + \mathrm{P}(Q_1 \leqslant K_3)^k\big)\big) + \mathrm{E}\tau_1 \\
&\qquad - \mathrm{E}\big[\min\{-Y_1, S_2, S_3, \ldots\} | Z_0 = z\big].
\end{aligned}$$

But $Y_1 \leqslant w(2) \leqslant K_2$, and so

$$\begin{aligned}
&\mathrm{E}\big[V(Z_k) | Z_0 = z\big] - V(z) \\
&\quad \leqslant k\big(\mathrm{E}Q_1 - \mathrm{E}\tau_1\big(1 + \mathrm{P}(Q_1 \leqslant K_3)^k\big)\big) + C < 0,
\end{aligned}$$

where

$$C = \mathrm{E}\tau_1 - \mathrm{E}\min\{-K_2, S_2, S_3, \ldots\}$$

does not depend on z. The constant C is finite since the random walk $S_2, S_3, \ldots$ has positive drift and the increments $Q_i - \tau_{i+1}$ have bounded negative part; see Asmussen (2003, p. 270). We have shown that after k steps the drift is negative, and this establishes Condition 2 as required.

References

Asmussen, S. (2003). *Applied Probability and Queues*, 2nd edition. *Applications of Mathematics: Stochastic Modeling and Applied Probability*, vol. 51. Springer-Verlag, New York.

Avramidis, A.N., Wilson, J.R. (1996). Integrated variance reduction strategies for simulation. *Operations Research* 44, 327–346.

Bhatnagar, S., Fu, M.C., Marcus, S.I., Bhatnagar, S. (2001). Two-timescale algorithms for simulation optimization of hidden Markov models. *IIE Transactions* 33, 245–258.

Billingsley, P. (1986). *Probability and Measure*, 2nd edition. Wiley, New York.

Cheng, R.C.H., Holland, W. (1998). Two-point methods for assessing variability in simulation output. *Journal of Statistical Computation and Simulation* 60, 183–205.

Dieudonné, J.A. (1960). *Foundations of Modern Analysis*. Academic Press, New York.

Fu, M.C., Hu, J.-Q. (1997). *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer, Boston.

Glasserman, P. (1991). *Gradient Estimation via Perturbation Analysis*. Kluwer, Dordrecht, The Netherlands.

Glasserman, P. (1993). Filtered Monte Carlo. *Mathematics of Operations Research* 18, 610–634.

Glynn, P.W. (1995). Some new results on the initial transient problem. In: Alexopoulos, C., Kang, K., Lilegdon, W.R., Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 165–170.

Glynn, P.W., Heidelberger, P. (1990). Bias properties of budget constrained simulations. *Operations Research* 38, 801–814.

Glynn, P.W., Iglehart, D.L. (1990). Simulation output analysis using standardized time series. *Mathematics of Operations Research* 15, 1–16.

Glynn, P.W., L'Ecuyer, P. (1995). Likelihood ratio gradient estimation for stochastic recursions. *Advances in Applied Probability* 27, 1019–1053.

Glynn, P.W., Whitt, W. (1992). The asymptotic validity of sequential stopping rules for stochastic simulations. *The Annals of Applied Probability* 2, 180–198.

Haas, P.J. (1999). On simulation output analysis for generalized semi-Markov processes. *Communications in Statistics: Stochastic Models* 15, 53–80.

Henderson, S.G. (2000). Mathematics for simulation. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 137–146.

Henderson, S.G. (2001). Mathematics for simulation. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 83–94.

Henderson, S.G., Glynn, P.W. (2001). Computing densities for Markov chains via simulation. *Mathematics of Operations Research* 26, 375–400.

Henderson, S.G., Glynn, P.W. (2002). Approximating martingales for variance reduction in Markov process simulation. *Mathematics of Operations Research* 27, 253–271.

Henderson, S.G., Glynn, P.W. (2004). A central limit theorem for empirical quantiles in the Markov chain setting, working paper.

Henderson, S.G., Meyn, S.P. (2003). Variance reduction for simulation in multiclass queueing networks. *IIE Transactions*, in press.

Irizarry, M.A., Kuhl, M.E., Lada, E.K., Subramanian, S., Wilson, J.R. (2003). Analyzing transformation-based simulation metamodels. *IIE Transactions* 35, 271–283.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Meyn, S.P., Tweedie, R.L. (1993). *Markov Chains and Stochastic Stability*. Springer-Verlag, London.

Prakasa Rao, B.L.S. (1983). *Nonparametric Functional Estimation*. Academic Press, Orlando, FL.

Rice, J.A. (1988). *Mathematical Statistics and Data Analysis*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Schruben, L.W. (1983). Confidence interval estimation using standardized time series. *Operations Research* 31, 1090–1108.

Serfling, R.J. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley, New York.

Shedler, G.S. (1987). *Regeneration and Networks of Queues*. Springer-Verlag, New York.

Wand, M.P., Jones, M.C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13003-0

Chapter 3
Uniform Random Number Generation

Pierre L'Ecuyer

Département d'Informatique et de Recherche Opérationnelle, Université de Montréal,
C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H9S 5B8, Canada
E-mail: lecuyer@iro.umontreal.ca
url: http://www.iro.umontreal.ca/~lecuyer

Abstract

This chapter covers the basic design principles and methods for uniform random number generators used in simulation. We also briefly mention the connections between these methods and those used to construct highly-uniform point sets for quasi-Monte Carlo integration. The emphasis is on the methods based on linear recurrences modulo a large integer, or modulo 2. This reflects the fact that their mathematical structure is much better understood than other types of generators, and that most generators used in simulation have that form. We discuss the main requirements for a good generator, theoretical figures of merit for certain classes of linear-type generators, implementation issues, nonlinear generators, and statistical testing.

1 Introduction

A reliable *(pseudo)random number generator* (RNG) is a basic and essential ingredient for any stochastic simulation. The mathematical theory underlying simulation methods is built over the elegant concepts of probability space and random variable. However, since the exact implementation of these concepts on conventional computers seems impossible, random variables and other random objects are *simulated* by *deterministic algorithms*. The aim of these algorithms is to produce sequences of numbers or objects whose behavior is hard to distinguish from that of their "truly random" counterparts, at least for the application of interest. The details of these requirements may differ depending on the context. For the (Monte Carlo) simulation methods discussed in this handbook, the main goal is to reproduce the statistical properties on which these methods are based, so that the estimators of interest behave as expected. For gaming machines and cryptology, observing the sequence of output values for some time should provide no practical advantage for predicting the forthcoming numbers better than by just guessing at random.

Random variate generation for simulation can be decomposed in two steps:

(1) generating imitations of independent and identically distributed (i.i.d.) random variables having the uniform distribution over the interval $(0, 1)$ and
(2) applying transformations to these i.i.d. $U(0, 1)$ random variates to generate (or imitate) random variates and random vectors from arbitrary distributions.

Step (2) is examined in Chapters 4 and 5. This chapter is devoted to algorithms used for Step (1).

In Section 2 we give a definition and the main requirements of a good uniform RNG. In Section 3 we cover RNGs defined by a linear recurrence modulo a large integer m. We examine their lattice structure, quality criteria, and implementation techniques. In Section 4 we provide a similar coverage for RNGs based on linear recurrences modulo 2 and examine the relationships between these two types of constructions. One example is given in each of these two sections. Nonlinear RNGs are briefly mentioned in Section 5. In Section 6 we discuss empirical statistical testing of RNGs. Additional textbooks and tutorial-like references on uniform RNGs include Knuth (1998), L'Ecuyer (1994, 1998), Niederreiter (1992) and Tezuka (1995).

2 Uniform random number generators

2.1 Generators based on a deterministic recurrence

RNGs used for simulation are almost always based on deterministic algorithms that fit the following framework (L'Ecuyer, 1994): an RNG is a structure $(\mathcal{S}, \mu, f, \mathcal{U}, g)$ where $\mathcal{S}$ is a finite set of *states* (the *state space*), μ is a probability distribution on $\mathcal{S}$ used to select the *initial state* (or *seed*) s_0, $f : \mathcal{S} \to \mathcal{S}$ is the *transition function*, $\mathcal{U}$ is the *output space* and $g : \mathcal{S} \to \mathcal{U}$ is the *output function*. In what follows, we assume that $\mathcal{U} = (0, 1)$. The state of the RNG evolves according to the recurrence $s_i = f(s_{i-1})$, for $i \geqslant 1$, and the *output* at step i is $u_i = g(s_i) \in \mathcal{U}$. The output values $u_0, u_1, u_2, \ldots$ are the so-called *random numbers* produced by the RNG.

Because the state space $\mathcal{S}$ is finite, there are necessarily finite integers $l \geqslant 0$ and $j > 0$ such that $s_{l+j} = s_l$. Then, for all $i \geqslant l$, one has $s_{i+j} = s_i$ and $u_{i+j} = u_i$, because both f and g are deterministic. This means that the state and output sequences are eventually periodic. The smallest positive j for which this happens is called the *period length* of the RNG, and is denoted by ρ. When $l = 0$, the sequence is said to be *purely periodic*. Obviously, the period length ρ cannot exceed $|\mathcal{S}|$, the cardinality of the state space. Good RNGs are designed so that their period length ρ is not far from that upper bound. For general recurrences, ρ may depend on the seed s_0, but good RNGs are normally designed so that ρ is the same for all admissible seeds.

In practice, it is important that the output be *strictly* between 0 and 1, because the transformations that generate nonuniform variates sometimes take infinite values when U is 0 or 1. For example, an exponential random variate X with mean 1 is usually generated via $X = -\ln(1 - U)$ and this gives $X = \infty$ when $U = 1$. All good implementations never produce 0 or 1. However, for the mathematical analysis of RNGs, we often assume that the output space is $[0, 1)$ (i.e., 0 is admissible), because this simplifies the analysis considerably without making a significant difference in the mathematical structure of the generator.

2.2 *Quality criteria*

What are the most important quality criteria to be considered when designing an RNG? An extremely *long period* is essential, to make sure that no wrap-around over the cycle can occur. The length of the period must be guaranteed by a mathematical proof. The RNG must also be *efficient* (run fast and use little memory), *repeatable* (able to reproduce exactly the same sequence as many times as we want) and *portable* (work the same way in different software/hardware environments). The availability of efficient *jump-ahead* methods that can quickly compute $s_{i+\nu}$ given s_i, for any large ν and any i, is also very useful, because it permits one to partition the RNG sequence into long disjoint *streams* and *substreams* of random numbers, to create an arbitrary number of *virtual generators* from a single RNG (Law and Kelton, 2000; L'Ecuyer et al., 2002a). These virtual generators can be used on parallel processors or to support different sources of randomness in a large simulation model, for example (see Chapter 7 for further discussion).

It is important to realize, however, that these elementary properties are *far from sufficient*. As a simple illustration, consider an RNG with state space $\mathcal{S} = \{0, \ldots, 2^{10000} - 1\}$, transition function $s_{i+1} = f(s_i) = (s_i + 1) \bmod 2^{10000}$ and $u_i = g(s_i) = s_i/2^{10000}$. This RNG has the huge period length 2^{10000} and enjoys all the nice properties described in the preceding paragraph, but it is certainly *not* imitating "randomness". The analysis outlined in the following paragraphs, although admittedly heuristic, goes a little deeper.

A sequence of real-valued random variables $u_0, u_1, u_2, \ldots$ are i.i.d. $U(0, 1)$ if and only if for all integers $i \geqslant 0$ and $t > 0$, the vector $\mathbf{u}_{i,t} = (u_i, \ldots, u_{i+t-1})$ is uniformly distributed over the t-dimensional unit hypercube $(0, 1)^t$. Of course, this cannot hold for algorithmic RNGs because any vector of t successive values produced by the generator must belong to

$$\Psi_t = \{(u_0, \ldots, u_{t-1}) \mid s_0 \in \mathcal{S}\},$$

which is the *finite set* of all vectors of t successive output values that can be produced, from all possible initial states. We interpret Ψ_t as a *multiset*, which means that the vectors are counted as many times as they appear, and the cardinality of Ψ_t is exactly equal to that of $\mathcal{S}$.

Suppose that the seed s_0 is selected randomly, uniformly over $\mathcal{S}$. Then, the vector $\mathbf{u}_{0,t}$ has the uniform distribution over the finite set Ψ_t. And if the sequence is purely periodic for all s_0, $\mathbf{u}_{i,t} = (u_i, \ldots, u_{i+t-1})$ is also uniformly distributed over Ψ_t for all $i \geqslant 0$. Since the goal is to approximate the uniform distribution over the unit hypercube $(0,1)^t$, it becomes clear that the set Ψ_t should provide a good uniform coverage of this hypercube. In other words, Ψ_t acts as an *approximation* of the theoretical *sample space* $(0,1)^t$ from which the vectors of successive output values are supposed to be drawn randomly. The vectors are generated uniformly over Ψ_t rather than over $(0,1)^t$. To design good-quality RNGs, we must therefore have practical and effective methods for measuring the uniformity of the corresponding sets Ψ_t, even when they have huge cardinalities. It is true that to have a long period, we need a large state space $\mathcal{S}$. However, a much more important reason for requesting a large number of states is to have a larger set Ψ_t, which can be used to obtain a better coverage of the unit hypercube $[0,1)^t$.

More generally, we may also want to measure the uniformity of sets of the form

$$\Psi_I = \left\{(u_{i_1}, \ldots, u_{i_t}) \mid s_0 \in \mathcal{S}\right\},$$

where $I = \{i_1, \ldots, i_t\}$ is a fixed set of nonnegative integers such that $0 \leqslant i_1 < \cdots < i_t$. As a special case, for $I = \{0, \ldots, t-1\}$, we recover $\Psi_t = \Psi_I$.

The uniformity of Ψ_I is typically assessed by measuring the *discrepancy* between the empirical distribution of its points and the uniform distribution over $(0,1)^t$ (Hellekalek and Larcher, 1998; L'Ecuyer and Lemieux, 2002; Niederreiter, 1992). Discrepancy measures are equivalent to goodness-of-fit test statistics for the multivariate uniform distribution. They can be defined in many ways. The choice of a specific definition typically depends on the mathematical structure of the RNG (different measures are used for different types of RNGs), the reason being that we must be able to compute these uniformity measures quickly even when $\mathcal{S}$ is very large. This excludes any method that requires explicit generation of the sequence over its entire period. The selected discrepancy measure is usually computed for each set I in some predefined class $\mathcal{J}$, these values are weighted or normalized by factors that depend on I, and the worst-case (or average) over $\mathcal{J}$ is adopted as a *figure of merit* used to rank RNGs. The choice of $\mathcal{J}$ and of the weights are arbitrary. Typically, $\mathcal{J}$ would contain sets I such that t and $i_t - i_1$ are rather small. Concrete examples of figures of merit are given later on, for specific types of RNGs.

Generating a random s_0 uniformly over $\mathcal{S}$ can be implemented (approximately) by using a physical device. However, for most practical simulation applications and robust RNGs, just picking an arbitrary s_0 would suffice.

2.3 Links with highly-uniform point sets for quasi-Monte Carlo integration

Point sets Ψ_t that are highly uniform in the t-dimensional unit hypercube are also used for purposes other than imitating randomness. Another major application is *quasi-Monte Carlo* (QMC) integration, where the integral of a function f over $[0, 1)^t$ is approximated by the average of f over the full point set Ψ_t (see Chapter 12). Usually, the point set Ψ_t is randomized in a way that (1) each individual point has the uniform distribution over $[0, 1)^t$, so the value of f at that (random) point is an unbiased estimator of the integral, and (2) the high uniformity of the point set as a whole is preserved. Under certain conditions, this reduces the variance of the average (which is an unbiased estimator of the integral) by inducing a negative correlation between the different evaluations of f.

The point sets used for QMC can actually be constructed in pretty much the same way as RNGs. Their uniformity can also be assessed by the same criteria. In fact, many of the criteria mentioned near the end of the previous subsection were originally introduced for QMC point sets. One general class of construction methods simply consists in selecting an RNG based on a recurrence over a small set of states $\mathcal{S}$ and adopting the corresponding point set Ψ_t. We call it a *recurrence-based* point set. In principle, any of the RNG construction methods discussed in the forthcoming sections could be used to define the recurrence. However, since the size of $\mathcal{S}$ must be kept small, all the techniques whose aim is to obtain efficient implementations of long-period generators become irrelevant. So recurrence-based point sets are usually defined via quite simple linear recurrences using modular arithmetic. Two primary examples are the Korobov lattice rules and the recurrences defined via linear feedback shift registers (see Sections 3 and 4). These methods turn out to be special cases of the two main classes of QMC point set construction techniques: *lattice rules* and *digital nets* (see Chapter 12).

2.4 Statistical testing

Good RNGs are designed based on mathematical analysis of their properties, then implemented and submitted to batteries of *empirical statistical tests*. These tests try to detect empirical evidence against the null hypothesis $\mathcal{H}_0$: "the u_i are realizations of i.i.d. U(0, 1) random variables". A test can be defined by any function T that maps a sequence $u_0, u_1, \dots$ in $(0, 1)$ to a real number X, and for which a good approximation is available for the distribution of the random variable X under $\mathcal{H}_0$. For the test to be implementable, X must depend on only a finite (but perhaps random) number of u_i's. Passing many tests may improve one's confidence in the RNG, but never guarantees that the RNG is foolproof for all kinds of simulations.

It is impossible to build an RNG that *passes all statistical tests*. Consider, for example, the class of all tests that examine the first (most significant) b bits of n successive output values, $u_0, \dots, u_{n-1}$, and return a binary value $X \in \{0, 1\}$. Select $\alpha \in (0, 1)$ so that $\alpha 2^{nb}$ is an integer and let $\mathcal{T}_{n,b,\alpha}$ be the tests in this class that return $X = 1$ for *exactly* $\alpha 2^{nb}$ of the 2^{nb} possible output sequences. The sequence is said to *fail* the test when $X = 1$. The set $\mathcal{T}_{n,b,\alpha}$ is the set of all statistical tests of (exact) level α. The number of tests in this set is equal to the number of ways of choosing $\alpha 2^{nb}$ distinct objects among 2^{nb}. The chosen objects are the sequences that fail the test. For any given output sequence, the number of tests in $\mathcal{T}_{n,b,\alpha}$ that return 1 for this sequence is equal to the number of ways of choosing the other $\alpha 2^{nb} - 1$ sequences that also fail the test. This is the number of ways of choosing $\alpha 2^{nb} - 1$ distinct objects among $2^{nb} - 1$. In other words, every output sequence fails exactly the same number of tests! This result, pointed out by Leeb (1995), should not be surprising. Viewed from a different angle, it is a restatement of the well-known fact that under $\mathcal{H}_0$, each of the 2^{nb} possible sequences has the same probability of occurring, so one may argue that none should be considered more random than any other (Knuth, 1998).

This viewpoint leads to a dead end. For statistical testing of RNG sequences to be meaningful, all tests should not be considered on equal footing. So which ones are more important? Any answer is certainly tainted with its share of arbitrariness. However, for large values of n, the number of tests is huge and all but a tiny fraction are too complicated even to be implemented. So we may say that *bad* RNGs are those that fail simple tests, whereas *good* RNGs fail only complicated tests that are hard to find and run. This common-sense compromise has been generally adopted in one way or another.

Experience shows that RNGs with very long periods, good structure of their set Ψ_t, and based on recurrences that are not too simplistic, pass most reasonable tests, whereas RNGs with short periods or bad structures are usually easy to crack by standard statistical tests. For sensitive applications, it is a good idea, whenever possible, to apply statistical tests designed in close relation with the random variable of interest (e.g., based on a *simplification* of the stochastic model being simulated, and for which the theoretical distribution can be computed). Further discussion of statistical testing for RNGs is given in Section 6.

3 Linear recurrences modulo *m*

3.1 The multiple recursive generator

The most widely used class of RNGs is based on the general linear recurrence

$$x_i = (a_1 x_{i-1} + \cdots + a_k x_{i-k}) \bmod m, \tag{1}$$

where m and k are positive integers called the *modulus* and the *order*, and the *coefficients* $a_1, \ldots, a_k$ are in $\mathbb{Z}_m$, interpreted as the set $\{0, \ldots, m-1\}$ on which all operations are performed with reduction modulo m. The *state* at step i is $s_i = \mathbf{x}_i = (x_{i-k+1}, \ldots, x_i)^{\mathsf{T}}$ (where "T" means "transposed"). When m is a prime number, the finite ring $\mathbb{Z}_m$ is a finite field and it is possible to choose the coefficients a_j so that the period length reaches $\rho = m^k - 1$, the largest possible value (Knuth, 1998). This maximal period length is achieved if and only if the characteristic polynomial of the recurrence, $P(z) = z^k - a_1 z^{k-1} - \cdots - a_k$, is a primitive polynomial over $\mathbb{Z}_m$, i.e., if and only if the smallest positive integer ν such that $(z^\nu \bmod P(z)) \bmod m = 1$ (interpreted as a constant polynomial) is $\nu = m^k - 1$. Knuth (1998) explains how to verify this for a given $P(z)$. For $k > 1$, for $P(z)$ to be a primitive polynomial, it is necessary that a_k and at least another coefficient a_j be nonzero. Finding primitive polynomials of this form is generally easy and they yield the simplified recurrence

$$x_n = (a_r x_{n-r} + a_k x_{n-k}) \bmod m. \tag{2}$$

A *multiple recursive generator* (MRG) uses (1) with a large value of m and defines the output as $u_i = x_i/m$. For $k = 1$, this is the classical *linear congruential generator* (LCG), which was the standard in most simulation software products and textbooks until a few years ago, usually with $m = 2^{31} - 1$. These LCGs have too short a period (ρ is at most $m - 1$) and too coarse a structure of their point set Ψ_t to be used as reliable RNGs (see Section 6). They should simply be discarded. On the other hand, small LCGs can be used to construct QMC point sets which are a special case of lattice rules (see Chapter 12).

In practice, the output function of MRGs is modified slightly to make sure that u_i never takes the value 0 or 1; e.g., one may define $u_i = (x_i + 1)/(m + 1)$, or $u_i = x_i/(m + 1)$ if $x_i > 0$ and $u_i = m/(m + 1)$ otherwise. To simplify the theoretical analysis, here we follow the usual convention of assuming that $u_i = x_i/m$ (in which case u_i *does* take the value 0 occasionally).

3.2 The lattice structure

Let $\mathbf{e}_i$ denote the ith unit vector in k dimensions, with a 1 in position i and 0's elsewhere. Denote by $x_{i,0}, x_{i,1}, x_{i,2}, \ldots$ the values of $x_0, x_1, x_2, \ldots$ produced by the recurrence (1) when the initial state $\mathbf{x}_0$ is $\mathbf{e}_i$. An arbitrary initial state $\mathbf{x}_0 = (z_1, \ldots, z_k)^\mathsf{T}$ can be written as the linear combination $z_1\mathbf{e}_1 + \cdots + z_k\mathbf{e}_k$ and the corresponding sequence is a linear combination of the sequences $(x_{i,0}, x_{i,1}, \ldots)$, with reduction of the coordinates modulo m. Conversely, any such linear combination reduced modulo m is a sequence that can be obtained from some initial state $\mathbf{x}_0 \in \mathcal{S} = \mathbb{Z}_m^k$. If we divide everything by m we find that for the MRG, for each $t \geqslant 1$, $\Psi_t = L_t \cap [0, 1)^t$, where

$$L_t = \left\{ \mathbf{v} = \sum_{i=1}^{t} z_i \mathbf{v}_i \;\middle|\; z_i \in \mathbb{Z} \right\}$$

is a t-dimensional *lattice* in $\mathbb{R}^t$, with basis

$$\mathbf{v}_1 = \frac{(1, 0, \ldots, 0, x_{1,k}, \ldots, x_{1,t-1})^\mathsf{T}}{m},$$
$$\vdots$$
$$\mathbf{v}_k = \frac{(0, 0, \ldots, 1, x_{k,k}, \ldots, x_{k,t-1})^\mathsf{T}}{m},$$

$$\mathbf{v}_{k+1} = (0, 0, \ldots, 0, 1, \ldots, 0)^{\mathsf{T}},$$
$$\vdots$$
$$\mathbf{v}_t = (0, 0, \ldots, 0, 0, \ldots, 1)^{\mathsf{T}}.$$

For $t \leqslant k$, L_t contains all vectors whose coordinates are multiples of $1/m$. For $t > k$, it contains a fraction m^{k-t} of those vectors.

This lattice structure implies that the points of Ψ_t are distributed according to a regular pattern, in equidistant parallel hyperplanes. Graphical illustrations of this lattice structure can be found in several papers and books; e.g., Gentle (2003), Knuth (1998), Law and Kelton (2000) and L'Ecuyer (1998). Define the *dual lattice* to L_t as

$$L_t^* = \left\{\mathbf{h} \in \mathbb{R}^t \colon\ \mathbf{h}^{\mathsf{T}}\mathbf{v} \in \mathbb{Z} \text{ for all } \mathbf{v} \in L_t\right\}.$$

Each $\mathbf{h} \in L_t^*$ is a normal vector that defines a family of equidistant parallel hyperplanes, at distance $1/\|\mathbf{h}\|_2$ apart (where $\|\cdot\|_2$ denotes the Euclidean norm), and these hyperplanes cover all the points of L_t unless $\mathbf{h}$ is an integer multiple of some other vector $\mathbf{h}' \in L_t^*$. Therefore, if ℓ_t is the Euclidean length of a shortest nonzero vector $\mathbf{h}$ in L_t^*, then there is a family of hyperplanes at distance $1/\ell_t$ apart that cover all the points of L_t. A small ℓ_t means that there are thick slices of empty space between the hyperplanes and we want to avoid that. A large ℓ_t means a better (more uniform) coverage of the unit hypercube by the point set Ψ_t. Computing the value of $1/\ell_t$ is often called the *spectral test* (Fishman, 1996; Knuth, 1998).

The lattice property holds as well for the point sets Ψ_I formed by values at arbitrary lags defined by a fixed set of indices $I = \{i_1, \ldots, i_t\}$. One has $\Psi_I = L_I \cap [0, 1)^t$ for some lattice L_I, and the largest distance between successive hyperplanes for a family of hyperplanes that cover all the points of L_I is $1/\ell_I$, where ℓ_I is the Euclidean length of a shortest nonzero vector in L_I^*, the dual lattice to L_I.

The lattice L_I and its dual can be constructed as explained in L'Ecuyer and Couture (1997). Finding the shortest nonzero vector in a lattice with basis $\mathbf{v}_1, \ldots, \mathbf{v}_t$ can be formulated as an integer programming problem with a quadratic objective function

$$\text{Minimize } \|\mathbf{v}\|_2^2 = \sum_{i=1}^{t}\sum_{j=1}^{t} z_i \mathbf{v}_i^{\mathsf{T}} \mathbf{v}_j z_j$$

subject to $z_1, \ldots, z_t$ integers and not all zero. This problem can be solved by a branch-and-bound algorithm (Fincke and Pohst, 1985; L'Ecuyer and Couture, 1997; Tezuka, 1995).

For any given dimension t and m^k points per unit of volume, there is an absolute upper bound on the best possible value of ℓ_I (Conway and Sloane, 1999; Knuth, 1998; L'Ecuyer, 1999b). Let $\ell_t^*(m^k)$ denote such an upper bound. To define a figure of merit that takes into account several sets I, in different numbers of dimensions, it is common practice to divide ℓ_I by an upper bound, to obtain a standardized value between 0 and 1, and then take the worst case over a given class $\mathcal{J}$ of sets I. This gives a figure of merit of the form

$$M_{\mathcal{J}} = \min_{I \in \mathcal{J}} \frac{\ell_I}{\ell^*_{|I|}(m^k)}. \tag{3}$$

A value of $M_{\mathcal{J}}$ too close to zero means that L_I has a bad lattice structure for at least one of the selected sets I. We want a value as close to 1 as possible. Computer searches for good MRGs with respect to this criterion have been reported by L'Ecuyer et al. (1993), L'Ecuyer and Andres (1997), L'Ecuyer (1999a), for example. In most cases, $\mathcal{J}$ was simply the sets of the form $I = \{1, \ldots, t\}$ for $t \leqslant t_1$, where t_1 was an arbitrary integer ranging from 8 to 45. L'Ecuyer and Lemieux (2000) also consider the small-dimensional sets I with indices not too far apart. They suggest taking $\mathcal{J} = \{\{0, 1, \ldots, i\}\colon\ i < t_1\} \cup \{\{i_1, i_2\}\colon\ 0 = i_1 < i_2 < t_2\} \cup \cdots \cup \{\{i_1, \ldots, i_d\}\colon\ 0 = i_1 < \cdots < i_d < t_d\}$ for some positive integers $d, t_1, \ldots, t_d$. We could also take a weighted average instead of the minimum in the definition of $M_{\mathcal{J}}$.

An important observation is that for $t > k$, the t-dimensional vector $\mathbf{h} = (-a_k, \ldots, -a_1, 1, 0, \ldots, 0)^{\mathsf{T}}$ always belongs to L_t^*, because for any vector $\mathbf{v} \in L_t$, the first $k + 1$ coordinates of $m\mathbf{v}$ must satisfy the recurrence (1), which implies that $(-a_k, \ldots, -a_1, 1, 0, \ldots, 0)\mathbf{v}$ must be an integer. Therefore, one always has $\ell_t^2 \leqslant \|\mathbf{h}\|_2^2 = 1 + a_1^2 + \cdots + a_k^2$. Likewise, if I contains 0 and all indices j such that $a_{k-j} \neq 0$, then $\ell_I^2 \leqslant 1 + a_1^2 + \cdots + a_k^2$ (L'Ecuyer, 1997). This means that the sum of squares of the coefficients a_j *must be large* if we want to have any chance that the lattice structure be good.

Constructing MRGs with only two nonzero coefficients and taking these coefficients small has been a very popular idea, because this makes the implementation easier and faster (Deng and Lin, 2000; Knuth, 1998). However, the MRGs thus obtained have a bad structure. As a worst-case illustration, consider the widely-available additive or subtractive *lagged-Fibonacci* generator, based on the recurrence (1) where the two coefficients a_r and a_k are both equal to ±1. In this case, whenever I contains $\{0, k - r, k\}$, one has $\ell_I^2 \leqslant 3$, so the distance between the hyperplanes is at least $1/\sqrt{3}$. In particular, for $I = \{0, k - r, k\}$, all the points of Ψ_I (aside from the zero vector) are contained in only two planes! This type of structure can have a dramatic effect on certain simulation problems and is a good reason for staying away from these lagged-Fibonacci generators, regardless of their parameters.

A similar problem occurs for the "fast MRG" proposed by Deng and Lin (2000), based on the recurrence

$$x_i = (-x_{i-1} + ax_{i-k}) \bmod m = \big((m - 1)x_{i-1} + ax_{i-k}\big) \bmod m,$$

with $a^2 < m$. If a is small, the bound $\ell_I^2 \leqslant 1 + a^2$ implies a bad lattice structure for $I = \{0, k - 1, k\}$. A more detailed analysis by L'Ecuyer and Touzin (2004) shows that this type of generator cannot have a good lattice structure even if the condition $a^2 < m$ is removed. Another special case proposed by Deng and Xu (2003) has the form

$$x_i = a(x_{i-j_2} + \cdots + x_{i-j_t}) \bmod m. \tag{4}$$

In this case, for $I = \{0, k - j_{t-1}, \ldots, k - j_2, k\}$, the vectors $(1, a, \ldots, a)$ and $(a^*, 1, \ldots, 1)$ both belong to the dual lattice L_I^*, where a^* is the multiplicative inverse of a modulo m. So neither a nor a^* should be small.

To get around this structural problem when I contains certain sets of indices, Lüscher (1994) and Knuth (1998) recommend skipping some of the output values to break up the bad vectors. For the lagged-Fibonacci generator, for example, one can output k successive values produced by the recurrence, then skip the next d values, output the next k, skip the next d, and so on. A large value of d (e.g., $d = 5k$ or more) may get rid of the bad structure, but slows down the generator. See Wegenkittl and Matsumoto (1999) for further discussion.

We saw that the point set Ψ_t of an LCG or MRG is the intersection of some special lattice L_t with the unit hypercube, where L_t contains all corners of the hypercube. A *lattice rule* is a QMC integration method defined by selecting an arbitrary lattice L_t with this property, and using its intersection with the unit hypercube as a QMC point set. The uniformity of lattice rules can be measured by the spectral test in the same way as MRGs (see Chapter 12).

3.3 MRG implementation techniques

The modulus m is often taken as a large prime number close to the largest integer directly representable on the computer (e.g., equal or near $2^{31} - 1$ for 32-bit computers). Since each x_{i-j} can be as large as $m-1$, one must be careful in computing the right-hand side of (1) because the product $a_j x_{i-j}$ is typically not representable as an ordinary integer. Various techniques for computing this product modulo m are discussed and compared by Fishman (1996), L'Ecuyer and Côté (1991), L'Ecuyer (1999a) and L'Ecuyer and Simard (1999). Note that if $a_j = m - a'_j > 0$, using a_j is equivalent to using the negative coefficient $-a'_j$, which is sometimes more convenient from the implementation viewpoint. In what follows, we assume that a_j can be either positive or negative.

One approach is to perform the arithmetic modulo m in 64-bit (double precision) floating-point arithmetic (L'Ecuyer, 1999a). Under this representation, assuming that the usual IEEE floating-point standard is respected, all positive integers up to 2^{53} are represented exactly. Then, if each coefficient a_j is selected to satisfy $|a_j|(m-1) \leqslant 2^{53}$, the product $|a_j|x_{i-j}$ will always be represented exactly and $z_j = |a_j|x_{i-j} \bmod m$ can be computed by the instructions

$$y = |a_j|x_{i-j}, \qquad z_j = y - m\left\lfloor \frac{y}{m} \right\rfloor.$$

Similarly, if $(|a_1| + \cdots + |a_k|)(m-1) \leqslant 2^{53}$, $a_1 x_{i-1} + \cdots + a_k x_{i-k}$ will always be represented exactly.

A second technique, called *approximate factoring* (L'Ecuyer and Côté, 1991), uses only the integer representation and works under the condition that $|a_j| = i$ or $|a_j| = \lfloor m/i \rfloor$ for some integer $i < \sqrt{m}$. One precomputes $q_j = \lfloor m/|a_j| \rfloor$ and $r_j = m \bmod |a_j|$. Then, $z_j = |a_j| x_{i-j} \bmod m$ can be computed by

$$y = \left\lfloor \frac{x_{i-j}}{q_j} \right\rfloor, \qquad z = |a_j|(x_{i-j} - yq_j) - yr_j,$$
$$\text{if } z < 0 \text{ then } z_j = z + m \text{ else } z_j = z.$$

All quantities involved in these computations are integers between $-m$ and m, so no overflow can occur if m can be represented as an ordinary integer (e.g., $m < 2^{31}$ on a 32-bit computer).

The *powers-of-two decomposition* approach selects coefficients a_j that can be written as a sum or difference of a small number of powers of 2 (L'Ecuyer and Simard, 1999; L'Ecuyer and Touzin, 2000; Wu, 1997). For example, one may take $a_j = \pm 2^q \pm 2^r$ and $m = 2^e - h$ for some positive integers q, r, e and h. To compute $y = 2^q x \bmod m$, decompose $x = z_0 + 2^{e-q} z_1$ (where $z_0 = x \bmod 2^{e-q}$) and observe that

$$y = 2^q\left(z_0 + 2^{e-q} z_1\right) \bmod\left(2^e - h\right) = \left(2^q z_0 + h z_1\right) \bmod\left(2^e - h\right).$$

Suppose now that

$$h < 2^q \quad \text{and} \quad h\left(2^q - (h+1)2^{-e+q}\right) < m. \tag{5}$$

Then $2^q z_0 < m$ and $hz_1 < m$, so y can be computed by shifts, masks, additions, subtractions, and a single multiplication by h. Intermediate results never exceed $2m - 1$. Things simplify further if $q = 0$ or $q = 1$ or $h = 1$. For $h = 1$, y is obtained simply by swapping the blocks of bits z_0 and z_1 (Wu, 1997). It has been pointed out by L'Ecuyer and Simard (1999) that LCGs with parameters of the form $m = 2^e - 1$ and $a = \pm 2^q \pm 2^r$ have bad statistical properties because the recurrence does not "mix the bits" well enough. However, good and fast MRGs can be obtained via the power-of-two decomposition method, as explained in L'Ecuyer and Touzin (2000).

Another interesting idea for improving efficiency is to take all nonzero coefficients a_j equal to the same constant a (Deng and Xu, 2003; Marsaglia, 1996). Then, computing the right-hand side of (1) requires a single multiplication. Deng and Xu (2003) provide specific parameter sets and concrete implementations for MRGs of this type, for prime m near 2^{31}, and $k = 102, 120$ and 1511.

One may be tempted to take m equal to a power of two, say $m = 2^e$, because computing the products and sums modulo m is then much easier: it suffices to keep the e least significant bits of the results. However, taking a power-of-two modulus has very important disadvantages in terms of the quality of the RNG (L'Ecuyer, 1990, 1998). In particular, the least significant bits have very short periodicity and the period length of the recurrence (1) cannot exceed $(2^k - 1)2^{e-1}$ if $k > 1$, and 2^{e-2} if $k = 1$ and $e \geqslant 4$. The maximal period length achievable with $k = 7$ and $m = 2^{31}$, for example, is more than 2^{180} times smaller than the maximal period length achievable with $k = 7$ and $m = 2^{31} - 1$ (a prime number).

3.4 Combined MRGs and LCGs

The conditions that make MRG implementations run faster (e.g., only two nonzero coefficients both close to zero) are generally in conflict with those required for having a good lattice structure and statistical robustness. *Combined MRGs* provide one solution to this problem. Consider J distinct MRGs evolving in parallel, based on the recurrences

$$x_{j,i} = (a_{j,1}x_{j,i-1} + \cdots + a_{j,k}x_{j,i-k}) \bmod m_j \tag{6}$$

where $a_{j,k} \neq 0$, for $j = 1, \ldots, J$. Let $\delta_1, \ldots, \delta_J$ be arbitrary integers,

$$z_i = (\delta_1 x_{1,i} + \cdots + \delta_J x_{J,i}) \bmod m_1, \qquad u_i = \frac{z_i}{m_1}, \tag{7}$$

and

$$w_i = \left(\frac{\delta_1 x_{1,i}}{m_1} + \cdots + \frac{\delta_J x_{J,i}}{m_J}\right) \bmod 1. \tag{8}$$

This defines two RNGs, with output sequences $\{u_i, i \geqslant 0\}$ and $\{w_i, i \geqslant 0\}$.

Suppose that the m_j are pairwise relatively prime, that δ_j and m_j have no common factor for each j, and that each recurrence (6) is purely periodic with period length ρ_j. Let $m = m_1 \cdots m_J$ and let ρ be the least common multiple of $\rho_1, \ldots, \rho_J$. Under these conditions, the following results have been proved by L'Ecuyer and Tezuka (1991) and L'Ecuyer (1996a): (a) the sequence (8) is exactly equivalent to the output sequence of an MRG with (composite) modulus m and coefficients a_j that can be computed explicitly as explained in L'Ecuyer (1996a); (b) the two sequences in (7) and (8) have period length ρ; and (c) if both sequences have the same initial state, then $u_i = w_i + \varepsilon_i$ where $\max_{i \geqslant 0} |\varepsilon_i|$ can be bounded explicitly by a constant ε which is very small when the m_j are close to each other.

Thus, these combined MRGs can be viewed as practical ways of implementing an MRG with a large m and several large nonzero coefficients. The idea is to cleverly select the components so that: (1) each one is easy to implement efficiently (e.g., has only two small nonzero coefficients) and (2) the MRG that corresponds to the combination has a good lattice structure. If each m_j is prime and if each component j has maximal period length $\rho_j = m_j^k - 1$, then each ρ_j is even and ρ cannot exceed $\rho_1 \cdots \rho_J / 2^{J-1}$. Tables of good parameters for combined MRGs of different sizes that reach this upper bound are given in L'Ecuyer (1999a) and L'Ecuyer and Touzin (2000), together with C implementations.

3.5 Jumping ahead

The recurrence (1) can be written in matrix form as

$$\mathbf{x}_i = \mathbf{A}\mathbf{x}_{i-1} \bmod m = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ a_k & a_{k-1} & \cdots & a_1 \end{pmatrix} \mathbf{x}_{i-1} \bmod m.$$

To jump ahead directly from $\mathbf{x}_i$ to $\mathbf{x}_{i+\nu}$, for an arbitrary integer ν, it suffices to exploit the relationship

$$\mathbf{x}_{i+\nu} = \mathbf{A}^\nu \mathbf{x}_i \bmod m = \left(\mathbf{A}^\nu \bmod m\right)\mathbf{x}_i \bmod m.$$

If this is to be done several times for the same ν, the matrix $\mathbf{A}^\nu \bmod m$ can be precomputed once for all. For a large ν, this can be done in $\mathrm{O}(\log_2 \nu)$ matrix multiplications via a standard divide-and-conquer algorithm (Knuth, 1998):

$$\mathbf{A}^\nu \bmod m = \begin{cases} \left(\mathbf{A}^{\nu/2} \bmod m\right)\left(\mathbf{A}^{\nu/2} \bmod m\right) \bmod m & \text{if } \nu \text{ is even,} \\ \mathbf{A}\left(\mathbf{A}^{\nu-1} \bmod m\right) \bmod m & \text{if } \nu \text{ is odd.} \end{cases}$$

3.6 Linear recurrences with carry

The basic idea here is to add a *carry* to the linear recurrence (1). The general form of this RNG, called *multiply-with-carry* (MWC), can be written as

$$x_i = (a_1 x_{i-1} + \cdots + a_k x_{i-k} + c_{i-1}) d \bmod b, \tag{9}$$

$$c_i = \left\lfloor \frac{a_0 x_i + a_1 x_{i-1} + \cdots + a_k x_{i-k} + c_{i-1}}{b} \right\rfloor, \tag{10}$$

$$u_i = \sum_{\ell=1}^{\infty} x_{i-\ell+1} b^{-\ell}, \tag{11}$$

where b is a positive integer (e.g., a power of two), $a_0, \ldots, a_k$ are arbitrary integers such that a_0 is relatively prime to b, and d is the multiplicative inverse of $-a_0$ modulo b. The state at step i is $s_i = (x_{i-k+1}, \ldots, x_i, c_i)^\mathsf{T}$. In practice, the sum in (11) is truncated to a few terms (it could be a single term if b is large), but the theoretical analysis is much easier for the infinite sum. These types of recurrences were introduced by Marsaglia and Zaman (1991) to obtain a large period even when m is a power of two (this may allow a faster implementation). They were studied and generalized by Couture and L'Ecuyer (1994, 1997), Goresky and Klapper (2003) and Tezuka et al. (1994).

Define $m = \sum_{\ell=0}^{k} a_\ell b^\ell$ and let a be the inverse of b in arithmetic modulo m, assuming for now that $m > 0$. A major result proved in Couture and L'Ecuyer (1997), Goresky and Klapper (2003) and Tezuka et al. (1994) is that if the initial states agree, the output sequence $\{u_i, i \geqslant 0\}$ is exactly the same as that produced by the LCG with modulus m and multiplier a. Therefore, the MWC can be seen as a clever way of implementing an LCG with very large modulus. It has been shown by Couture and L'Ecuyer (1997) that the value of ℓ_t for this LCG satisfies $\ell_t^2 \leqslant a_0^2 + \cdots + a_k^2$ for $t \geqslant k$, which means that the lattice structure will be bad unless the sum of squares of coefficients a_j is large.

In the original proposals of Marsaglia and Zaman (1991), called *add-with-carry* and *subtract-with-borrow*, one has $-a_0 = \pm a_r = \pm a_k = 1$ for some $r < k$ and the other coefficients a_j are zero, so $\ell_t^2 \leqslant 3$ for $t \geqslant k$ and the generator has essentially the same structural defect as the additive lagged-Fibonacci generator. In the version studied by Couture and L'Ecuyer (1997), it was assumed that $-a_0 = d = 1$. Then, the period length cannot exceed $(m-1)/2$ if b is a power of two. A concrete implementation was given in that paper. Goresky and Klapper (2003) pointed out that the maximal period length of $\rho = m - 1$ can be achieved by allowing a more general a_0. They provided specific parameters that give a maximal period for b ranging from 2^{21} to 2^{35} and ρ up to approximately 2^{2521}.

3.7 Computer searches for good parameters and an example

When searching for specific instances of MRGs with good parameters, one would usually impose constraints on the parameters k, m and a_j's (for each component in the case of a combined generator). These constraints are based on implementation efficiency considerations. One of the constraints might be that the MRG (or each component) has maximal period length. The constraints determine a set of feasible solutions in parameter space. A figure of merit measuring the uniformity of the MRG point set, such as $M_{\mathcal{J}}$ in (3) for some set $\mathcal{J}$, is also selected. Then, an "intelligent" random search algorithm (that usually employs several heuristics) is used to find a feasible solution with the largest possible figure of merit. Such computer searches can take days of CPU time, because checking for maximal period and computing the figure of merit is generally very demanding computationally. Nevertheless, unless the constraints are too restrictive, it is typically easy to find good parameter sets by random search, because there is usually a large number of nearly optimal solutions.

As a concrete illustration, consider the combined generator `MRG32k3a` proposed in L'Ecuyer (1999a). It has $J = 2$ components of order $k = 3$ defined as in (6), with $m_1 = 2^{32} - 209$, $a_{11} = 0$, $a_{12} = 1403580$, $a_{13} = -810728$, $m_2 = 2^{32} - 22853$, $a_{21} = 527612$, $a_{22} = 0$, $a_{23} = -1370589$. The combination is defined by $z_i = (x_{1,i} - x_{2,i}) \bmod m_1$ and the MRG that corresponds to this combination has order $k = 3$, modulus $m = m_1 m_2 = 18446645023178547541$ and multipliers $a_1 = 18169668471252892557$, $a_2 = 3186860506199273833$ and $a_3 = 8738613264398222622$. Its period length is $(m_1^3 - 1)(m_2^3 - 1)/2 \approx 2^{191}$.

This generator was found by a computer search with a computing budget of a few days of CPU time, as follows. The values of J, k, m_1 and m_2 were fixed. These values of m_1 and m_2 have special properties explained in L'Ecuyer

(1999a), which make the maximal period length conditions easier to verify. The constraints $a_{11} = a_{22} = 0$ and $(|a_{j,0}| + |a_{j,1}| + |a_{j,2}|)(m_j - 1) < 2^{53}$ were also imposed to make sure that the recurrence of each component was easily implementable in floating-point arithmetic. The figure of merit (to maximize) was $M_{\mathcal{J}}$ with $\mathcal{J} = \{\{0, \dots, i\}\colon\ i < 32\}$ (i.e., the worst-case standardized spectral test value in up to 32 dimensions). The retained generator (given above) has $M_{\mathcal{J}} = 0.6336$. We later verified that $M_{\mathcal{J}} = 0.6225$ if we go up to 45 dimensions instead of 32. Computer codes that implement this particular generator in several languages (such as C, C++, Java) are available from the author's web page. It is also implemented in many commercial simulation packages such as Arena, Automod, Witness, etc.

4 Generators based on recurrences modulo 2

4.1 A general framework

It is certainly a good idea to exploit the fact that computers work in binary arithmetic by designing RNGs defined directly in terms of bit strings and sequences. This is the idea underlying the following framework. Let $\mathbb{F}_2$ denote the finite field with two elements, 0 and 1, in which the operations are equivalent to addition and multiplication modulo 2. Consider the RNG defined by the following matrix linear recurrence over $\mathbb{F}_2$:

$$\mathbf{x}_i = \mathbf{A}\mathbf{x}_{i-1}, \tag{12}$$

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i, \tag{13}$$

$$u_i = \sum_{\ell=1}^{w} y_{i,\ell-1} 2^{-\ell} = y_{i,0} y_{i,1} y_{i,2} \cdots, \tag{14}$$

where $\mathbf{x}_i = (x_{i,0}, \dots, x_{i,k-1})^{\mathsf{T}} \in \mathbb{F}_2^k$ is the k-bit *state vector* at step i, $\mathbf{y}_i = (y_{i,0}, \dots, y_{i,w-1})^{\mathsf{T}} \in \mathbb{F}_2^w$ is the w-bit *output vector* at step i, k and w are positive integers, $\mathbf{A}$ is a $k \times k$ *transition matrix* with elements in $\mathbb{F}_2$, $\mathbf{B}$ is a $w \times k$ *output transformation matrix* with elements in $\mathbb{F}_2$, and $u_i \in [0, 1)$ is the *output* at step i. All operations in (12) and (13) are performed in $\mathbb{F}_2$.

Let

$$P(z) = \det(\mathbf{A} - z\mathbf{I}) = z^k - \alpha_1 z^{k-1} - \cdots - \alpha_{k-1} z - \alpha_k$$

be the characteristic polynomial of $\mathbf{A}$, where $\mathbf{I}$ is the identity matrix and each α_j is in $\mathbb{F}_2$. For any sequence of $\mathbf{x}_i$'s that satisfies (12), for each j, the sequence $\{x_{i,j},\ i \geqslant 0\}$ obeys the linear recurrence

$$x_{i,j} = (\alpha_1 x_{i-1,j} + \cdots + \alpha_k x_{i-k,j}) \bmod 2 \tag{15}$$

(L'Ecuyer, 1994; Niederreiter, 1992). The sequences $\{y_{i,j},\ i \geqslant 0\}$, for $0 \leqslant j < w$, also obey the same recurrence (although they may also follow recurrences of shorter order in certain situations, depending on **B**). We assume that $\alpha_k = 1$, so that the recurrence (15) has *order* k and is purely periodic. Its period length is $2^k - 1$ (i.e., maximal) if and only if $P(z)$ is a primitive polynomial over $\mathbb{F}_2$ (Knuth, 1998; Niederreiter, 1992).

To jump ahead directly from $\mathbf{x}_i$ to $\mathbf{x}_{i+\nu}$ with this type of generator, it suffices to precompute the matrix $\mathbf{A}^\nu$ (in $\mathbb{F}_2$) and then multiply $\mathbf{x}_i$ by this matrix. However, this multiplication could becomes unacceptably time consuming when k exceeds a few hundreds.

Several popular classes of RNGs fit this framework as special cases, by appropriate choices of the matrices **A** and **B**. This includes the Tausworthe or LFSR, polynomial LCG, GFSR, twisted GFSR, Mersenne twister, multiple recursive matrix generators, and combinations of these (L'Ecuyer and Panneton, 2002; Matsumoto and Nishimura, 1998; Niederreiter, 1995; Tezuka, 1995). We detail some of them after discussing how to measure their uniformity. The point sets Ψ_t produced by these RNGs generally contain 2^k points and turn out to be instances of *digital nets*, which form a large class of construction methods for QMC point sets (Chapter 12). This means that any of these RNG implementation methods can be employed to construct recurrence-based QMC point sets, by taking a small value of k.

4.2 Measures of equidistribution

The uniformity of point sets Ψ_I produced by RNGs based on linear recurrences over $\mathbb{F}_2$ is usually assessed by measures of equidistribution defined as follows (L'Ecuyer, 1996b, 2004a; L'Ecuyer and Panneton, 2002; Tezuka, 1995). For an arbitrary vector $\mathbf{q} = (q_1, \dots, q_t)$ of nonnegative integers, partition the unit hypercube $[0, 1)^t$ into 2^{q_j} intervals of the same length along axis j, for each j. This determines a partition of $[0, 1)^t$ into $2^{q_1+\cdots+q_t}$ rectangular boxes of the same size and shape. We call this partition the **q**-equidissection of the unit hypercube.

For some index set $I = \{i_1, \dots, i_t\}$, if Ψ_I has 2^k points, we say that Ψ_I is **q**-*equidistributed in base* 2 if there are exactly 2^q points in each box of the **q**-equidissection, where $k - q = q_1 + \cdots + q_t$. This means that among the 2^k points $(x_{j_1}, \dots, x_{j_t})$ of Ψ_I, if we consider the first q_1 bits of x_{j_1}, the first q_2 bits of x_{j_2}, ..., and the first q_t bits of x_{j_t}, each of the 2^{k-q} possibilities occurs exactly the same number of times. This is possible only if $q \leqslant k$.

The **q**-equidistribution of Ψ_I depends only on the first q_j bits of x_{i_j} for $1 \leqslant j \leqslant t$, for the points $(x_{i_1}, \dots, x_{i_t})$ that belong to Ψ_I. The vector of these $q_1 + \cdots + q_t = k - q$ bits can always be expressed as a linear function of the k bits of the initial state $\mathbf{x}_0$, i.e., as $\mathbf{M}_\mathbf{q}\mathbf{x}_0$ for some $(k - q) \times k$ binary matrix $\mathbf{M}_\mathbf{q}$, and it is easily seen that Ψ_I is **q**-equidistributed if and only if $\mathbf{M}_\mathbf{q}$ has full rank $k - q$. This provides a simple way of checking equidistribution (Fushimi, 1983; L'Ecuyer, 1996b; Tezuka, 1995).

If Ψ_I is $(\ell, \dots, \ell)$-equidistributed for some $\ell \geqslant 1$, it is called *t-distributed with ℓ bits of accuracy*, or *(t, ℓ)-equidistributed* (L'Ecuyer, 1996b). The largest value of ℓ for which this holds is called the *resolution* of the set Ψ_I and is denoted by ℓ_I. This value has the upper bound $\ell_t^* = \min(\lfloor k/t \rfloor, w)$. The *resolution gap* of Ψ_I is defined as $\delta_I = \ell_t^* - \ell_I$. In the same vein as for MRGs, a worst-case figure of merit can be defined here by

$$\Delta_{\mathcal{J}} = \max_{I \in \mathcal{J}} \delta_I,$$

where $\mathcal{J}$ is a preselected class of index sets I.

The point set Ψ_I is a (q, k, t)*-net in base* 2, often called a (t, m, s)-net in the context of QMC methods, where a different notation is used (Niederreiter, 1992), if it is $(q_1, \ldots, q_t)$-equidistributed in base 2 for all nonnegative integers $q_1, \ldots, q_t$ summing to $k - q$. We call the smallest such q the q*-value* of Ψ_I. The smaller it is, the better. One candidate for a figure of merit could be the q-value of Ψ_t for some large t. This measure is frequently used for QMC point sets, for which k is small (Hellekalek and Larcher, 1998; Niederreiter, 1992). However, when $k - q$ is large, i.e., for long-period generators having good equidistribution, it is extremely difficult to compute because there are too many vectors $\mathbf{q}$ for which equidistribution needs to be checked. In practice, for RNGs, one must settle for figures of merit that involve a smaller number of equidissections.

If $\delta_I = 0$ for all sets I of the form $I = \{0, \ldots, t-1\}$, for $1 \leqslant t \leqslant k$, the RNG is said to be *maximally equidistributed* or *asymptotically random* for the word size w (L'Ecuyer, 1996b; Tezuka, 1995; Tootill et al., 1973). This property ensures perfect equidistribution of all sets Ψ_t, for any partition of the unit hypercube into subcubes of equal sizes, as long as $\ell \leqslant w$ and the number of subcubes does not exceed the number of points in Ψ_t. Large-period maximally equidistributed generators, together with their implementations, can be found in L'Ecuyer (1999c), L'Ecuyer and Panneton (2002) and Panneton and L'Ecuyer (2004), for example.

4.3 Lattice structure in spaces of polynomials and formal series

The RNGs defined via (12)–(14) do not have a lattice structure in the real space like MRGs, but they do have a lattice structure in a space of formal series, as explained in Couture and L'Ecuyer (2000), L'Ecuyer (2004a), Lemieux and L'Ecuyer (2003) and Tezuka (1995). The real space $\mathbb{R}$ is replaced by the space $\mathbb{L}_2$ of formal power series with coefficients in $\mathbb{F}_2$, of the form $\sum_{\ell=\omega}^{\infty} x_\ell z^{-\ell}$ for some integer ω. In that setting, the lattices have the form

$$\mathcal{L}_t = \left\{ \mathbf{v}(z) = \sum_{j=1}^{t} h_j(z)\mathbf{v}_j(z) \text{ such that each } h_j(z) \in \mathbb{F}_2[z] \right\},$$

where $\mathbb{F}_2[z]$ is the ring of polynomials with coefficients in $\mathbb{F}_2$, and the basis vectors $\mathbf{v}_j(z)$ are in $\mathbb{L}_2^t$. The elements of the dual lattice $\mathcal{L}_t^*$ are the vectors $\mathbf{h}(z)$ in $\mathbb{L}_2^t$ whose scalar product with any vector of $\mathcal{L}_t$ is a polynomial in $\mathbb{F}_2[z]$.

In one setting that applies for instance to LFSR generators, we define the mapping $\varphi : \mathbb{L}_2 \to \mathbb{R}$ by

$$\varphi\left(\sum_{\ell=\omega}^{\infty} x_\ell z^{-\ell}\right) = \sum_{\ell=\omega}^{\infty} x_\ell 2^{-\ell}$$

and it turns out that the point set Ψ_t produced by the generator is equal to $\varphi(\mathcal{L}_t) \cap [0,1)^t$ for some lattice $\mathcal{L}_t$. The general case is covered by defining the lattice in a different way (adopting the *resolutionwise* lattice) as explained in Couture and L'Ecuyer (2000). Moreover, the equidistribution properties examined in Section 4.2 can be expressed in terms of lengths of shortest vectors in the dual lattice, with appropriate definitions of the length (or norm). Much of the theory and algorithms developed for lattices in the real space can be adapted to these new types of lattices (Couture and L'Ecuyer, 2000; L'Ecuyer, 2004a; Lemieux and L'Ecuyer, 2003; Panneton, 2004; Tezuka, 1995).

4.4 The LFSR generator

Linear feedback shift register (LFSR) (or *Tausworthe*) generators (L'Ecuyer, 1996b; Tausworthe, 1965; Tezuka, 1995) are a special case of (12)–(14) with $\mathbf{A} = \mathbf{A}_0^s$ (in $\mathbb{F}_2$) for some positive integer s, where

$$\mathbf{A}_0 = \begin{pmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ a_k & a_{k-1} & \dots & a_1 \end{pmatrix}, \tag{16}$$

$a_1, \dots, a_k$ are in $\mathbb{F}_2$, $a_k = 1$, and all blank entries in the matrix are zeros. We take $w \leqslant k$ and the matrix $\mathbf{B}$ contains the first w lines of the $k \times k$ identity matrix. The RNG thus obtained can be defined equivalently by

$$x_i = a_1 x_{i-1} + \cdots + a_k x_{i-k} \bmod 2, \tag{17}$$

$$u_i = \sum_{\ell=1}^{w} x_{is+\ell-1} 2^{-\ell}, \tag{18}$$

where $x_{is+\ell-1} = x_{i,\ell-1}$. Here, $P(z)$ is not the characteristic polynomial of the recurrence (17), but the characteristic polynomial of the matrix $\mathbf{A}_0^s$. The choice of s has an important impact on the quality of the generator. A common special case is when a single a_j is nonzero in addition to a_k; then $P(z)$ is a trinomial and we say that we have a *trinomial-based* LFSR generator. These generators are known to have important statistical deficiencies (Matsumoto and Kurita, 1996; Tezuka, 1995) but they can be used as components of combined RNGs (see Section 4.6).

LFSR generators can be expressed as LCGs in a space of polynomials (L'Ecuyer, 1994; Tezuka and L'Ecuyer, 1991; Tezuka, 1995). With this representation, their lattice structure as discussed in Section 4.3 follows immediately.

4.5 The GFSR and twisted GFSR

Here we take $\mathbf{A}$ as the $pq \times pq$ matrix

$$\mathbf{A} = \begin{pmatrix} & & \mathbf{I}_p & & \mathbf{S} \\ \mathbf{I}_p & & & & \\ & \mathbf{I}_p & & & \\ & & \ddots & & \\ & & & \mathbf{I}_p & \end{pmatrix}$$

for some positive integers p and q, where $\mathbf{I}_p$ is the $p \times p$ identity matrix, $\mathbf{S}$ is a $p \times p$ matrix, and the matrix $\mathbf{I}_p$ on the first line is in columns $(r-1)p+1$ to rp for some positive integer r. Often, $w = p$ and $\mathbf{B}$ contains the first w lines of the $pq \times pq$ identity matrix. If $\mathbf{S}$ is also the identity matrix, the generator thus obtained is the trinomial-based *generalized feedback shift register* (GFSR), for which $\mathbf{x}_i$ is obtained by a bitwise exclusive-or of $\mathbf{x}_{i-r}$ and $\mathbf{x}_{i-q}$. This gives a very fast RNG, but its period length cannot exceed $2^q - 1$, because each bit of $\mathbf{x}_i$ follows the same binary recurrence of order $k = q$, with characteristic polynomial $P(z) = z^q - z^{q-r} - 1$.

More generally, we can define $\mathbf{x}_i$ as the bitwise exclusive-or of $\mathbf{x}_{i-r_1}$, $\mathbf{x}_{i-r_2}, \ldots, \mathbf{x}_{i-r_d}$, where $r_d = q$, so that each bit of $\mathbf{x}_i$ follows a recurrence in $\mathbb{F}_2$ whose characteristic polynomial $P(z)$ has $d+1$ nonzero terms. However, the period length is still bounded by $2^q - 1$, whereas considering the pq-bit state, we should rather expect a period length close to 2^{pq}. This was the main motivation for the *twisted GFSR* (TGFSR) generator. In the original version introduced by Matsumoto and Kurita (1992), $w = p$ and the matrix $\mathbf{S}$ is defined as the transpose of $\mathbf{A}_0$ in (16), with k replaced by p. The characteristic polynomial of $\mathbf{A}$ is then $P(z) = P_S(z^q + z^m)$, where $P_S(z) = z^p - a_p z^{p-1} - \cdots - a_1$ is the characteristic polynomial of $\mathbf{S}$, and its degree is $k = pq$. If the parameters are selected so that $P(z)$ is primitive over $\mathbb{F}_2$, then the TGFSR has period length $2^k - 1$. Tezuka (1994) pointed out important weaknesses of the original TGFSR and Matsumoto and Kurita (1994) proposed an improved version that uses a well-chosen matrix $\mathbf{B}$ whose lines differ from those of the identity. The operations implemented by this matrix are called *tempering* and their purpose is to improve the uniformity of the points produced by the RNG. The *Mersenne twister* (Matsumoto and Nishimura, 1998; Nishimura, 2000) is a variant of the TGFSR where k is slightly less than pq and can be a prime number. A specific instance proposed by Matsumoto and Nishimura (1998) is fast, robust, has the huge period length of $2^{19937} - 1$, and has become quite popular.

In the *multiple recursive matrix method* of Niederreiter (1995), the first row of $p \times p$ matrices in $\mathbf{A}$ contains arbitrary matrices. However, a fast implementation is possible only when these matrices are sparse and have a special structure.

4.6 Combined linear generators over $\mathbb{F}_2$

Many of the best generators based on linear recurrences over $\mathbb{F}_2$ are constructed by combining the output of two or more RNGs having a simple structure. The idea is the same as for MRGs: select simple components that can run fast but such that their combination has a more complicated structure and highly-uniform sets Ψ_I for the sets I considered important.

Consider J distinct recurrences of the form (12)–(13), where the jth recurrence has parameters $(k, w, \mathbf{A}, \mathbf{B}) = (k_j, w, \mathbf{A}_j, \mathbf{B}_j)$ and state $\mathbf{x}_{j,i}$ at step i, for $j = 1, \ldots, J$. The output of the combined generator at step i is defined by

$$\mathbf{y}_i = \mathbf{B}_1\mathbf{x}_{1,i} \oplus \cdots \oplus \mathbf{B}_J\mathbf{x}_{J,i},$$
$$u_i = \sum_{\ell=1}^{w} y_{i,\ell-1}2^{-\ell},$$

where $\oplus$ denotes the bitwise exclusive-or operation. One can show (Tezuka, 1995) that the period length ρ of this combined generator is the least common multiple of the period lengths ρ_j of its components. Moreover, this combined generator is equivalent to the generator (12)–(14) with $k = k_1 + \cdots + k_J$, $\mathbf{A} = \mathrm{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_J)$ and $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_J)$.

With this method, by selecting the parameters carefully, the combination of LFSR generators with characteristic polynomials $P_1(z), \ldots, P_J(z)$ gives yet another LFSR with characteristic polynomial $P(z) = P_1(z) \cdots P_J(z)$ and period length equal to the product of the period lengths of the components (L'Ecuyer, 1996b; Tezuka and L'Ecuyer, 1991; Tezuka, 1995; Wang and Compagner, 1993). Tables and fast implementations of maximally equidistributed combined LFSR generators are given in L'Ecuyer (1999c).

The TGFSR and Mersenne twister generators proposed in Matsumoto and Kurita (1994), Matsumoto and Nishimura (1998), and Nishimura (2000) cannot be maximally equidistributed. L'Ecuyer and Panneton (2002), on the other hand, have constructed concrete examples of maximally equidistributed combined TGFSR generators, with period lengths near 2^{466} and 2^{1250}. These generators have the additional property that the resolution gaps δ_I are zero for a class of small sets I with indices not too far apart.

4.7 An example

Consider the combined generator with $J = 4$ LFSR components whose recurrences (17) have the following characteristic polynomials: $P_1(z) = z^{31} - z^6 - 1$, $P_2(z) = z^{29} - z^2 - 1$, $P_3(z) = z^{28} - z^{13} - 1$, and $P_4(z) = z^{25} - z^3 - 1$, and whose values of s in (18) are $s_1 = 18$, $s_2 = 2$, $s_3 = 7$ and $s_4 = 13$, respectively. The corresponding combined LFSR generator has a characteristic polynomial $P(z) = P_1(z)P_2(z)P_3(z)P_4(z)$ of degree 113, with 58 coefficients equal to 0 and 55 equal to 1, and its period length is

$(2^{31}-1)(2^{29}-1)(2^{28}-1)(2^{25}-1) \approx 2^{113}$. This combined generator is also maximally equidistributed (as defined in Section 4.2). An implementation in C is given in L'Ecuyer (1999c), under the name of `lfsr113`. This generator is faster than `MRG32k3a`: it needs approximately 30 seconds to produce 10^9 (one billion) uniform random numbers on a 2.8 GHz Athlon-based computer, compared to approximately 100 seconds for `MRG32k3a`.

Its parameters were selected as follows. The degrees of the characteristic polynomials $P_j(z)$ were fixed at $k_1 = 31$, $k_2 = 29$, $k_3 = 28$ and $k_4 = 25$, and these polynomials were required to be primitive trinomials of the form $P_j(z) = z^{k_j} - z^{q_j} - 1$ with $0 < 2q_j < k_j$ and with step size s_j satisfying $0 < s_j \leqslant k_j - q_j < k_j \leqslant w = 32$ and $\gcd(s_j, 2^{k_j} - 1) = 1$. Components that satisfy these conditions have maximal period length $2^{k_j} - 1$ and can be implemented efficiently as described in L'Ecuyer (1996b). These values of k_j were selected so that the period lengths $2^{k_j} - 1$ of the components have no common factor, which implies that the period length of the combined generator is their product. An exhaustive search was performed to find all parameter values that satisfy these conditions; there are approximately 3.28 million. Among them, there are 4744 for which the combined generator is maximally equidistributed and also collision-free (which means that when the number of points does not exceed the number of boxes in the equidissection, there is never more than one point in a box). The `lfsr113` generator given above is one of them.

5 Nonlinear RNGs

The linear RNGs discussed so far have point sets Ψ_t with a very regular structure. To get away from this regularity, one can either use a nonlinear transition function f, or keep the transition function linear but use a nonlinear output function g. Several types of nonlinear RNGs have been proposed over the years; see, e.g., Blum et al. (1986), Eichenauer-Herrmann (1995), Eichenauer-Herrmann et al. (1998), Hellekalek and Wegenkittl (2003), Knuth (1998), L'Ecuyer (1994), L'Ecuyer and Granger-Piché (2003) and Niederreiter and Shparlinski (2002). Their nonlinear mappings are defined in various ways by multiplicative inversion in a finite field, quadratic and cubic functions in the finite ring of integers modulo m, and other more complicated transformations. Many of them have output sequences that tend to behave much like $U(0, 1)$ sequences even over their entire period length, in contrast with "good" linear RNGs, whose point sets Ψ_t are much more regular than typical random points. In most cases, on the other hand, their statistical properties can be analyzed only empirically or via asymptotic theoretical results. They are generally slower than the linear ones.

Various ways of *combining* RNGs also give rise to nonlinear RNGs whose output sequence shows less regularity; see, e.g., Fishman (1996), Gentle (2003), Knuth (1998), Law and Kelton (2000), L'Ecuyer (1994) and Marsaglia (1985), and other references given there. This includes *shuffling* the output sequence of one generator using another one (or the same one), alternating between several streams, or just adding them in different ways. It is important to understand that to assess the quality of the combined generator, one must analyze the mathematical structure of the combined generator itself rather than the structure of its components (L'Ecuyer, 1996b, 1996a; L'Ecuyer and Granger-Piché, 2003; Tezuka, 1995). Otherwise, these combination techniques are heuristics which often improve the uniformity (empirically), but can also make it worse.

6 Empirical statistical tests

A statistical test for RNGs can be defined by any random variable X whose distribution under $\mathcal{H}_0$ can be well approximated. When X takes the value x, we define the right and left *p-values* of the test by

$$p_{\mathrm{R}} = \mathrm{P}[X \geqslant x|\mathcal{H}_0] \quad \text{and} \quad p_{\mathrm{L}} = \mathrm{P}[X \leqslant x|\mathcal{H}_0].$$

When testing RNGs, there is no need to prespecify the level of the test. If either of the right or left p-value is extremely close to zero, e.g., less than 10^{-15}, then it is clear that $\mathcal{H}_0$ (and the RNG) must be rejected. When a *suspicious* p-value is obtained, e.g., near 10^{-2} or 10^{-3}, one can just repeat this particular test a few more times, perhaps with a larger sample size. Almost always, things will then clarify.

Statistical tests are defined by partitioning the possible realizations of $(u_0, \dots, u_{\tau-1})$ into a finite number of subsets (where the integer τ can be random or deterministic), computing the probability p_j of each subset j under $\mathcal{H}_0$, and measuring the discrepancy between these probabilities and empirical frequencies from realizations simulated by the RNG.

A simple and natural way of doing that is to take $\tau = t$ (a constant) and cut the interval $[0, 1)$ into d equal segments for some positive integer d, to partition the hypercube $[0, 1)^t$ into $k = d^t$ subcubes of volume $1/k$. We then generate n points $\mathbf{u}_i = (u_{ti}, \dots, u_{ti+t-1}) \in [0, 1)^t$, for $i = 0, \dots, n-1$, and count the number N_j of points falling in subcube j, for $j = 0, \dots, k-1$. Any measure of distance (or divergence) between the numbers N_j and their expectations n/k can define a test statistic X. The tests thus defined are generally called *serial tests* of uniformity (Knuth, 1998; L'Ecuyer et al., 2002b). They can be *sparse* (if $n/k \ll 1$), or *dense* (if $n/k \gg 1$), or something in between. There are also *overlapping* versions, where the points are defined by $\mathbf{u}_i = (u_i, \dots, u_{i+t-1})$ for $i = 0, \dots, n-1$.

For further details, specific instances of serial tests, and other empirical tests commonly applied to RNGs (based, e.g., on close pairs of points among in the space, random walks on the real line or over the integers, the linear complexity of a binary output sequence, the simulation of dice or poker hands, etc.), we refer the reader to (Knuth, 1998; L'Ecuyer and Hellekalek, 1998;

L'Ecuyer and Simard, 2001; L'Ecuyer, 2001; L'Ecuyer et al., 2002b; L'Ecuyer and Simard, 2002; Marsaglia, 1985; Rukhin et al., 2001; Vattulainen et al., 1995).

When testing RNGs, there is no specific alternative hypothesis to $\mathcal{H}_0$. Different tests are needed to detect different types of departures from $\mathcal{H}_0$. *Test suites* for RNGs include a selection of tests, with predetermined parameters and sample sizes. The best known are DIEHARD (Marsaglia, 1996) and the NIST test suite (Rukhin et al., 2001). The library *TestU01* (L'Ecuyer and Simard, 2002) implements a large selection of tests in the C language and provides a variety of test suites, some designed for $\mathrm{U}(0, 1)$ output sequences and others for strings of bits.

7 Conclusion, future work and open issues

The ultimate goal of RNG design is to obtain a fast algorithm or device whose output cannot be distinguished in any way from a realization of an infinite sequence of i.i.d. uniform random variables. This requirement is equivalent to passing all possible statistical tests of uniformity and independence. It seems that this can only be achieved through a physical device based on quantum physics. Using this type of device for simulation has several drawbacks, one of them being that the sequence cannot be reproduced without storing it.

RNGs based on linear recurrences and output transformations, on the other hand, are known to fail statistical tests of linear complexity (for obvious reasons), even when their period length is huge. This seems to have no impact for the great majority of relevant discrete-event simulation applications, but it would nevertheless be good to have efficient alternative nonlinear RNGs that also pass these linear complexity tests. Work in that direction has been initiated in L'Ecuyer and Granger-Piché (2003), for instance. In fact, what is needed is a collection of RNGs having different types of structures, different sizes of their state space, for both 32-bit and 64-bit computers, perhaps some faster and some slower but more robust, and where each RNG can provide multiple streams of random numbers as in L'Ecuyer et al. (2002a) (see also Chapter 7). It should also be easy and simple to replace the pseudorandom numbers by (possibly randomized) quasirandom numbers in a simulation.

Work is currently in progress to develop generators with huge period lengths (e.g., near 2^{20000} or more) as well as faster generators based on linear recurrences modulo 2 and good equidistribution properties. The huge-period generators are not necessarily the way to go because they require a large amount of memory and managing multiple streams involves much more overhead than for the smaller generators. Their huge periods may also hide rather long bad subsequences, due to the fact that the transition function typically modifies only a small part of their state at each step. For example Panneton and L'Ecuyer (2006) have shown that if the Mersenne twister proposed by Matsumoto and Nishimura (1998) is initialized to a state that contains almost only zeros, then the fraction of zeros in the state tends to remain very large for several thousand steps. These issues require further study.

Poor (or plain bad) generators can still be found in popular commercial statistical and simulation software, spreadsheets, etc. Do not trust the default RNGs available in these products: many of them are quite unreliable (L'Ecuyer, 2001). Vendors should be pressured to change this state of affairs. Each year, several new RNGs are proposed in the scientific literature or over the Internet. Many of them are based on very little theoretical analysis. An important task of RNG experts is to study these proposals carefully to shed light on their potential weaknesses. This is an area where negative results are often as important to publish as the positive ones.

Acknowledgements

This text is an adaptation of a chapter entitled "Random Number Generation", written for the *Handbook of Computational Statistics*, published by Springer-Verlag (L'Ecuyer, 2004b). This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Grant No. ODGP0110050, NATEQ–Québec Grant No. 02ER3218, and a Canada Research Chair to the author. François Panneton, Richard Simard, Shane Henderson, Barry Nelson and Christiane Lemieux made helpful comments and corrections on an earlier draft.

References

Blum, L., Blum, M., Schub, M. (1986). A simple unpredictable pseudo-random number generator. *SIAM Journal on Computing* 15 (2), 364–383.

Conway, J.H., Sloane, N.J.A. (1999). *Sphere Packings, Lattices and Groups*, 3rd edition. *Grundlehren der Mathematischen Wissenschaften*, vol. 290. Springer-Verlag, New York.

Couture, R., L'Ecuyer, P. (1994). On the lattice structure of certain linear congruential sequences related to AWC/SWB generators. *Mathematics of Computation* 62 (206), 798–808.

Couture, R., L'Ecuyer, P. (1997). Distribution properties of multiply-with-carry random number generators. *Mathematics of Computation* 66 (218), 591–607.

Couture, R., L'Ecuyer, P. (2000). Lattice computations for random numbers. *Mathematics of Computation* 69 (230), 757–765.

Deng, L.-Y., Lin, D.K.J. (2000). Random number generation for the new century. *The American Statistician* 54 (2), 145–150.

Deng, L.-Y., Xu, H. (2003). A system of high-dimensional, efficient, long-cycle and portable uniform random number generators. *ACM Transactions on Modeling and Computer Simulation* 13 (4), 299–309.

Eichenauer-Herrmann, J. (1995). Pseudorandom number generation by nonlinear methods. *International Statistical Reviews* 63, 247–255.

Eichenauer-Herrmann, J., Herrmann, E., Wegenkittl, S. (1998). A survey of quadratic and inversive congruential pseudorandom numbers. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1996*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 66–97.

Fincke, U., Pohst, M. (1985). Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation* 44, 463–471.

Fishman, G.S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. *Springer Series in Operations Research*. Springer-Verlag, New York.

Fushimi, M. (1983). Increasing the orders of equidistribution of the leading bits of the Tausworthe sequence. *Information Processing Letters* 16, 189–192.

Gentle, J.E. (2003). *Random Number Generation and Monte Carlo Methods*, 2nd edition. Springer-Verlag, New York.

Goresky, M., Klapper, A. (2003). Efficient multiply-with-carry random number generators with maximal period. *ACM Transactions on Modeling and Computer Simulation* 13 (4), 310–321.

Hellekalek, P., Larcher, G. (Eds.) (1998). *Random and Quasi-Random Point Sets*. *Lecture Notes in Statistics*, vol. 138. Springer-Verlag, New York.

Hellekalek, P., Wegenkittl, S. (2003). Empirical evidence concerning AES. *ACM Transactions on Modeling and Computer Simulation* 13 (4), 322–333.

Knuth, D.E. (1998). *Seminumerical Algorithms*, 3rd edition. *The Art of Computer Programming*, vol. 2. Addison-Wesley, Reading, MA.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

L'Ecuyer, P. (1990). Random numbers for simulation. *Communications of the ACM* 33 (10), 85–97.

L'Ecuyer, P. (1994). Uniform random number generation. *Annals of Operations Research* 53, 77–120.

L'Ecuyer, P. (1996a). Combined multiple recursive random number generators. *Operations Research* 44 (5), 816–822.

L'Ecuyer, P. (1996b). Maximally equidistributed combined Tausworthe generators. *Mathematics of Computation* 65 (213), 203–213.

L'Ecuyer, P. (1997). Bad lattice structures for vectors of non-successive values produced by some linear recurrences. *INFORMS Journal on Computing* 9 (1), 57–60.

L'Ecuyer, P. (1998). Random number generation. In: Banks, J. (Ed.), *Principles, Methodology, Advances, Applications, and Practice*. *Handbook of Simulation*. Wiley, New York, pp. 93–137, Chapter 2.

L'Ecuyer, P. (1999a). Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47 (1), 159–164.

L'Ecuyer, P. (1999b). Tables of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation* 68 (225), 249–260.

L'Ecuyer, P. (1999c). Tables of maximally equidistributed combined LFSR generators. *Mathematics of Computation* 68 (225), 261–269.

L'Ecuyer, P. (2001). Software for uniform random number generation: Distinguishing the good and the bad. In: *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 95–105.

L'Ecuyer, P. (2004a). Polynomial integration lattices. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer-Verlag, Berlin, pp. 73–98.

L'Ecuyer, P. (2004b). Random number generation. In: Gentle, J.E., Haerdle, W., Mori, Y. (Eds.), *Concepts and Methods*. *Handbook of Computational Statistics*. Springer-Verlag, Berlin, pp. 35–70, Chapter II.2.

L'Ecuyer, P., Andres, T.H. (1997). A random number generator based on the combination of four LCGs. *Mathematics and Computers in Simulation* 44, 99–107.

L'Ecuyer, P., Côté, S. (1991). Implementing a random number package with splitting facilities. *ACM Transactions on Mathematical Software* 17 (1), 98–111.

L'Ecuyer, P., Couture, R. (1997). An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS Journal on Computing* 9 (2), 206–217.

L'Ecuyer, P., Granger-Piché, J. (2003). Combined generators with components from different families. *Mathematics and Computers in Simulation* 62, 395–404.

L'Ecuyer, P., Hellekalek, P. (1998). Random number generators: Selection criteria and testing. In: Hellekalek, P., Larcher, G. (Eds.), *Random and Quasi-Random Point Sets*. *Lecture Notes in Statistics*, vol. 138. Springer-Verlag, New York, pp. 223–265.

L'Ecuyer, P., Lemieux, C. (2000). Variance reduction via lattice rules. *Management Science* 46 (9), 1214–1235.

L'Ecuyer, P., Lemieux, C. (2002). Recent advances in randomized quasi-Monte Carlo methods. In: Dror, M., L'Ecuyer, P., Szidarovszki, F. (Eds.), *Modeling Uncertainty: An Examination of Stochastic Theory, Methods and Applications*. Kluwer Academic Publishers, Boston, pp. 419–474.

L'Ecuyer, P., Panneton, F. (2002). Construction of equidistributed generators based on linear recurrences modulo 2. In: Fang, K.-T., Hickernell, F.J., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 318–330.

L'Ecuyer, P., Simard, R. (1999). Beware of linear congruential generators with multipliers of the form $a = \pm 2^q \pm 2^r$. *ACM Transactions on Mathematical Software* 25 (3), 367–374.

L'Ecuyer, P., Simard, R. (2001). On the performance of birthday spacings tests for certain families of random number generators. *Mathematics and Computers in Simulation* 55 (1–3), 131–137.

L'Ecuyer, P., Simard, R. (2002). TestU01: A Software Library in ANSI C for Empirical Testing of Random Number Generators. Software user's guide. Available at *http://www.iro.umontreal.ca/~lecuyer*.

L'Ecuyer, P., Tezuka, S. (1991). Structural properties for two classes of combined random number generators. *Mathematics of Computation* 57 (196), 735–746.

L'Ecuyer, P., Touzin, R. (2000). Fast combined multiple recursive generators with multipliers of the form $a = \pm 2^q \pm 2^r$. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 683–689.

L'Ecuyer, P., Touzin, R. (2004). On the Deng–Lin random number generators and related methods. *Statistics and Computing* 14, 5–9.

L'Ecuyer, P., Blouin, F., Couture, R. (1993). A search for good multiple recursive random number generators. *ACM Transactions on Modeling and Computer Simulation* 3 (2), 87–98.

L'Ecuyer, P., Simard, R., Chen, E.J., Kelton, W.D. (2002a). An object-oriented random-number package with many long streams and substreams. *Operations Research* 50 (6), 1073–1075.

L'Ecuyer, P., Simard, R., Wegenkittl, S. (2002b). Sparse serial tests of uniformity for random number generators. *SIAM Journal on Scientific Computing* 24 (2), 652–668.

Leeb, H. (1995). Random numbers for computer simulation. Master's thesis, University of Salzburg.

Lemieux, C., L'Ecuyer, P. (2003). Randomized polynomial lattice rules for multivariate integration and simulation. *SIAM Journal on Scientific Computing* 24 (5), 1768–1789.

Lüscher, M. (1994). A portable high-quality random number generator for lattice field theory simulations. *Computer Physics Communications* 79, 100–110.

Marsaglia, G. (1985). A current view of random number generators. In: *Computer Science and Statistics, Sixteenth Symposium on the Interface*. Elsevier (North-Holland), Amsterdam, pp. 3–10.

Marsaglia, G. (1996). The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness. Available at *http://stat.fsu.edu/pub/diehard*.

Marsaglia, G., Zaman, A. (1991). A new class of random number generators. *The Annals of Applied Probability* 1, 462–480.

Matsumoto, M., Kurita, Y. (1992). Twisted GFSR generators. *ACM Transactions on Modeling and Computer Simulation* 2 (3), 179–194.

Matsumoto, M., Kurita, Y. (1994). Twisted GFSR generators II. *ACM Transactions on Modeling and Computer Simulation* 4 (3), 254–266.

Matsumoto, M., Kurita, Y. (1996). Strong deviations from randomness in m-sequences based on trinomials. *ACM Transactions on Modeling and Computer Simulation* 6 (2), 99–106.

Matsumoto, M., Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8 (1), 3–30.

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. *SIAM CBMS–NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia.

Niederreiter, H. (1995). The multiple-recursive matrix method for pseudorandom number generation. *Finite Fields and Their Applications* 1, 3–30.

Niederreiter, H., Shparlinski, I.E. (2002). Recent advances in the theory of nonlinear pseudorandom number generators. In: Fang, K.-T., Hickernell, F.J., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 86–102.

Nishimura, T. (2000). Tables of 64-bit Mersenne twisters. *ACM Transactions on Modeling and Computer Simulation* 10 (4), 348–357.

Panneton, F. (2004). Construction d'ensembles de points basée sur des récurrences linéaires dans un corps fini de caractéristique 2 pour la simulation Monte Carlo et l'intégration quasi-Monte Carlo. PhD thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada, August 2004.

Panneton, F., L'Ecuyer, P. (2004). Random number generators based on linear recurrences in F_{2^w}. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer-Verlag, Berlin, pp. 367–378.

Panneton, F., L'Ecuyer, P. (2006). Improved long-period generators based on linear recurrences modulo 2. *ACM Transactions on Mathematical Software* 32 (1), in press.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S. (2001). A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST special publication 800-22, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. Available at *http://csrc.nist.gov/rng/*.

Tausworthe, R.C. (1965). Random numbers generated by linear recurrence modulo two. *Mathematics of Computation* 19, 201–209.

Tezuka, S. (1994). A unified view of large-period random number generators. *Journal of the Operations Research Society of Japan* 37, 211–227.

Tezuka, S. (1995). *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, Norwell, MA.

Tezuka, S., L'Ecuyer, P. (1991). Efficient and portable combined Tausworthe random number generators. *ACM Transactions on Modeling and Computer Simulation* 1 (2), 99–112.

Tezuka, S., L'Ecuyer, P., Couture, R. (1994). On the add-with-carry and subtract-with-borrow random number generators. *ACM Transactions on Modeling and Computer Simulation* 3 (4), 315–331.

Tootill, J.P.R., Robinson, W.D., Eagle, D.J. (1973). An asymptotically random Tausworthe sequence. *Journal of the ACM* 20, 469–481.

Vattulainen, I., Ala-Nissila, T., Kankaala, K. (1995). Physical models as tests of randomness. *Physical Review E* 52 (3), 3205–3213.

Wang, D., Compagner, A. (1993). On the use of reducible polynomials as random number generators. *Mathematics of Computation* 60, 363–374.

Wegenkittl, S., Matsumoto, M. (1999). Getting rid of correlations among pseudorandom numbers: Discarding versus tempering. *ACM Transactions on Modeling and Computer Simulation* 9 (3), 282–294.

Wu, P.-C. (1997). Multiplicative, congruential random number generators with multiplier $\pm 2^{k_1} \pm 2^{k_2}$ and modulus $2^p - 1$. *ACM Transactions on Mathematical Software* 23 (2), 255–265.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13004-2

Chapter 4

Nonuniform Random Variate Generation

Luc Devroye

School of Computer Science, McGill University,
3480 University Street, Montreal, H3A 2K6, Canada

Abstract

This chapter provides a survey of the main methods in nonuniform random variate generation, and highlights recent research on the subject. Classical paradigms such as inversion, rejection, guide tables, and transformations are reviewed. We provide information on the expected time complexity of various algorithms, before addressing modern topics such as indirectly specified distributions, random processes, and Markov chain methods.

1 The main paradigms

The purpose of this chapter is to review the main methods for generating random variables, vectors and processes. Classical workhorses such as the inversion method, the rejection method and table methods are reviewed in Section 1. In Section 2 we discuss the expected time complexity of various algorithms, and give a few examples of the design of generators that are uniformly fast over entire families of distributions. In Section 3 we develop a few universal generators, such as generators for all log concave distributions on the real line. Section 4 deals with random variate generation when distributions are indirectly specified, e.g., via Fourier coefficients, characteristic functions, the moments, the moment generating function, distributional identities, infinite series or Kolmogorov measures. Random processes are briefly touched upon in Section 5. Finally, the latest developments in Markov chain methods are discussed in Section 6. Some of this work grew from Devroye (1986a), and we are carefully documenting work that was done since 1986. More recent references can be found in the book by Hörmann et al. (2004).

Nonuniform random variate generation is concerned with the generation of random variables with certain distributions. Such random variables are often discrete, taking values in a countable set, or absolutely continuous, and thus described by a density. The methods used for generating them depend upon the computational model one is working with, and upon the demands on the part of the output.

For example, in a RAM (random access memory) model, one accepts that real numbers can be stored and operated upon (compared, added, multiplied, and so forth) in one time unit. Furthermore, this model assumes that a source capable of producing an i.i.d. (independent identically distributed) sequence of uniform[0, 1] random variables is available. This model is of course unrealistic, but designing random variate generators based on it has several advantages: first of all, it allows one to disconnect the theory of nonuniform random variate generation from that of uniform random variate generation, and secondly, it permits one to plan for the future, as more powerful computers will be developed that permit ever better approximations of the model. Algorithms designed under finite approximation limitations will have to be redesigned when the next generation of computers arrives.

For the generation of discrete or integer-valued random variables, which includes the vast area of the generation of random combinatorial structures, one can adhere to a clean model, the pure bit model, in which each bit operation takes one time unit, and storage can be reported in terms of bits. Typically, one now assumes that an i.i.d. sequence of independent perfect bits is available. In this model, an elegant information-theoretic theory can be derived. For example, Knuth and Yao (1976) showed that to generate a random integer X described by the probability distribution

$$\mathrm{P}\{X = n\} = p_n, \quad n \geqslant 1,$$

any method must use an expected number of bits greater than the binary entropy of the distribution,

$$\sum_n p_n \log_2 \frac{1}{p_n}.$$

They also showed how to construct tree-based generators that can be implemented as finite or infinite automata to come within three bits of this lower bound for any distribution. While this theory is elegant and theoretically important, it is somewhat impractical to have to worry about the individual bits in the binary expansions of the p_n's, so that we will, even for discrete distributions, consider only the RAM model. Noteworthy is that attempts have been made (see, e.g., Flajolet and Saheb, 1986) to extend the pure bit model to obtain approximate algorithms for random variables with densities.

1.1 The inversion method

For a univariate random variable, the inversion method is theoretically applicable: given the distribution function F, and its inverse F^{inv}, we generate a random variate X with that distribution as $F^{\mathrm{inv}}(U)$, where U is a uniform[0, 1] random variable. This is the method of choice when the inverse is readily

Table 1.
Some densities with distribution functions that are explicitly invertible

Name	Density	Distribution function	Random variate
Exponential	$e^{-x}, x > 0$	$1 - e^{-x}$	$\log(1/U)$
Weibull(a), $a > 0$	$ax^{a-1}e^{-x^a}, x > 0$	$1 - e^{-x^a}$	$(\log(1/U))^{1/a}$
Gumbel	$e^{-x}e^{-e^{-x}}$	$e^{-e^{-x}}$	$-\log\log(1/U)$
Logistic	$1/(2 + e^x + e^{-x})$	$1/(1 + e^{-x})$	$-\log((1 - U)/U)$
Cauchy	$1/(\pi(1 + x^2))$	$1/2 + (1/\pi)\arctan x$	$\tan(\pi U)$
Pareto(a), $a > 0$	$a/x^{a+1}, x > 1$	$1 - 1/x^a$	$1/U^{1/a}$

computable. For example, a standard exponential random variable (which has density e^{-x}, $x > 0$), can be generated as $\log(1/U)$. Table 1 gives some further examples.

The fact that there is a monotone relationship between U and X has interesting benefits in the area of coupling and variance reduction. In simulation, one sometimes requires two random variates from the same distribution that are maximally anti-correlated. This can be achieved by generating the pair $(F^{\text{inv}}(U), F^{\text{inv}}(1 - U))$. In coupling, one has two distribution functions F and G and a pair of (possibly dependent) random variates (X, Y) with these marginal distribution functions is needed so that some appropriate metric measuring distance between them is minimized. For example, the Wasserstein metric $d_2(F, G)$ is the minimal value over all couplings of (X, Y) of $\sqrt{E\{(X - Y)^2\}}$. That minimal coupling occurs when $(X, Y) = (F^{\text{inv}}(U), G^{\text{inv}}(U))$ (see, e.g., Rachev, 1991). Finally, if we wish to simulate the maximum M of n i.i.d. random variables, each distributed as $X = F^{\text{inv}}(U)$, then noting that the maximum of n i.i.d. uniform[0, 1] random variables is distributed as $U^{1/n}$, we see that M can be simulated as $F^{\text{inv}}(U^{1/n})$.

1.2 Simple transformations

We call a simple transformation one that uses functions and operators that are routinely available in standard libraries, such as the trigonometric, exponential and logarithmic functions. For example, the inverse of the normal and stable distribution functions cannot be computed using simple transformations of one uniform random variate. For future reference, the standard normal density is given by $\exp(-x^2/2)/\sqrt{2\pi}$. However, the next step is to look at simple transformations of k uniform[0, 1] random variates, where k is either a small fixed integer or a random integer with a small mean. It is remarkable that one can obtain the normal and indeed all stable distributions using simple transformations with $k = 2$. In the Box–Müller method (Box and Müller, 1958) a pair of independent standard normal random variates is obtained by setting

$$(X, Y) = \left(\sqrt{2\log\frac{1}{U_1}}\cos(2\pi U_2), \sqrt{2\log\frac{1}{U_1}}\sin(2\pi U_2)\right),$$

where U_1, U_2 are independent uniform[0, 1] random variates. For the computational perfectionists, we note that the random cosine can be avoided: just generate a random point in the unit circle by rejection from the enclosing square (more about that later), and then normalize it so that it is of unit length. Its first component is distributed as a random cosine.

There are many other examples that involve the use of a random cosine, and for this reason, they are called polar methods. We recall that the beta(a, b) density is

$$\frac{x^{a-1}(1-x)^{b-1}}{B(a, b)}, \quad 0 \leqslant x \leqslant 1,$$

where $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a+b)$. A symmetric beta$(a, a)$ random variate may be generated as (Ulrich, 1984)

$$\frac{1}{2}\left(1 + \sqrt{1 - U_1^{2/(2a-1)}}\cos(2\pi U_2)\right),$$

where $a \geqslant 1/2$. Devroye (1996) provided a recipe valid for all $a > 0$,

$$\frac{1}{2}\left(1 + \frac{S}{\sqrt{1 + 1/((U_1^{-1/a} - 1)\cos^2(2\pi U_2))}}\right),$$

where S is a random sign. Perhaps the most striking result of this kind is due to Bailey (1994), who showed that

$$\sqrt{a\left(U_1^{-2/a} - 1\right)}\cos(2\pi U_2)$$

has the Student t density (invented by William S. Gosset in 1908) with parameter $a > 0$,

$$\frac{1}{\sqrt{a}B(a/2, 1/2)(1 + x^2/a)^{(a+1)/2}}, \quad x \in \mathbb{R}.$$

Until Bailey's paper, only rather inconvenient rejection methods were available for the t density.

There are many random variables that can be represented as $\psi(U)E^{\alpha}$, where ψ is a function, U is uniform[0, 1], α is a real number and E is an independent exponential random variable. These lead to simple algorithms for a host of useful yet tricky distributions. A random variable $S_{\alpha,\beta}$ with characteristic function

$$\varphi(t) = \exp\left(-|t|^{\alpha} - \frac{\mathrm{i}\pi\beta(\alpha - 2\,\mathbf{1}_{\alpha>1})\,\mathrm{sign}(t)}{2}\right)$$

is said to be stable with parameters $\alpha \in (0, 2]$ and $|\beta| \leqslant 1$. Its parameter α determines the size of its tail. Using integral representations of distribution functions, Kanter (1975) showed that for $\alpha < 1$, $S_{\alpha,1}$ is distributed as

$$\psi(U)E^{1-1/\alpha},$$

where

$$\psi(u) = \left(\frac{\sin(\alpha\pi u)}{\sin(\pi u)}\right)^{1/\alpha}\left(\frac{\sin((1-\alpha)\pi u)}{\sin(\alpha\pi u)}\right)^{(1-\alpha)/\alpha}.$$

For general α, β, Chambers et al. (1976) showed that it suffices to generate it as

$$\psi\left(U - \frac{1}{2}\right)E^{1-1/\alpha},$$

where

$$\psi(u) = \left(\frac{\cos(\pi((\alpha-1)u + \alpha\theta)/2)}{\cos(\pi u/2)}\right)^{1/\alpha}\left(\frac{\sin(\pi\alpha(u+\theta)/2)}{\cos(\pi((\alpha-1)u+\alpha\theta)/2)}\right).$$

Zolotarev (1959, 1966, 1981, 1986) has additional representations and a thorough discussion on these families of distributions. The paper by Devroye (1990) contains other examples with $k = 3$, including

$$S_{\alpha,0}E^{1/\alpha},$$

which has the so-called Linnik distribution (Linnik, 1962) with characteristic function

$$\varphi(t) = \frac{1}{1+|t|^{\alpha}}, \quad 0 < \alpha \leqslant 2.$$

See also Kotz and Ostrovskii (1996). It also shows that

$$S_{\alpha,1}E^{1/\alpha}$$

has the Mittag–Leffler distribution with characteristic function

$$\varphi(t) = \frac{1}{1+(-\mathrm{i}t)^{\alpha}}, \quad 0 < \alpha \leqslant 1.$$

Despite these successes, some distributions seem hard to treat with any k-simple transformation. For example, to date, we do not know how to generate a gamma random variate (i.e., a variate with density $x^{a-1}\mathrm{e}^{-x}/\Gamma(a)$ on $(0, \infty)$) with arbitrary parameter $a > 0$ using only a fixed finite number k of uniform random variates and simple transformations.

1.3 Simple methods for multivariate distributions

For random vectors $(X_1, \ldots, X_d)$, the inversion method proceeds by peeling off the dimensions. That is, we first generate X_d from its marginal distribution, then X_{d-1} given X_d, using a conditional distribution, and so forth. There is of course no reason to employ conditioning in that rigid manner. For example, if a distribution has a density that is a function of $\sum_i x_i^2$ only, then it is advantageous to find the marginal of $\sum_i X_i^2$, generate a random variate from this marginal distribution, and then generate a random variate uniformly distributed on the surface of the ball with the given random radius. And to generate a uniformly distributed point on the surface of the unit ball, just generate d i.i.d. standard normal random variates $X_1, \ldots, X_d$, and normalize the length to one.

Sometimes, one wishes to generate random vectors with a certain dependence structure. This can be captured by various measures of correlation. For a survey of random variate generators in this vein, see Devroye (1986a). An interesting new area of research is related to the development of generators with a given copula structure. The copula of two uniform[0, 1] random variables X and Y is given by the joint distribution function

$$C(x, y) = \mathrm{P}\{X \leqslant x, Y \leqslant y\}.$$

We say that two arbitrary continuous random variables X and Y (with distribution functions F and G, respectively) have the copula structure given by C if $F(X)$ and $G(Y)$ (which are both uniform[0, 1]) have joint distribution function C. Thus, we need methods for generating random vectors with distribution function C. Various interesting families of copulas and some seedlings of algorithms, are given in Nelsen's (1999) book and in Chapter 5 of the present volume.

1.4 Inversion for integer-valued random variables

For integer-valued random variables with $\mathrm{P}\{X = n\} = p_n$, $n \geqslant 0$, the inversion method is always applicable:

```
$X \leftarrow 0$
generate $U$ uniform[0, 1]
$S \leftarrow p_0$ ($S$ holds the partial sums of the $p_n$'s)
while $U > S$ do
    $X \leftarrow X + 1, S \leftarrow S + p_X$
return $X$.
```

The expected number of steps here is $\mathrm{E}\{X + 1\}$. Improvements are possible by using data structures that permit one to invert more quickly. When there are only a finite number of values, a binary search tree may help. Here the leaves correspond to various outcomes for X, and the internal nodes are there to guide the search by comparing U with appropriately picked thresholds. If the cost of setting up this tree is warranted, then one could always permute the leaves to make this into a Huffman tree for the weights p_n (Huffman, 1952), which insures that the expected time to find a leaf is not more than one plus the binary entropy,

$$\sum_n p_n \log_2 \frac{1}{p_n}.$$

In any case, this value does not exceed $\log_2 N$, where N is the number of possible values X can take.

Another way of organizing this is to store an array with the partial sums $p_1, p_1 + p_2, \ldots, p_1 + \cdots + p_N = 1$, and to search for the interval in which U falls by binary search, which would take $O(\log_2 N)$ time. As mentioned above, the Huffman tree is optimal, and thus better, but requires more work to set up.

1.5 Guide tables

Hash tables, or guide tables (Chen and Asau, 1974), have been used to accelerate the inversion even further. Given the size of the universe N, we create a hash table with N entries, and store in the ith entry the value of X if U were $i/N, 0 \leqslant i < N$. Then U gets "hashed" to $Z = \lfloor NU \rfloor$. Then return to the table of partial sums and start the search at $p_1 + \cdots + p_Z$. It is easy to show that the expected time, table set-up excluded, is bounded by a constant uniformly over all distributions on N values. The table can be constructed in $O(N)$ time.

Walker (1974, 1977) showed that one can construct in time $O(N)$ a table (q_i, r_i), $1 \leqslant i \leqslant N$, such that the following method works: pick a uniform integer Z from $1, 2, \ldots, N$. Generate a uniform[0, 1] random variate U. If $U \leqslant q_Z$, then return Z, else return r_Z. The values r_i are called the aliases, and the method is now known as the alias method. If a distribution is fixed once and for all, and N is such that the storage of the table is of no concern, then the alias method is difficult to beat.

1.6 Mixture methods

Densities that can be written as mixtures

$$\sum_n p_n f_n(x),$$

with nonnegative weights p_n summing to one, are trivially dealt with by first generating a random index Z with discrete distribution $\{p_n\}$, and then generating a random variate with density f_Z. They do, therefore, not require any special treatment.

Sometimes, we have more intricate mixtures, when, e.g.,

$$f(x) = \mathrm{E}\{g_Z(x)\},$$

where Z is a random variable and g_Z is a density in which Z is a parameter. Clearly, it suffices to generate Z first and then to generate a random variate from g_Z. The textbook example here is $N/\sqrt{G_a}$, where N is standard normal, and G_a is a gamma random variable with parameter a, independent of N. The ratio has the Student t distribution with parameter a. Other examples relevant for further on are U/V, a ratio of two i.i.d. uniform[0, 1] random variates, which has density $(1/2)\min(1, 1/x^2)$, and UV, which has density $\log(1/x)$ on $(0, 1]$. Gamma random variates of parameter less than one are often cumbersome, but we know that $G_a \overset{\mathcal{L}}{=} U^{1/a}G_{a+1}$ where the notation is as above and all variates are independent on the right. Finally, a Cauchy random variate can be obtained as N_1/N_2, the ratio of two independent standard normal random variates.

Mixtures are also useful in discrete settings. For example, the negative binomial distribution (the number of failures before the nth success) with parameters (n, p), $n \geqslant 1$, $p \in (0, 1)$, is given by the probabilities

$$p_k = \binom{n+k-1}{k}(1-p)^k p^n, \quad k \geqslant 0.$$

One can check that this can be generated as a Poisson(Y) random variate where Y in turn is $(1-p)/p$ times a gamma(n) random variate. A special case is the geometric(p) distribution, which is negative binomial(1, p). Here, though, it is better to use the inversion method which can be made explicit by the truncation operator: it suffices to take

$$\lfloor \log_{1-p} U \rfloor,$$

where U is uniform[0, 1].

1.7 The rejection method

Von Neumann (1951) proposed the rejection method, which uses the notion of a dominating measure. Let X have density f on $\mathbb{R}^d$. Let g be another density with the property that for some finite constant $c \geqslant 1$, called the rejection constant,

$$f(x) \leqslant cg(x), \quad x \in \mathbb{R}^d.$$

For any nonnegative integrable function h on $\mathbb{R}^d$, define the body of h as $B_h = \{(x, y)\colon x \in \mathbb{R}^d, 0 \leqslant y \leqslant h(x)\}$. Note that if (X, Y) is uniformly distributed on B_h, then X has density proportional to h. Vice versa, if X has density proportional to h, then $(X, Uh(X))$, where U is uniform[0, 1] and independent of X, is uniformly distributed on B_h. These facts can be used to show the validity of the rejection method:

```
repeat
    generate $U$ uniformly on [0, 1]
    generate $X$ with density $g$
    until $Ucg(X) \leqslant f(X)$
    return $X$.
```

The expected number of iterations before halting is c, so the rejection constant must be kept small. This method requires some analytic work, notably to determine c, but one attractive feature is that we only need the ratio $f(x)/(cg(x))$, and thus, cumbersome normalization constants often cancel out.

1.8 Rejection: a simple example

The rejection principle also applies in the discrete setting, so a few examples follow to illustrate its use in all settings. We begin with the standard normal density. The start is an inequality such as

$$e^{-x^2/2} \leqslant e^{\alpha^2/2-\alpha|x|}.$$

The area under the dominating curve is $e^{\alpha^2/2} \times 2/\alpha$, which is minimized for $\alpha = 1$. Generating a random variate with the Laplace density $e^{-|x|}$ can be done either as SE, where S is a random sign and E is exponential, or as $E_1 - E_2$, a difference of two independent exponential random variables. The rejection algorithm thus reads:

```
    repeat
        generate $U$ uniformly on $[0, 1]$
        generate $X$ with the Laplace density
    until $Ue^{1/2-|X|} \leqslant e^{-X^2/2}$
    return $X$.
```

However, taking logarithms in the last condition, and noting that $\log(1/U)$ is exponential, we can tighten the code using a random sign S and two independent exponentials, E_1, E_2:

```
    generate a random sign $S$
    repeat
        generate $E_1, E_2$
    until $2E_2 > (E_1 - 1)^2$
    return $X \leftarrow SE_1$.
```

It is easy to verify that the rejection constant (the expected number of iterations) is $\sqrt{2e/\pi} \approx 1.35$.

1.9 Rejection: a more advanced example

Assume that f is a monotonically decreasing density on $[0, 1]$, and define $M = f(0)$. We could apply rejection with M as a bound:

```
    repeat
        generate $U$ uniformly on $[0, 1]$
        generate $X$ uniformly on $[0, 1]$
    until $UM \leqslant f(X)$
    return $X$.
```

The expected number of iterations is M. However, the algorithm does not make any use of the monotonicity. As we know that $f(x) \leqslant \min(M, 1/x)$, and the integral of the upper bound (or the rejection constant) is $1 + \log M$, we can exponentially speed up the performance by basing our rejection on the latter bound. It is a simple exercise to show that the indefinite integral of the upper bound is

$$\begin{cases} Mx, & 0 \leqslant x \leqslant \frac{1}{M}, \\ 1 + \log(Mx), & \frac{1}{M} \leqslant x \leqslant 1. \end{cases}$$

A random variate can easily be generated by inversion of a uniform[0, 1] random variate U: if $U \leqslant 1/(1 + \log M)$, return $(1 + \log M)U/M$, else return $(1/M)\exp((1 + \log M)U - 1)$. Thus, we have the improved algorithm:

```
repeat
    generate $U, V$ independently and uniformly on [0, 1]
    set $W = V(1 + \log M)$
    if $W \leqslant 1$ then set $X \leftarrow W/M$
            else set $X \leftarrow \exp(WU - 1)/M$
until $U \min(M, 1/X) \leqslant f(X)$
return $X$.
```

It is possible to beat even this either by spending more time setting up (a one-time cost) or by using additional information on the density. In the former case, partition $[0, 1]$ into k equal intervals, and invest once in the calculation of $f(i/k)$, $0 \leqslant i < k$. Then the histogram function of height $f(i/k)$ on $[i/k, (i + 1)/k)$ dominates f and can be used for rejection purposes. This requires the bin probabilities $p_i = f(i/k)/\sum_{j=0}^{k-1} f(j/k)$. Rejection can proceed as follows:

```
repeat
    generate $U, V$ independently and uniformly on [0, 1]
    generate $Z$ on $\{0, 1, \ldots, k - 1\}$ according to the probabilities $\{p_i\}$
    set $X = (Z + V)/k$
until $Uf(Z/k) \leqslant f(X)$
return $X$.
```

Here the area under the dominating curve is at most one plus the area sandwiched between f and it. By shifting that area over to the left, bin by bin, one can bound it further by $1 + M/k$. Thus, we can adjust k to reduce the rejection constant at will. For example, a good choice would be $k = \lceil M \rceil$. The expected time per variate would be O(1) if we can generate the random integer Z in time O(1). This can be done, but only at the expense of the set-up of a table of size k (see above). In one shot situations, this is a prohibitive expense. In Devroye (1986a), methods based on partitioning the density in slabs are called strip methods. Early papers in this direction include Ahrens and Dieter (1989), Ahrens (1993, 1995), Marsaglia and Tsang (1984) and Marsaglia et al. (1964).

Assume next that we also know the mean μ of f. Then, even dropping the requirement that f be supported on [0, 1], we have

$$\mu = \int_0^\infty y f(y)\,\mathrm{d}y \geqslant f(x) \int_0^x y\,\mathrm{d}y = f(x)\frac{x^2}{2}.$$

Hence,

$$f(x) \leqslant \min\left(M, \frac{2\mu}{x^2}\right).$$

The integral of the upper bound from 0 to x is

$$\begin{cases} Mx, & 0 \leqslant x < \sqrt{2\frac{\mu}{M}}, \\ 2\sqrt{2\mu M} - \frac{2\mu}{x}, & x \geqslant \sqrt{2\frac{\mu}{M}}. \end{cases}$$

The area under the upper bound is therefore $2\sqrt{2\mu M}$. To generate a random variate with density proportional to the upper bound, we can proceed in various ways, but it is easy to check that $\sqrt{2\mu/M}U_1/U_2$ fits the bill, where U_1, U_2 are independent uniform[0, 1] random variables. Thus, we summarize:

```
repeat
  generate $U, V, W$ independently and uniformly on [0, 1]
  set $X = \sqrt{2\mu/M}V/W$
until $UM \min(1, (W/V)^2) \leqslant f(X)$
return $X$.
```

This example illustrates the development of universal generators, valid for large classes of densities, as a function of just a few parameters. The reader will be able to generalize the above example to unimodal densities with known rth moment about the mode. The example given above applies with trivial modifications to all unimodal densities, and the expected time is bounded by a constant times $\sqrt{f(m)\mathrm{E}\{|X-m|\}}$, where m is the mode. For example, the gamma density with parameter $a \geqslant 1$ has mean a, and mode at $a-1$, with $f(a-1) \leqslant C/\sqrt{a}$ for some constant C, and $\mathrm{E}\{|X-(a-1)|\} \leqslant C'\sqrt{a}$ for another constant C'. Thus, the expected time taken by our universal algorithm takes expected time bounded by a constant, uniformly over $a \geqslant 1$. We say that the algorithm is uniformly fast over this class of densities.

1.10 The alternating series method

To apply the rejection method, we do not really need to know the ratio $f(x)/(cg(x))$ exactly. It suffices that we have an approximation $\phi_n(x)$ that tends, as $n \to \infty$, to $f(x)/(cg(x))$, and for which we know a monotone error bound $\varepsilon_n(x) \downarrow 0$. In that case, we let n increase until for the first time, either

$$U \leqslant \phi_n(X) - \varepsilon_n(X)$$

(in which case we accept X) or

$$U \geqslant \phi_n(X) + \varepsilon_n(X)$$

(in which case we reject X). Equivalently, we have computable bounds $\xi_n(x)$ and $\psi_n(x)$ with the property that $\xi_n(x) \uparrow f(x)/(cg(x))$ and $\psi_n(x) \downarrow f(x)/(cg(x))$ as $n \to \infty$. This approach is useful when the precise computation of f is impossible, e.g., when f is known as infinite series or when f can never be computed exactly using only finitely many resources. It was first developed for the Kolmogorov–Smirnov limit distribution in Devroye (1981a). For another use of this idea, see Keane and O'Brien's Bernoulli (1994) factory:

```
repeat
   generate $U$ uniformly on [0, 1]
   generate $X$ with density $g$
   set $n = 0$
   repeat $n \leftarrow n + 1$ until $U \leqslant \xi_n(X)$ or $U \geqslant \psi_n(X)$
until $U \leqslant \xi_n(X)$
return $X$.
```

The expected number of iterations in the outer loop is still c, as in the rejection method. However, to take the inner loop into account, let N be the largest index n attained in the inner loop. Note that $N > t$ implies that $U \in [\xi_t(X), \psi_t(X)]$. Thus,

$$\mathrm{E}\{N|X\} = \sum_{t=0}^{\infty} \mathrm{P}\{N > t|X\} \leqslant \sum_{t=0}^{\infty} \bigl(\psi_t(X) - \xi_t(X)\bigr).$$

Unconditioning,

$$\mathrm{E}\{N\} \leqslant \sum_{t=0}^{\infty} \mathrm{E}\bigl\{\psi_t(X) - \xi_t(X)\bigr\}.$$

To be finite, this requires a rate of decrease in the approximation error that is at least $1/t$. One must also take into account that the computation of ξ_t and ψ_t may also grow with t. The bound above is indicative of the expected complexity only when, say, ξ_t can be computed from ξ_{t-1} in one time unit.

We cannot stress strongly enough how important the alternating series method is, as it frees us from having to compute f exactly. It is indeed the key to the solution of a host of difficult nonuniform random variate generation problems.

2 Uniformly bounded times

If $\mathcal{F}$ is a class of distributions, it is useful to have generators for which the expected time is uniformly bounded over $\mathcal{F}$. In the case of the rejection method, this often (but not always, see, e.g., the alternating series method) means bounding the rejection constant uniformly. For example, pre-1970 papers routinely recommended generating gamma(k) random variables as a sum of k independent exponentials, and generating a binomial(n, p) random variate as a sum of n independent Bernoulli(p) random variables. Even today, many may still simulate a sum of n i.i.d. random variables by generating all n of them and summing. So, whether $\mathcal{F}$ is the class of all gamma distributions, all binomial distributions, or all sums of distributions with a fixed density f, the idea of a uniformly bounded time is important, and has, in fact, given new life to the area of nonuniform random variate generation. In this section, we briefly survey some key classes $\mathcal{F}$, and provide some references to good generators.

Many methods in the literature are rejection methods in disguise. For example, for real-valued random variates, the ratio of uniforms method is based upon the use of bounding curves of the form

$$\min\left(A, \frac{B}{x^2}\right).$$

The area under the bounding curve is $4\sqrt{AB}$, and a random variate with density proportional to it can be generated as $\sqrt{B/A}SU/V$, where S is a random sign, and U, V are independent uniform[0, 1] random variates. The method is best applied after shifting the mode or mean of the density f to the origin. One could for example use

$$A = \sup_x f(x), \qquad B = \sup_x x^2 f(x).$$

In many cases, such as for mode-centered normals, gamma, beta, and Student t distributions, the rejection constant $4\sqrt{AB}$ is uniformly bounded over all or a large part of the parameter space. Discretized versions of it can be used for the Poisson, binomial and hypergeometric distributions.

Uniformly fast generators have been obtained for all major parametric families. Interesting contributions include the gamma generators of Cheng (1977), Le Minh (1988), Marsaglia (1977), Ahrens and Dieter (1982), Cheng and Feast (1979, 1980), Schmeiser and Lal (1980), Ahrens et al. (1983) and Best (1983), the beta generators of Ahrens and Dieter (1974), Zechner and Stadlober (1993), Best (1978a, 1978b) and Schmeiser and Babu (1980), the binomial methods of Stadlober (1988, 1989), Ahrens and Dieter (1980), Hörmann (1993a) and Kachitvichyanukul and Schmeiser (1988, 1989), the hypergeometric generators of Stadlober (1988, 1990), and the code for Poisson variates by Ahrens and Dieter (1980, 1991), Devroye (1981c) and Hörmann (1993b). Some of these algorithms are described in Devroye (1986a), where some additional uniformly fast methods can be found.

All the distributions mentioned above are log-concave, and it is thus no surprise that we can find uniformly fast generators. The emphasis in most papers is on the details, to squeeze the last millisecond out of each piece of code. To conclude this section, we will just describe a recent gamma generator, due to Marsaglia and Tsang (2001). It uses almost-exact inversion. Its derivation is unlike anything in the present survey, and shows the careful planning that goes into the design of a nearly perfect generator. The idea is to consider a monotone transformation $h(x)$, and to note that if X has density

$$\mathrm{e}^{g(x)} \overset{\text{def}}{=} \frac{h(x)^{a-1}\mathrm{e}^{-h(x)}h'(x)}{\Gamma(a)},$$

then $Y = h(X)$ has the gamma(a) density. If $g(x)$ is close to $c - x^2/2$ for some constant c, then a rejection method from a normal density can be developed. The choice suggested by Marsaglia and Tsang is

$$h(x) = d(1+cx)^3, \quad -\frac{1}{c} < x < \infty,$$

with $d = a - 1/3$, $c = 1/\sqrt{9d}$. In that case, simple computations show that

$$g(x) = 3d\log(1+cx) - d(1+cx)^3 + d + C, \quad -\frac{1}{c} < x < \infty,$$

for some constant C. Define $w(x) = -x^2/2 - g(x)$. Note that $w'(x) = -x - 3dc/(1+cx) + 3dc(1+cx)^2$, $w''(x) = -1 + 3dc^2/(1+cx)^2 + 6dc^2(1+cx)$, $w'''(x) = -6dc^3/(1+cx)^3 + 6dc^3$. The third derivative is zero at $(1+cx)^3 = 1$, or $x = 0$. Thus, w'' reaches a minimum there, which is $-1 + 9dc^2 = 0$. Therefore, by Taylor's series expansion and the fact that $w(0) = -C$, $w'(0) = 0$, we have $w(x) \geqslant C$. Hence,

$$\mathrm{e}^{g(x)} \leqslant \mathrm{e}^{C-x^2/2}$$

and the rejection method immediately yields the following:

```
$d \leftarrow a - 1/3$
$c \leftarrow 1/\sqrt{9d}$
**repeat**
   **generate** $E$ exponential
   **generate** $X$ normal
   $Y \leftarrow d(1+cX)^3$
**until** $-X^2/2 - E \leqslant d\log(Y) - Y + d$
**return** $Y$.
```

We picked d so that $\int \mathrm{e}^{g-C}$ is nearly maximal. By the transformation $y = d(1+cx)^3$ and by Stirling's approximation, as $d \to \infty$,

$$\int_{-1/c}^{\infty} \mathrm{e}^{g(x)-C}\,\mathrm{d}x = \int_0^{\infty} \frac{y^{d-2/3}\mathrm{e}^{d-y}}{3cd^{d+1/3}}\,\mathrm{d}y = \frac{\Gamma(d+1/3)\mathrm{e}^d}{3cd^{d+1/3}}$$
$$\sim \frac{(1+1/3d)^{d+1/3}\sqrt{2\pi}}{3c\mathrm{e}^{1/3}\sqrt{d+1/3}} \sim \frac{\sqrt{18\pi d}}{3\sqrt{d+1/3}} \sim \sqrt{2\pi}.$$

Using $\int \mathrm{e}^{-x^2/2}\,\mathrm{d}x = \sqrt{2\pi}$, we note that the asymptotic (as $a \to \infty$) rejection constant is 1: we have a perfect fit! Marsaglia and Tsang recommend the method only for $a \geqslant 1$. For additional speed, a squeeze step (or quick acceptance step) may be added.

3 Universal generators

It is quite important to develop generators that work well for entire families of distributions. An example of such development is the generator for log concave densities given in Devroye (1984c), that is, densities for which $\log f$ is concave. On the real line, this has several consequences: these distributions are unimodal and have sub-exponential tails. Among the log concave densities, we find the normal densities, the gamma family with parameter $a \geqslant 1$, the Weibull family with parameter $a \geqslant 1$, the beta family with both parameters greater than or equal to 1, the exponential power distribution with parameter greater than or equal to 1 (the density being proportional to $\exp(-|x|^a)$), the Gumbel or double exponential distribution ($k^k \exp(-kx - k\mathrm{e}^{-x})/(k-1)!$ is the density, $k \geqslant 1$ is the parameter), the generalized inverse Gaussian distribution, the logistic distribution, Perks' distribution (with density proportional to $1/(\mathrm{e}^x + \mathrm{e}^{-x} + a)$, $a > -2$), the hyperbolic secant distribution (with distribution function $(2/\pi)\arctan(\mathrm{e}^x)$), and Kummer's distribution (for a definition, see Devroye, 1984a). Also, we should consider simple nonlinear transformations of other random variables. Examples include $\arctan X$ with X Pearson IV, $\log|X|$ (X being Student t for all parameters), $\log X$ (for X gamma, all parameters), $\log X$ (for X log-normal), $\log X$ (for X Pareto), $\log(X/(1-X))$ (for all beta variates X) and $\log X$ (for X beta(a,b) for all $a > 0$, $b \geqslant 1$).

In its most primitive version, assuming one knows that a mode occurs at m, we consider the generation problem for the normalized random variable $Y = f(m)(X - m)$. One may obtain X as $m + Y/f(m)$. The new random variable has a mode at 0 of value 1. Call its density g. Devroye (1984c) showed that

$$g(y) \leqslant \min\bigl(1, \mathrm{e}^{1-|y|}\bigr).$$

The integral of the upper bound is precisely 4. Thus, the rejection constant will be 4, uniformly over this vast class of densities. A random variable with density proportional to the upper bound is generated by first flipping a perfect coin. If it is heads, then generate $(1 + E)S$, where E is exponential, and S is another perfect coin. Otherwise generate US, where U is uniform$[0, 1]$ and independent of S. The algorithm is as follows:

```
repeat
    generate $U$ (a uniform on $[0, 1]$), $E$ (exponential) and $S$ (a fair random bit)
    generate a random sign $S'$
    if $S = 0$ then $Y \leftarrow 1 + E$
               else $Y \leftarrow V$, $V$ uniform$[0, 1]$
    set $Y \leftarrow YS'$
until $U \min(1, \mathrm{e}^{1-|Y|}) \leqslant g(Y)$
return $Y$.
```

Various improvements can be found in the original paper. In adaptive rejection sampling, Gilks and Wild (1992) bound the density adaptively from above and below by a number of exponential pieces, to reduce the integral under the bounding curve. If we denote the log concave density by f, and if f' is available, then for all x, log-concavity implies that

$$\log f(y) \leqslant \log f(x) + (y-x)\frac{f'(x)}{f(x)}, \quad y \in \mathbb{R}.$$

This is equivalent to

$$f(y) \leqslant f(x)\mathrm{e}^{(y-x)f'(x)/f(x)}, \quad y \in \mathbb{R}.$$

Assume that we have already computed f at k points. Then we can construct $k+1$ exponential pieces (with breakpoints at those k points) that bound the density from above, and in fact, sampling from the piecewise exponential dominating curve is easily carried out by inversion. The integral under a piece anchored at x and spanning $[x, x+\delta)$ is

$$\frac{f^2(x)(\mathrm{e}^{\delta f'(x)/f(x)} - 1)}{f'(x)}.$$

Gilks and Wild also bound f from below. Assume that $y \in [x, x']$, where x and x' are two of those anchor points. Then

$$\log\bigl(f(y)\bigr) \geqslant \log\bigl(f(x)\bigr) + \frac{y-x}{x'-x}\bigl(\log\bigl(f(x')\bigr) - \log\bigl(f(x)\bigr)\bigr).$$

The lower bound can be used to avoid the evaluation of f most of the time when accepting or rejecting. It is only when f is actually computed that a new point is added to the collection to increase the number of exponential pieces.

Hörmann (1995) shows that it is best to add two exponential tails that touch g at the places where $g(x) = 1/\mathrm{e}$, assuming that the derivative of g is available. In that case, the area under the bounding curve is reduced to $\mathrm{e}/(\mathrm{e}-1) \approx 1.582$. Hörmann and Derflinger (1994) and Hörmann (1994, 1995) (see also Leydold, 2000a, 2000b, 2001; Hörmann et al., 2004) define the notion of T_c-concavity, related to the concavity of $-1/(f(x))^c$, where $0 < c < 1$. For example, the Student t distribution of parameter greater than or equal to 1 is $T_{1/2}$-concave. Similar universal methods related to the one above are developed by them. Rejection is done from a curve with polynomial tails, and the rejection constant is $(1-(1-c)^{1/c-1})^{-1}$. The notion of T-convexity proposed by Evans and Swartz (1996, 2000) is useful for unbounded densities.

Discrete distributions are called log concave if for all n,

$$p_n^2 \geqslant p_{n-1}p_{n+1}.$$

Some examples are shown in Table 2. Others not shown here included the Pólya–Eggenberger distribution (see Johnson and Kotz, 1969) and the hyper-Poisson distribution. For all discrete log-concave distributions having a mode at m, we have (Devroye, 1987a)

$$p_{m+k} \leqslant p_m \min\bigl(1, \mathrm{e}^{1-p_m|k|}\bigr), \quad \text{all } k.$$

Table 2.
Some important discrete distributions and their parameter ranges

Name	Probability vector p_k	Parameter(s)
Binomial(n, p)	$\binom{n}{k} p^k (1-p)^{n-k}, 0 \leqslant k \leqslant n$	$n \geqslant 1, 0 \leqslant p \leqslant 1$
Poisson(λ)	$\lambda^k \mathrm{e}^{-\lambda}/k!, k \geqslant 0$	$\lambda > 0$
Negative binomial(n, p)	$\binom{n+k-1}{n-1} (1-p)^k p^n, k \geqslant 0$	$n \geqslant 1, 0 \leqslant p \leqslant 1$
Geometric(p)	$p(1-p)^k, k \geqslant 0$	$0 \leqslant p \leqslant 1$
Logarithmic series(p)	$p^k/(k! \log(1/(1-p))), k \geqslant 1$	$0 < p < 1$
Hypergeometric(b, w, n)	$\frac{\binom{w}{k}\binom{b}{n-k}}{\binom{w+b}{n}}, 0 \leqslant k \leqslant \min(w, n)$	$n \geqslant 1, b \geqslant 0, w \geqslant 0$

Mimicking the development for densities, we may use rejection from a curve that is flat in the middle, and has two exponential tails. After taking care of rounding to the nearest integer, we obtain the following code, in which the expected number of iterations before halting is $4 + p_m \leqslant 5$ for all distributions in the family:

```
compute $w \leftarrow 1 + p_m/2$ (once)
repeat
    generate $U, V, W$ uniformly on [0, 1], and let $S$ be a random sign
    if $U \leqslant w/(1+w)$ then $Y \leftarrow Vw/p_m$
                           else $Y \leftarrow (w - \log V)/p_m$
    $X \leftarrow S\,\mathrm{round}(Y)$
until $W \min(1, \mathrm{e}^{w - p_m Y}) \leqslant p_{m+X}/p_m$
return $m + X$.
```

We can tighten the rejection constant further to $2 + p_m$ for log concave distributions that are monotonically decreasing. The disadvantage in the code is the requirement that p_m must be computed.

The development for log-concave distributions can be aped for other classes. For example, the adaptive rejection method of Gilks and Wild can be used for unimodal densities with known mode m. One just needs to replace the exponential upper and lower bounds by histograms. This is covered in an exercise in Devroye (1986a). General algorithms are known for all Lipschitz densities with known bounds on the Lipschitz constant C and variance σ^2, and all unimodal densities with mode at m, and variance bounded by σ^2, to give two examples (see Devroye, 1984b, 1986a).

4 Indirect problems

A certain amount of work is devoted to indirect problems, in which distributions are not described by their density, distribution function or discrete probability vector. A partial list follows, with brief descriptions on how the solutions may be constructed.

4.1 Characteristic functions

If the characteristic function

$$\varphi(t) = \mathrm{E}\{\mathrm{e}^{\mathrm{i}tX}\}$$

is known in black box format, very little can be done in a universal manner. In particular cases, we may have enough information to be able to deduce that a density exists and that it is bounded in a certain way. For example, from the inversion formula

$$f(x) = \frac{1}{2\pi}\int \varphi(t)\mathrm{e}^{-\mathrm{i}tx}\,\mathrm{d}t,$$

we deduce the bounds

$$\sup_x f(x) \leqslant M \stackrel{\text{def}}{=} \frac{1}{2\pi}\int \left|\varphi(t)\right|\mathrm{d}t$$

and

$$\sup_x x^2 f(x) \leqslant M' \stackrel{\text{def}}{=} \frac{1}{2\pi}\int \left|\varphi''(t)\right|\mathrm{d}t.$$

Thus, we have

$$f(x) \leqslant \min\left(M, \frac{M'}{x^2}\right),$$

a bound that is easily dealt with by rejection. In principle, then rejection can be used, with $f(x)$ approximated at will by integral approximations based on the inversion integral. This method requires bounds on the integral approximation, and these depend in turn on the smoothness of φ. Explicit bounds on some smoothness parameters must be available. There are descriptions and worked out examples in Devroye (1981b, 1986b, 1988). Devroye (1988) uses this method to simulate the sum S_n of n i.i.d. random variables with common characteristic function φ in expected time not depending upon n, as S_n has characteristic function φ^n.

Pólya showed that any convex function φ on the positive halfline that decreases from 1 to 0 is the characteristic function of a symmetric random variable if we extend it to the real line by setting $\varphi(-t) = \varphi(t)$. These distributions correspond to random variables distributed as Y/Z, where Y and Z are independent, Y has the de la Vallée–Poussin density

$$\frac{1}{2\pi}\left(\frac{\sin(x/2)}{x/2}\right)^2, \quad x \in \mathbb{R}$$

(for which random variate generation by rejection is trivial), and Z has distribution function on the positive halfline given by $1 - \varphi(t) + t\varphi'(t)$. See Dugué and Girault (1955) for this property and Devroye (1984a) for its implications in random variate generation. Note that we have made a space warp from the complex to the real domain. This has unexpected corollaries. For example, if E_1, E_2 are independent exponential random variables, U is a uniform$[0, 1]$ random variable and $\alpha \in (0, 1]$, then Y/Z with

$$Z = (E_1 + E_2 \mathbf{1}_{U<\alpha})^{1/\alpha}$$

is symmetric stable of parameter (α), with characteristic function $\mathrm{e}^{-|t|^\alpha}$. The Linnik–Laha distribution (Laha, 1961) with parameter $\alpha \in (0, 1]$ is described by

$$\varphi(t) = \frac{1}{1 + |t|^\alpha}.$$

Here Y/Z with

$$Z = \left(\frac{\alpha + 1 - \sqrt{(\alpha + 1)^2 - 4\alpha U}}{2U} - 1\right)^{1/\alpha}$$

has the desired distribution.

4.2 Fourier coefficients

Assume that the density f, suitably scaled and translated, is supported on $[-\pi, \pi]$. Then the Fourier coefficients are given by

$$a_k = \mathrm{E}\left\{\frac{\cos(kX)}{\pi}\right\}, \qquad b_k = \mathrm{E}\left\{\frac{\sin(kX)}{\pi}\right\}, \quad k \geqslant 0.$$

They uniquely describe the distribution. If the series absolutely converges,

$$\sum_k |a_k| + |b_k| < \infty,$$

then the following trigonometric series is absolutely and uniformly convergent to f,

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \bigl(a_k \cos(kx) + b_k \sin(kx)\bigr).$$

If we only use terms with index up to n, then the error made is not more than

$$R_{n+1} = \sum_{k=n+1}^{\infty} \sqrt{a_k^2 + b_k^2}.$$

If we know bounds on R_n, then one can use rejection with an alternating series method to generate random variates (Devroye, 1989).

A particularly simple situation occurs when only the cosine coefficients are nonzero, as occurs for symmetric distributions. Assume furthermore that the Fourier cosine series coefficients a_k are convex and $a_k \downarrow 0$ as $k \to \infty$. Even without absolute convergence, the Fourier cosine series converges, and can in fact be rewritten as follows:

$$f(x) = \sum_{k=0}^{\infty} \pi(k+1)\Delta^2 a_k K_k(x),$$

where $\Delta^2 a_k = a_{k+2} - 2a_{k+1} + a_k$ (a positive number, by convexity) and

$$K_k(x) = \frac{1}{2\pi(k+1)}\left(\frac{\sin((k+1)x/2)}{\sin(x/2)}\right)^2, \quad |x| \leqslant \pi,$$

is the Fejer kernel. As $K_k(x) \leqslant \min((k+1)/4, \pi/(2(k+1)x^2))$, random variates from it can be generated in expected time uniformly bounded in k by rejection. Thus, the following simple algorithm does the job:

```
**generate** $U$ uniformly on [0, 1]
$Z \leftarrow 0, S \leftarrow \pi\Delta^2 a_0$.
**while** $U > S$ **do**
   $Z \leftarrow Z + 1$.
   $S \leftarrow S + \pi(Z+1)\Delta^2 a_Z$
**generate** $X$ with Fejer density $K_Z$
**return** $X$.
```

4.3 The moments are known

Let μ_n denote the nth moment of a random variable X. One could ask to generate a random variate when for each n, μ_n is computable (in a black box). Again, this is a nearly impossible problem to solve, unless additional information is available. For one thing, there may not be a unique solution. A sufficient condition that guarantees the uniqueness is Carleman's condition

$$\sum_{n=0}^{\infty} |\mu_{2n}|^{-1/(2n)} = \infty$$

(see Akhiezer, 1965). Sufficient conditions in terms of a density f exist, such as Krein's condition

$$\int \frac{-\log(f(x))}{1+x^2}\,\mathrm{d}x = \infty,$$

combined with Lin's condition, applicable to symmetric and differentiable densities, which states that $x|f'(x)|/f(x) \uparrow \infty$ as $x \to \infty$ (Lin, 1997; Krein, 1944; Stoyanov, 2000). Whenever the distribution is of compact support, the moments determine the distribution. In fact, there are various ways for reconstructing the density from the moments. An example is provided by the series

$$f(x) = \sum_{k=0}^{\infty} a_k \phi_j(x), \quad |x| \leqslant 1,$$

where ϕ_k is the Legendre polynomial of degree k and a_k is a linear function of all moments up to the kth. The series truncated at index n has an error not exceeding

$$\frac{C_r \int |f^{(r+1)}|}{(1-x^2)^{1/4} n^{r-1/2}},$$

where r is an integer and C_r is a constant depending upon r only (Jackson, 1930). Rejection with dominating curve of the form $C'/(1-x^2)^{1/4}$ (a symmetric beta) and with the alternating series method can thus lead to a generator (Devroye, 1989).

The situation is much more interesting when X is supported on the positive integers. Let

$$M_r = \mathrm{E}\{X(X-1)\cdots(X-r+1)\}$$

denote the rth factorial moment. Then approximate $p_j = \mathrm{P}\{X = j\}$ by

$$p_{nj} = \frac{1}{j!} \sum_{i=0}^{n} (-1)^i \frac{M_{j+i}}{i!}.$$

Note that $p_{nj} \geqslant p_j$ for n even, $p_{nj} \leqslant p_j$ for n odd, and $p_{nj} \to p_j$ as $n \to \infty$ provided that $\mathrm{E}\{(1+u)^X\} < \infty$ for some $u > 0$. Under this condition, an alternating series rejection method can be implemented, with a dominating curve suggested by the crude bound

$$p_j \leqslant \min\left(1, \frac{\mu_r}{j^r}\right).$$

In fact, we may even attempt inversion, as the partial sums $p_0 + p_1 + \cdots + p_j$ are available with any desired accuracy: it suffices to increase n until we are absolutely sure in which interval a given uniform random variate U lies. For more details, see Devroye (1991).

4.4 The moment generating function

For random variables on the positive integers, as in the previous section, we may have knowledge of the moment generating function

$$k(s) = p_0 + p_1 s + p_2 s^2 + \cdots = \mathrm{E}\{s^X\}.$$

Here we have trivially

$$p_j \leqslant \frac{k(s)}{s^j},$$

where $s > 0$ can be picked at will. Approximations for p_j can be constructed based on differences of higher orders. For example, if we take $t > 0$ arbitrary, then we can define

$$p_{nj} = \frac{\sum_{i=0}^{j} (-1)^{j-i} \binom{j}{i} k(it)}{j!t^j}$$

and note that

$$0 \leqslant p_{nj} - p_j \leqslant \frac{1}{(1-jt)^{j+1}} - 1.$$

This is sufficient to apply the alternating series method (Devroye, 1991).

4.5 An infinite series

Some densities are known as infinite series. Examples include the theta distribution (Rényi and Szekeres, 1967) with distribution function

$$F(x) = \sum_{j=-\infty}^{\infty} (1 - 2j^2x^2)\mathrm{e}^{-j^2x^2} = \frac{4\pi^{5/2}}{x^3} \sum_{j=1}^{\infty} j^2 \mathrm{e}^{-\pi^2 j^2/x^2}, \quad x > 0,$$

and the Kolmogorov–Smirnov distribution (Feller, 1948) with distribution function

$$F(x) = 1 - 2\sum_{j=1}^{\infty} (-1)^j \mathrm{e}^{-2j^2x^2}, \quad x > 0.$$

In the examples above, it is relatively straightforward to find tight bounding curves, and to apply the alternating series method (Devroye, 1981a, 1997).

4.6 Hazard rates

Let X be a positive random variable with density f and distribution function F. Then the hazard rate, the probability of instantaneous death given that one is still alive, is given by

$$h(x) = \frac{f(x)}{1 - F(x)}.$$

Assume that one is given h (note that $h \geqslant 0$ must integrate to ∞). Then f can be recovered as follows:

$$f(x) = h(x) \exp\left\{-\int_0^x h(y)\,\mathrm{d}y\right\}.$$

This may pose a problem for random variate generation. Various algorithms for simulation in the presence of hazard rates are surveyed by Devroye (1986c), the most useful among which is the thinning method of Lewis and Shedler

(1979): assume that $h \leqslant g$, where g is another hazard rate. If $0 < Y_1 < Y_2 < \cdots$ is a nonhomogeneous Poisson point process with rate function g, and $U_1, U_2, \ldots$ is a sequence of independent uniform[0, 1] random variables, independent of the Y_i's, and if i is the smallest index for which $U_i g(Y_i) \leqslant f(Y_i)$, then $X = Y_i$ has hazard rate h.

This is not a circular argument if we know how to generate a nonhomogeneous Poisson point process with rate function g. Define the cumulative rate function $G(x) = \int_0^x g(y)\,\mathrm{d}y$. Let $E_1, E_2, \ldots$ be i.i.d. exponential random variables with partial sums $S_i = \sum_{j=1}^{i} E_j$. Then the sequence $Y_i = G^{\mathrm{inv}}(S_i)$, $i \geqslant 1$, is a nonhomogeneous Poisson point process with rate g. Thus, the thinning method is a Poisson process version of the rejection method.

If $h(0)$ is finite, and X has the given hazard rate, then for a DHR (decreasing hazard rate) distribution, we can use $g = h(0)$. In that case, the expected number of iterations before halting is $\mathrm{E}\{h(0)X\}$. However, we can dynamically thin (Devroye, 1986c) by lowering g as values $h(Y_i)$ trickle in. In that case, the expected time is finite when X has a finite logarithmic moment, and in any case, it is not more than $4 + \sqrt{24\,\mathrm{E}\{h(0)X\}}$.

4.7 A distributional identity

The theory of fixed points and contractions can be used to derive many limit laws in probability (see, e.g., Rösler and Rüschendorf, 2001). These often are described as distributional identities known to have a unique solution. For example, the identity

$$X \overset{\mathcal{L}}{=} W(X + 1),$$

where W is a fixed random variable on $[0, 1]$ and $X \geqslant 0$, sometimes has a unique solution for X. By iterating the identity, we see that if the solution exists, it can be represented as

$$X \overset{\mathcal{L}}{=} W_1 + W_1 W_2 + W_1 W_2 W_3 + \cdots,$$

where the W_i's are i.i.d. These are known as perpetuities (Vervaat, 1979; Goldie and Grübel, 1996). Some more work in the complex domain can help out: for example, when $W = U^{1/\alpha}$, where U is uniform[0, 1] and $\alpha > 0$ leads to the characteristic function

$$\varphi(t) = \exp\left\{\alpha \int_0^1 \frac{\mathrm{e}^{\mathrm{i}tx} - 1}{x}\,\mathrm{d}x\right\}.$$

For the particular case $\alpha = 1$, we have Dickman's distribution (Dickman, 1930), which is the limit law of

$$\frac{1}{n} \sum_{i=1}^{n} i Z_i,$$

where Z_i is Bernoulli$(1/i)$. None of the three representations above (infinite series, Fourier transform, limit of a discrete sum) gives a satisfactory path to an exact algorithm (although approximations are trivial). The solution in Devroye (2001) uses the characteristic function representation in an alternating series method, after first determining explicit bounds on the density of f that can be used in the rejection method. A simpler approach to this particular distributional identity is still lacking, and a good exact solution for distributional identities in general is sorely needed.

Consider distributional identities of the form

$$X \overset{\mathcal{L}}{=} f_1(U)X + f_2(U)X' + f_3(U),$$

where the f_i's are known functions, U is a uniform random variate and X, X' are i.i.d. and independent of U. Exact generation here is approached by discretization in Devroye and Neininger (2002): suitably discretize the distributions of $f_i(U)$, and apply the map n times, starting with arbitrary values (say, zero) for X, X'. If X_n is the random variable thus obtained, and F_n is its distribution function, note that F_n can be calculated as a sum, due to the discretization. Some theoretical work is needed (which involves parameters of the f_i's) to obtain inequalities of the type

$$\left|\frac{F_n(x+\delta)-F_n(x)}{\delta} - f(x)\right| \leqslant R_n(x),$$

where $R_n(x)$ is explicitly known and can be made as small as desired by appropriate choice of all the parameters (the discretization, δ and n). This suffices to apply the alternating series method, yet again. Devroye et al. (2000) use this method to generate random variates from the limit distribution for the quicksort complexity. Fill (1998) has another way of doing perfect sampling which can be used for some simple distributional identities.

4.8 Kolmogorov and Lévy measures

Infinitely divisible distributions (Sato, 2000) have several representations. One of them is Kolmogorov's canonical representation for the logarithm of the characteristic function

$$\log \varphi(t) = \mathrm{i}ct + \int \frac{\mathrm{e}^{\mathrm{i}tx} - 1 - \mathrm{i}tx}{x^2}\, \mathrm{d}K(x),$$

where $K(-\infty) = 0$, $K \uparrow$, and $K(\infty) - K(-\infty) = \sigma^2$. Thus, K can be viewed as some sort of measure. When K puts mass σ^2 at the origin, then we obtain the normal distribution. When K puts mass λ at 1, then the Poisson(λ) distribution is obtained. However, in general, we only have representations as integrals. For example, with $c = 1$, $K(x) = \min(x^2/2, 1/2)$ on the positive halfline, we obtain Dickman's distribution,

$$\log \varphi(t) = \int_0^1 \frac{\mathrm{e}^{\mathrm{i}tx} - 1}{x}\, \mathrm{d}x.$$

Bondesson (1982) presents approximations when one is given the measure K. Basically, if K is approximated by n point masses each of total weight σ^2/n, then the characteristic function can be written as a product, where each term in the product, assuming $c = 0$, is of the form

$$\exp\{\alpha(\mathrm{e}^{\mathrm{i}t\beta} - 1 - \mathrm{i}t\beta)\}$$

for fixed values of α and β. This is the characteristic function of $\beta(X - \alpha)$ where X is Poisson(α). Thus, approximations consist of sums of such random variables. However, if the true distribution has a density, such discrete approximations are not acceptable.

5 Random processes

A random process X_t, $t \in \mathbb{R}^d$, cannot be produced using a finite amount of resources, as the domain of the index is uncountable. So, it is customary to say that we can generate a random process exactly if for any k, and for any set of indices $t_1, \ldots, t_k$, we can generate the random vector $(X_{t_1}, \ldots, X_{t_k})$ exactly. Often, these points are nicely spaced out. In many processes, we have particularly elegant independence properties. For example, for all Lévy processes and setting $d = 1$, we have $X_0 = 0$, $X_{t_1}, X_{t_2} - X_{t_1}, \ldots, X_{t_k} - X_{t_{k-1}}$ independent, and furthermore, less important to us, the distribution of $X_{s+t} - X_s$ depends upon t only and we have continuity: $\mathrm{P}\{|X_{s+t} - X_s| > \varepsilon\} \to 0$ as $t \downarrow 0$ for all $\varepsilon > 0$. With that independence alone, we can thus generate each piece independently. When k is large, to protect against a proliferation of errors, it is wise to use a dyadic strategy: first generate the process at $t_{k/2}$, then at $t_{k/4}$ and $t_{3k/4}$, and so on. The dyadic trick requires that we know the distribution of $X_{s+u} - X_s$ conditional on $X_{s+t} - X_s$ for all $0 \leqslant u \leqslant t$.

An example suffices to drive home this point. In Brownian motion ($S(t)$, $t \geqslant 0$) (a Lévy process), we have differences $X_{s+t} - X_s$ that are distributed as $\sqrt{t}N$, where N is a standard normal random variable. Furthermore, setting $u = t/2$, we see that

$$X_{s+t} - X_s = (X_{s+t} - X_{s+u}) + (X_{s+u} - X_s)$$

is a sum of two independent normal random variables with variances $t/2$ each. Conditional on $X_{s+t} - X_s$, $X_{s+u} - X_s$ thus is distributed as

$$\frac{X_{s+t} - X_s}{2} + \sqrt{\frac{t}{2}}N.$$

Therefore, the dyadic trick applies beautifully. In the same manner, one can simulate the Brownian bridge on $[0, 1]$, $B(t) = S(t) - tS(1)$. For the first systematic examples of the use of this splitting trick, see Caflisch et al. (1997) and Fox (1999).

In a more advanced example, consider a gamma process $S(t)$, where $S(t)$ is distributed as a gamma random variable with parameter αt. This is not a Lévy process, but rather a monotonically increasing process. We have the following recursive property: given $S(t)$, $S(t/2)/S(t)$ is distributed as a symmetric beta with parameter $\alpha t/2$. This allows one to apply the dyadic trick.

Perhaps the most basic of all processes is the Poisson point process. These processes are completely described by an increasing sequence of occurrences $0 < X_1 < X_2, \cdots$. In a homogeneous point process of density λ, the inter-occurrence distances are i.i.d. and distributed as exponentials of mean $1/\lambda$. In a nonhomogenoeous Poisson point process of density $\lambda(t)$, a nonnegative function, the integrals $\int_{X_i}^{X_{i+1}} \lambda(t)\,\mathrm{d}t$ are i.i.d. exponential. If $\Lambda(t) = \int_0^t \lambda(t)\,\mathrm{d}t$, then it is easy to see that $X_{i+1}-X_i$ is distributed as $\Lambda^{\mathrm{inv}}(\Lambda(X_i)+E)-X_i$, where E is exponential. We already discussed sampling by thinning (rejection) for this process elsewhere. This process view also yields a simple but not uniformly fast generator for the Poisson distribution: if $E_1, E_2, \ldots$ are i.i.d. exponential random variables, and X is the smallest integer such that

$$\sum_{i=1}^{X+1} E_i > \lambda,$$

then X is Poisson(λ), as we are just counting the number of occurrences of a unit density homogeneous Poisson point process in the interval $[0, \lambda]$. By noting that an exponential is distributed as the logarithm of one over a uniform, we see that we may equivalently generate i.i.d. uniform[0, 1] random variables $U_1, U_2, \ldots$ and let X be the smallest integer such that

$$\prod_{i=1}^{X+1} U_i < \mathrm{e}^{-\lambda}.$$

This is the so-called product method for Poisson random variates.

6 Markov chain methodology

We say that we generate X by the Markov chain method if we can generate a Markov chain X_n, $n \geqslant 1$, with the property that

$$\lim_{n\to\infty} \mathrm{P}\{X_n \leqslant z\} = \mathrm{P}\{X_\infty \leqslant z\},$$

at all z at which the function $\mathrm{P}\{X_\infty \leqslant z\}$ is continuous (here $\leqslant$ is to be taken componentwise). We write $X_n \overset{\mathcal{L}}{\to} X_\infty$, and say that X_n converges in distribution to X_∞. This is a weak notion, as X_n may be discrete and X_∞ continuous. For example, we may partition $\mathbb{R}^d$ into a grid of cells of sides $1/n^{1/(2d)}$, and let p_u be the probability content of cell u. Let X_n be the midpoint of cell u, where cell u is selected with probability p_u. A simple exercise shows that $X_n \overset{\mathcal{L}}{\to} X_\infty$ for any distribution of X_∞. Yet, X_n is discrete, and X_∞ may have any type of distribution. One needs of course a way of computing the p_u's. For this, knowledge of the distribution function F of X_∞ at each point suffices as p_u can be written as a finite linear combination of the values of F at the vertices of the cell u. If X_n needs to have a density, one can define X_n as a uniform variate over the cell u, picked as above. In the latter case, if X_∞ has any density, we have a stronger kind of convergence, namely convergence in total variation (see below), by virtue of the Lebesgue density theorem (Wheeden and Zygmund, 1977; Devroye, 1987b). A good general reference for this section is Häggström (2002).

The paradigm described in the previous paragraph is a special case of what one could call the limit law method. Many processes, such as averages, maxima and so forth, have asymptotic distributions that are severely restricted (e.g., averages of independent random variables have a stable limit law). Markov chains can easily be molded to nearly all distributions we can imagine, hence the emphasis on them as a tool.

In the exact Markov chain method, we can use properties of the Markov chain to generate a random variate X_T, where $X_T \overset{\mathcal{L}}{=} X_\infty$, where we recall that $\overset{\mathcal{L}}{=}$ denotes "is distributed as". There are a certain number of tricks that one can use to define a computable stopping time T. The first one was the so-called coupling from the past method of Propp and Wilson (1996), which has been shown to work well for discrete Markov chains, especially for the generation of random combinatorial objects. For general continuous distributions, a sufficiently general construction of stopping times is still lacking.

6.1 The Metropolis–Hastings chain

The Metropolis–Hastings chain (Metropolis et al., 1953; Hastings, 1970) can be used for the generation of a random variate with an arbitrary density, provided that some care is taken in its choice of Markov transition probabilities. It requires transition probabilities $q(x, y)$ representing the density of Y on $[0, 1]^d$ given $X = x$. Thus, for every fixed $x \in [0, 1]^d$, $q(x, y)$ is a valid density in y from which we can generate random variates at will. The following algorithm produces a chain on random variates $\{X_n, n \geqslant 1\}$, which we call the Metropolis–Hastings chain:

```
set $X \leftarrow x \in [0, 1]^d$
repeat
    generate $U$ uniformly on $[0, 1]$
    generate $Y$ with density $q(X, \cdot)$ on $[0, 1]^d$
    if $U < \frac{f(Y)q(Y,X)}{f(X)q(X,Y)}$ then $X \leftarrow Y$.
```

The limit distribution of X_n is not necessarily a distribution with density f. For example, if $d = 1$, and $q(x, y) = 2\,\mathbf{1}_{(x,y)\in B}$, where $B = [0, 1/2]^2 \cup [1/2, 1]^2$, then if $x < 1/2$ is the starting point, regardless of what f is, $X_n < 1/2$ for all n, and thus, universal convergence is excluded. A sufficient condition for convergence is that for every $x, y \in [0, 1]^d$, $q(x, y) > 0$. Here X_∞ is a random variable with density f and $\leqslant$ is to be taken componentwise in $\mathbb{R}^d$. In particular, in the absence of any information on f, we may well just take $q(x, y) = 1$ for all x, y. We then obtain the following simple version of the chain:

```
set $X \leftarrow x \in [0,1]^d$.
repeat
  generate $U$ uniformly on $[0,1]$
  generate $Y$ uniformly on $[0,1]^d$
  if $U < f(Y)/f(X)$ then $X \leftarrow Y$.
```

We would have obtained the same algorithm if we had the symmetry condition $q(x, y) = q(y, x)$, so we call this the symmetric Metropolis–Hastings chain. The algorithm that uses a uniform Y thus produces a sequence of X's that are selected from among uniforms. In this sense, they are "pure" and amenable to theoretical analysis. In the subsection below, we recall a lower bound for such pure methods.

6.2 The independence sampler

Tierney (1994) proposed $q(x, y) \equiv q(y)$ for all x, y. The independence sampler thus reads:

```
set $X \leftarrow x \in [0,1]^d$
repeat
  generate $U$ uniformly on $[0,1]$
  generate $Y$ with density $q$ on $[0,1]^d$
  if $U < \frac{f(Y)q(X)}{f(X)q(Y)}$ then $X \leftarrow Y$.
```

If $q = f$, then no rejection takes place, and this algorithm produces an i.i.d. sequence of random variates with density f. Otherwise, if $q > 0$ there is convergence (in total variation, see below), and even geometric convergence at the rate $1 - \inf_x q(x)/f(x)$ (Liu, 1996). One thus should try and match q as well as possible to f.

6.3 The discrete Metropolis chain

The chains given above all remain valid if the state space is finite, provided that the density $f(x)$ and conditional density $q(x, y)$ are replaced by a stationary probability π_x and a transition probability $p(x, y)$ (summing to one with respect to y). In the discrete case, there is an important special case. A graph model for the uniform generation of random objects is the following. Let each node in the finite graph (V, E) correspond to an object. Create a connected graph by joining nodes that are near. For example, if a node represents an n-bit vector, then connect it to all nodes at Hamming distance one. The degree $\delta(x)$ of a node x is its number of neighbors, and $N(x)$ is the set of its neighbors. Set

$$q(x, y) = \frac{\mathbf{1}_{y \in N(x)}}{\delta(x)}.$$

If we wish to have a uniform stationary vector, that is, each node x has asymptotic probability $1/|V|$, then the Metropolis–Hastings chain method reduces to this:

```
set $X \leftarrow x \in V$
repeat
   generate $Y$ uniformly in $N(X)$
   generate $U$ uniformly on $[0, 1]$
   if $U < \delta(X)/\delta(Y)$ then $X \leftarrow Y$.
```

If the degrees are the same, then we always accept. Note also that we have a uniform limit only if the graph thus obtained is aperiodic, which in this case is equivalent to asking that it is not bipartite.

In some situations, there is a positive energy function $H(x)$ of the variable x, and the desired asymptotic density is

$$f(x) = C\mathrm{e}^{-H(x)},$$

where C is a normalization constant. As an example, consider x is an n-bit pixel vector for an image, where $H(x)$ takes into account occurrences of patterns in the image. We select $q(x, y)$ as above by defining neighborhoods for all x. As x is a bit vector, neighborhoods can be selected based on Hamming distance, and in that case, $q(x, y) = q(y, x)$ for all x, y. The symmetric Metropolis–Hastings chain then applies (in a discrete version) and we have:

```
set $X \leftarrow x \in V$
repeat
   generate $U$ uniformly on $[0, 1]$
   generate $Y$ uniformly in $N(X)$
   if $U < \exp(H(X) - H(Y))$ then $X \leftarrow Y$.
```

For further discussion and examples see Besag (1986), Geman and Geman (1984) and Geman and McClure (1985).

6.4 Letac's lower bound

Generating a random variate with density f on the real line can always be done by generating a random variate with some density g on $[0, 1]$ via a monotone transformation. Thus, restricting oneself to $[0, 1]$ is reasonable and universal. Assume thus that our random variate X is supported on $[0, 1]$. Letac (1975) showed that any generator which outputs $X \leftarrow U_T$, where T is a stopping time and $U_1, U_2, \ldots$ is an i.i.d. sequence of uniform[0, 1] random variables must satisfy

$$\mathrm{E}\{T\} \geqslant \sup_x f(x).$$

Applying a simple rejection method with the bound $\sup_x f(x)$ requires a number of iterations equal to $\sup_x f(x)$, and is thus optimal. (In fact, two uniforms are consumed per iteration, but there are methods of recirculating the second uniform.) Since some of the symmetric Metropolis–Hastings chains and independence samplers are covered by Letac's model, we note that even if they were stopped at some stopping time T tuned to return an exact random variate at that point, Letac's bound indicates that they cannot do better than the rejection method. There are two caveats though: firstly, the rejection method requires knowledge of $\sup_x f(x)$; secondly, for the Markov chains, we do not know when to stop, and thus have only an approximation. Thus, for fair comparisons, we need lower bounds in terms of approximation errors.

6.5 Rate of convergence

The rate of convergence can be measured by the total variation distance

$$V_n =^{\text{def}} \sup_A \left| \mathrm{P}\{X_n \in A\} - \mathrm{P}\{X_\infty \in A\} \right|,$$

where A ranges over all Borel sets of $\mathbb{R}^d$, and X_∞ is the limit law for X_n. Note Scheffé's identity (Scheffé, 1947)

$$V_n = \frac{1}{2} \int \left| f_n(x) - f(x) \right| \mathrm{d}x,$$

where f_n is the density of X_n (with the above continuous versions of the chains, started at X_0, we know that X_n has a density for $n > 0$).

Under additional restrictions on q, e.g.,

$$\inf_x \inf_y \frac{q(x, y)}{f(x)} > 0,$$

we know that $V_n \leqslant \exp(-\rho n)$ for some $\rho > 0$ (Holden, 1998; Jarner and Hansen, 1998). For more on total variation convergence of Markov chains, see Meyn and Tweedie (1993). Gilks et al. (1995) introduce another method worthy of consideration, the adaptive rejection Metropolis sampler.

6.6 The Metropolis random walk

The Metropolis random walk is the Metropolis chain obtained by setting

$$q(x, y) = q(y, x) = g(y - x)$$

where g is symmetric about 0. On the real line, we could take, e.g., $g(u) = (1/2)\mathrm{e}^{-|u|}$. The algorithm thus becomes:

```
set $X \leftarrow x \in \mathbb{R}^d$
repeat
  generate $U$ uniformly on $[0, 1]$
  generate $Z$ with density $g$
  set $Y \leftarrow X + Z$
  if $U < f(Y)/f(X)$ then $X \leftarrow Y$.
```

If g has support over all of $\mathbb{R}^d$, then the Metropolis random walk converges.

6.7 The Gibbs sampler

Among Markov chains that do not use rejection, we cite the Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990) which uses the following method to generate X_{n+1} given $X_n = (X_{n,1}, \ldots, X_{n,d})$: first generate

$$X_{n+1,1} \sim f(\cdot | X_{n,2}, \ldots, X_{n,d}).$$

Then generate

$$X_{n+1,2} \sim f(\cdot | X_{n+1,1}, X_{n,3}, \ldots, X_{n,d}),$$

and continue on until

$$X_{n+1,d} \sim f(\cdot | X_{n+1,1}, \ldots, X_{n+1,d-1}).$$

Under certain mixing conditions, satisfied if each conditional density is supported on all of the real line, this chain converges. Of course, one needs to be able to generate from all the $(d-1)$-dimensional conditional densities. It is also possible to group variables and generate random vectors at each step, holding all the components not involved in the random vector fixed.

6.8 Universal generators

A universal generator for densities generates a random variate X with density f without any a priori knowledge about f. The existence of such a generator is discussed in this section. It is noteworthy that a universal discrete generator (where X is supported on the positive integers) is easily obtained by inversion (see above). In fact, given any distribution function F on the reals, an approximate inversion method can be obtained by constructing a convergent iterative solution of $U = F(X)$, where U is uniform[0, 1]. For example, binary search could be employed.

Assume that we can compute f as in a black box, and that X is supported on $[0,1]^d$ (see the previous section for a discussion of this assumption). We recall that if (X, Y) is uniformly distributed on the set $A = \{(x, y)\colon x \in [0,1]^d, 0 \leqslant y \leqslant f(x)\}$, then X has density f. Thus, the purpose is to generate a uniform random vector on A. This can be achieved by the Gibbs sampler, starting from any $x \in [0,1]^d$:

```
set $X \leftarrow x \in [0,1]^d$
repeat
  generate $U$ uniformly on $[0,1]$
  set $Y \leftarrow Uf(X)$
  repeat
    generate $X$ uniformly on $[0,1]^d$
  until $f(X) \geqslant Y$.
```

This algorithm is also known as the slice sampler (Swendsen and Wang, 1987; Besag and Green, 1993). In the first step of each iteration, given X, Y is produced with the conditional distribution required, that is, the uniform distribution on $[0, f(X)]$. Given Y, a new X is generated uniformly on the part of $[0, 1]^d$ that is tucked under the curve of f. We chose to do this by rejection, because nothing is known about f. If some information is known about f, serious accelerations are possible. For example, if f is monotone decreasing on $[0, 1]$, binary search may be used to reduce the search region rapidly. If f is strictly unimodal, then a slight adaptation of binary search may help. To illustrate the latter point for $d = 1$, assume that both f and f^{inv} are available in black box format, and that f is supported on the real line. Then the following algorithm always converges, where f_ℓ and f_r denote the left and right solutions of $f(x) = Y$:

```
set $X \leftarrow x \in \mathbb{R}$
repeat
    generate $U$ uniformly on $[0, 1]$
    set $Y \leftarrow Uf(X)$
    generate $X$ uniformly on $[f_\ell(Y), f_r(Y)]$.
```

Note that normalization or multiplicative constants in f and f^{inv} can be dropped altogether.

In any case, convergence theorems for the Gibbs sampler imply that the rejection Markov chain given above produces a sequence of vectors $\{(X_n, Y_n), n \geqslant 1\}$, with the property that (X_n, Y_n) tends in total variation to (X, Y), a random vector with the uniform distribution on A. Therefore, X_n tends in total variation to X, which has density f. It is thus a universal, but approximate, method.

There are stumbling blocks if one wishes to obtain an exact universal method. One which requires the knowledge of the tail function

$$T(t) = \int_{x:f(x)>t} f(x)\,\mathrm{d}x, \quad t > 0,$$

can easily be constructed. We consider the density in slabs of size one and generate a uniform random vector in A slab-wise. Define $p_n = T(n+1) - T(n)$, $n \geqslant 0$, and note that $\sum_n p_n = 1$. Selection of a random slab can be done by inversion on the fly – it does not require storage of the p_n's. Thus, a simple mixture and rejection algorithm would do this:

```
generate $Z \geqslant 0$ with probability distribution $(p_0, p_1, p_2, \ldots)$
repeat
    generate $(X, Y)$ uniformly on $[0, 1]^d \times [0, 1]$
until $Z + Y \leqslant f(X)$
return $X$.
```

6.9 Coupling from the past

Asmussen et al. (1992) and Propp and Wilson (1996) shocked the simulation world by announcing a way of simulating a finite discrete time ergodic Markov chain that produces an output with the stationary distribution. Their method is called perfect sampling, or CFTP, coupling from the past. We shall denote the unique stationary probability vector by $(\pi_1, \ldots, \pi_n)$, where n is the number of states. We reserve t for the time index.

Let ω_t denote a random element at time t, governing transitions from all states. That is, X_{t+1}, the state at time $t+1$, is a function $\varphi(X_t, \omega_t)$, and indeed, we know and design φ and ω_t such that the transition probabilities

$$\mathrm{P}\{X_{t+1} = y | X_t = x\} = q(x, y),$$

for all x, y, as prescribed. The random elements ω_t form an independent sequence in t. Interestingly, the transitions at time t may be coupled. For example, if the Markov chain is a random walk on a cycle graph, then ω_t may just flip a fair coin. If it comes up heads, then all transitions are to the left, and if comes up tails, all transitions are to the right. The coupling is thus up to the designer. In the CFTP method, the sequence ω_t is fixed once and for all, but we need to generate only a finite number of them, luckily, to run the algorithm.

Pick a positive integer T. A basic run of length T consists of n Markov chains, each fired up from a different state for T time units, starting at time $-T$ until time 0, using the random elements ω_t, $-T \leqslant t < 0$. This yields n runs $X_t(i)$, $1 \leqslant i \leqslant n$ (i denotes the starting point), $-T \leqslant t \leqslant 0$, with $X_{-T}(i) = i$. Observe that if two runs collide at a given time t, then they will remain together forever. If $X_0(i)$ is the same for all i, then all runs have collided. In that case, Propp and Wilson showed that $X_0(\cdot)$ has the stationary distribution. If there is no agreement among the $X_0(i)$'s, then replace T by $2T$ and repeat, always doubling the horizon until all $X_0(i)$'s agree. Return the first $X_0(\cdot)$ for which agreement was observed.

It is interesting that coupling in the future does not work. We cannot simply start the n chains $X_0(i) = i$ and let T be the first time when they coalesce, because there are no guarantees that $X_T(\cdot)$ has the stationary distribution. To see this, consider a directed cycle with n vertices, in which state 1 is the only state with a loop (occurring with probability $\varepsilon > 0$). Coalescence can only take place at this state, so $X_T(\cdot) = 1$, no matter what. Yet, the limiting distribution is roughly uniform over the states when $\varepsilon > 0$ is small.

The effort needed may be prohibitive, as the time until agreement in a well designed coupling is roughly related to the mixing time of a Markov chain, so that n times the mixing time provides a rough bound of the computational complexity. Luckily, we have the freedom to design the random elements in such a manner that coalescence can be checked without doing n runs. For example, for a random walk on the chain graph with n vertices, where a loop is added (with probability 1/2) at each state to make the Markov chain aperiodic, we design the ω_t such that no two paths ever cross. Then coalescence can be checked by following the chains started at states 1 and n only. The coupling in ω_t is as follows: draw a uniform integer from $\{-2, -1, 1, 2\}$. If it is -2, then all transitions are to the left, except the transition from state 1, which stays. If it is 2, then all transitions are to the right, except the transition from state n, which stays. If it is -1, then all states stay, except state n, which takes a step to the left. If it is 1, then all states stay, except state 1, which takes a step to the right. As the number of steps until the two extreme runs meet is of the order of n^2, the expected complexity is also of that order. For more on the basics of random walks on graphs and mixing times for Markov chains, see Aldous and Fill (2004).

If the state space is countably infinite or uncountable (say, $\mathbb{R}^d$), then special constructions are needed to make sure that we only have to run finitely many chains, see, e.g., Green and Murdoch (1999), Wilson (2000), Murdoch (2000) or Mira et al. (2001). Most solutions assume the existence of a known dominating density g, with $f \leqslant cg$ for a constant c, with random samples from g needed to move ahead. The resulting algorithms are still inferior to the corresponding rejection method, so further research is needed in this direction.

6.10 Further research

If one accepts approximations with density f_n of the density f, where n is the computational complexity of the approximation, then a study must be made that compares $\int |f_n - f|$ among various methods. For example, f_n could be the nth iterate in the slice sampler, while a competitor could be g_n, the density in the approximate rejection method, which (incorrectly) uses rejection from n (so that g_n is proportional to $\min(f, n)$, and $\int |g_n - f| = 2\int (f - n)_+$). Both methods have similar complexities (linear in n), but a broad comparison of the total variation errors is still lacking. In fact, the rejection method could be made adaptive: whenever an X is generated for which $f(X) > n$, then replace n by $f(X)$ and continue.

Simple exact simulation methods that are based upon iterative non-Markovian schemes could be of interest. Can coupling from the past be extended and simplified?

Acknowledgements

The authors' research was sponsored by NSERC Grant A3456 and FCAR Grant 90-ER-0291.

References

Ahrens, J.H. (1993). Sampling from General Distributions by Suboptimal Division of Domains. *Grazer Mathematische Berichte* 319.

Ahrens, J.H. (1995). A one-table method for sampling from continuous and discrete distributions. *Computing* 54, 127–146.

Ahrens, J.H., Dieter, U. (1974). Computer methods for sampling from gamma, beta, Poisson and binomial distributions. *Computing* 12, 223–246.

Ahrens, J.H., Dieter, U. (1980). Sampling from binomial and Poisson distributions: A method with bounded computation times. *Computing* 25, 193–208.

Ahrens, J.H., Dieter, U. (1982). Generating gamma variates by a modified rejection technique. *Communications of the ACM* 25, 47–54.

Ahrens, J.H., Dieter, U. (1989). An alias method for sampling from the normal distribution. *Computing* 42, 159–170.

Ahrens, J.H., Dieter, U. (1991). A convenient sampling method with bounded computation times for Poisson distributions. In: Nelson, P.R., Dudewicz, E.J., Öztürk, A., van der Meulen, E.C. (Eds.), *The Frontiers of Statistical Computation, Simulation and Modeling*. American Sciences Press, Columbus, OH, pp. 137–149.

Ahrens, J.H., Kohrt, K.D., Dieter, U. (1983). Algorithm 599. Sampling from gamma and Poisson distributions. *ACM Transactions on Mathematical Software* 9, 255–257.

Akhiezer, N.I. (1965). *The Classical Moment Problem*. Hafner, New York.

Aldous, D., Fill, J. (2004). Reversible Markov chains and random walks on graphs. Manuscript. Available at *http://128.32.135.2/users/aldous/RWG/book.html*.

Asmussen, S., Glynn, P., Thorisson, H. (1992). Stationarity detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation* 2, 130–157.

Bailey, R.W. (1994). Polar generation of random variates with the t distribution. *Mathematics of Computation* 62, 779–781.

Besag, J. (1986). On the statistical analysis of dirty pictures (with discussions). *Journal of the Royal Statistical Society B* 48, 259–302.

Besag, J., Green, P.J. (1993). Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society B* 55, 25–37.

Best, D.J. (1978a). A simple algorithm for the computer generation of random samples from a Student's t or symmetric beta distribution. In: Corsten, L.C.A., Hermans, J. (Eds.), *COMPSTAT 1978: Proceedings in Computational Statistics*. Physica Verlag, Wien, Austria, pp. 341–347.

Best, D.J. (1978b). Letter to the editor. *Applied Statistics* 27, 181.

Best, D.J. (1983). A note on gamma variate generators with shape parameter less than unity. *Computing* 30, 185–188.

Bondesson, L. (1982). On simulation from infinitely divisible distributions. *Advances in Applied Probability* 14, 855–869.

Box, G.E.P., Müller, M.E. (1958). A note on the generation of random normal deviates. *Annals of Mathematical Statistics* 29, 610–611.

Caflisch, R.E., Morokoff, W.J., Owen, A.B. (1997). Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension. *Journal of Computational Finance* 1, 27–46.

Chambers, J.M., Mallows, C.L., Stuck, B.W. (1976). A method for simulating stable random variables. *Journal of the American Statistical Association* 71, 340–344.

Chen, H.C., Asau, Y. (1974). On generating random variates from an empirical distribution. *AIIE Transactions* 6, 163–166.

Cheng, R.C.H. (1977). The generation of gamma variables with non-integral shape parameter. *Applied Statistics* 26, 71–75.

Cheng, R.C.H., Feast, G.M. (1979). Some simple gamma variate generators. *Applied Statistics* 28, 290–295.

Cheng, R.C.H., Feast, G.M. (1980). Gamma variate generators with increased shape parameter range. *Communications of the ACM* 23, 389–393.

Devroye, L. (1981a). The series method in random variate generation and its application to the Kolmogorov–Smirnov distribution. *American Journal of Mathematical and Management Sciences* 1, 359–379.

Devroye, L. (1981b). The computer generation of random variables with a given characteristic function. *Computers and Mathematics with Applications* 7, 547–552.

Devroye, L. (1981c). The computer generation of Poisson random variables. *Computing* 26, 197–207.

Devroye, L. (1984a). Methods for generating random variates with Pólya characteristic functions. *Statistics and Probability Letters* 2, 257–261.

Devroye, L. (1984b). Random variate generation for unimodal and monotone densities. *Computing* 32, 43–68.

Devroye, L. (1984c). A simple algorithm for generating random variates with a log-concave density. *Computing* 33, 247–257.

Devroye, L. (1986a). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

Devroye, L. (1986b). An automatic method for generating random variables with a given characteristic function. *SIAM Journal of Applied Mathematics* 46, 698–719.

Devroye, L. (1986c). The analysis of some algorithms for generating random variates with a given hazard rate. *Naval Research Logistics Quarterly* 33, 281–292.

Devroye, L. (1987a). A simple generator for discrete log-concave distributions. *Computing* 39, 87–91.

Devroye, L. (1987b). *A Course in Density Estimation*. Birkhäuser, Boston.

Devroye, L. (1988). Generating sums in constant average time. In: Abrams, M.A., Haigh, P.L., Comfort, J.C. (Eds.), *Proceedings of the 1988 Winter Simulation Conference*. IEEE, San Diego, CA, pp. 425–431.

Devroye, L. (1989). On random variate generation when only moments or Fourier coefficients are known. *Mathematics and Computers in Simulation* 31, 71–89.

Devroye, L. (1990). A note on Linnik's distribution. *Statistics and Probability Letters* 9, 305–306.

Devroye, L. (1991). Algorithms for generating discrete random variables with a given generating function or a given moment sequence. *SIAM Journal on Scientific and Statistical Computing* 12, 107–126.

Devroye, L. (1996). Random variate generation in one line of code. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings 1996 Winter Simulation Conference*. ACM, San Diego, CA, pp. 265–272.

Devroye, L. (1997). Simulating theta random variates. *Statistics and Probability Letters* 31, 2785–2791.

Devroye, L. (2001). Simulating perpetuities. *Methodologies and Computing in Applied Probability* 3, 97–115.

Devroye, L., Neininger, R. (2002). Density approximation and exact simulation of random variables that are solutions of fixed-point equations. *Advances of Applied Probability* 34, 441–468.

Devroye, L., Fill, J., Neininger, R. (2000). Perfect simulation from the quicksort limit distribution. *Electronic Communications in Probability* 5, 95–99.

Dickman, K. (1930). On the frequency of numbers containing prime factors of a certain magnitude. *Arkiv för Mathematik, Astronomi och Fysik* 22 (A), 1014.

Dugué, D., Girault, M. (1955). Fonctions convexes de Pólya. *Publications de l'Institut de Statistique des Universités de Paris* 4, 3–10.

Evans, M., Swartz, T. (1996). Random variable generation using concavity properties of transformed densities. *Journal of Computational and Graphical Statistics* 7, 514–528.

Evans, M., Swartz, T. (2000). *Approximating Integrals via Monte Carlo and Deterministic Methods*. Oxford University Press, Oxford, UK.

Feller, W. (1948). On the Kolmogorov–Smirnov limit theorems for empirical distributions. *Annals of Mathematical Statistics* 19, 177–189.

Fill, J. (1998). An interruptible algorithm for perfect sampling via Markov chains. *The Annals of Applied Probability* 8, 131–162.

Flajolet, P., Saheb, N. (1986). The complexity of generating an exponentially distributed variate. *Journal of Algorithms* 7, 463–488.

Fox, B.L. (1999). *Strategies for Quasi-Monte Carlo*. Kluwer Academic Publishers, Boston, MA.

Gelfand, A.E., Smith, A.F.M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85, 398–409.

Geman, S., Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741.

Geman, S., McClure, D. (1985). Bayesian image analysis: An application to single photon emission tomography. In: *Proceedings of the Statistical Computing Section*. American Statistical Association, Washington, DC, pp. 12–18.

Gilks, W.R., Wild, P. (1992). Adaptive rejection for Gibbs sampling. *Applied Statistics* 41, 337–348.

Gilks, W.R., Best, N.G., Tan, K.K.C. (1995). Adaptive rejection Metropolis sampling. *Applied Statistics* 41, 337–348.

Goldie, C.M., Grübel, R. (1996). Perpetuities with thin tails. *Advances in Applied Probability* 28, 463–480.

Green, P.J., Murdoch, D.J. (1999). Exact sampling for Bayesian inference: Towards general purpose algorithms (with discussion). In: Bernardo, J.M., Berger, J.O., Dawid, A.P., Smith, A.F.M. (Eds.), *Proceedings of the Sixth Valencia International Meeting*. *Bayesian Statistics*, vol. 6. Oxford University Press, Oxford, pp. 301–321.

Häggström, O. (2002). *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, Cambridge, UK.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–110.

Holden, L. (1998). Geometric convergence of the Metropolis–Hastings simulation algorithm. *Statistics and Probability Letters* 39, 371–377.

Hörmann, W. (1993a). The generation of binomial random variates. *Journal of Statistical Computation and Simulation* 46, 101–110.

Hörmann, W. (1993b). The transformed rejection method for generating Poisson random variables. *Insurance: Mathematics and Economics* 12, 39–45.

Hörmann, W. (1994). A universal generator for discrete log-concave distributions. *Computing* 52, 89–96.

Hörmann, W. (1995). A rejection technique for sampling from T-concave distributions. *ACM Transactions on Mathematical Software* 21, 182–193.

Hörmann, W., Derflinger, G. (1994). Universal generators for correlation induction. In: Dutter, R., Grossmann, W. (Eds.), *COMPSTAT 94*. Physica-Verlag, Heidelberg, pp. 52–57.

Hörmann, W., Leydold, J., Derflinger, G. (2004). *Automatic Nonuniform Random Variate Generation*. Springer-Verlag, Berlin.

Huffman, D. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* 40, 1098–1101.

Jackson, D. (1930). *The Theory of Approximation*. *Colloquia Publications*, vol. 11. American Mathematical Society, Providence, RI.

Jarner, S.F., Hansen, E. (1998). Geometric ergodicity of Metropolis algorithms. Department of Statistics, University of Copenhagen.

Johnson, N.L., Kotz, S. (1969). *Distributions in Statistics: Discrete Distributions*. Wiley, New York, NY.

Kachitvichyanukul, V., Schmeiser, B.W. (1988). Binomial random variate generation. *Communications of the ACM* 31, 216–222.

Kachitvichyanukul, V., Schmeiser, B.W. (1989). Algorithm 678: BTPEC: Sampling from the binomial distribution. *ACM Transactions on Mathematical Software* 15, 394–397.

Kanter, M. (1975). Stable densities under change of scale and total variation inequalities. *The Annals of Probability* 3, 697–707.

Keane, M.S., O'Brien, G.L. (1994). A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation* 4, 213–219.

Knuth, D.E., Yao, A.C. (1976). The complexity of nonuniform random number generation. In: Traub, J.E. (Ed.), *Algorithms and Complexity*. Academic Press, New York, NY, pp. 357–428.

Kotz, S., Ostrovskii, I.V. (1996). A mixture representation of the Linnik distribution. *Statistics and Probability Letters* 26, 61–64.

Krein, M.G. (1944). On one extrapolation problem of A.N. Kolmogorov. *Dokl. Akad. Nauk SSSR* 46, 339–342.

Laha, R.G. (1961). On a class of unimodal distributions. *Proceedings of the American Mathematical Society* 12, 181–184.

Le Minh, D. (1988). Generating gamma variates. *ACM Transactions on Mathematical Software* 14, 261–266.

Letac, G. (1975). On building random variables of a given distribution. *The Annals of Probability* 3, 298–306.

Lewis, P.A.W., Shedler, G.S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* 26, 403–413.

Leydold, J. (2000a). Automatic sampling with the ratio-of-uniforms method. *ACM Transactions on Mathematical Software* 26, 78–98.

Leydold, J. (2000b). A note on transformed density rejection. *Computing* 65, 187–192.

Leydold, J. (2001). A simple universal generator for continuous and discrete univariate T-concave distributions. *ACM Transactions on Mathematical Software* 27, 66–82.

Lin, G.D. (1997). On the moment problem. *Statistics and Probability Letters* 35, 85–90.

Linnik, Yu.V. (1962). Linear forms and statistical criteria, I, II. *Selected Translations in Mathematical Statistics and Probability* 3, 1–90.

Liu, J.S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing* 6, 113–119.

Marsaglia, G. (1977). The squeeze method for generating gamma variates. *Computers and Mathematics with Applications* 3, 321–325.

Marsaglia, G., Tsang, W.W. (1984). A fast, easily implemented method for sampling from decreasing or symmetric unimodal density functions. *SIAM Journal on Scientific and Statistical Computing* 5, 349–359.

Marsaglia, G., Tsang, W.W. (2001). A simple method for generating gamma variables. *ACM Transactions on Mathematical Software* 24, 341–350.

Marsaglia, G., Maclaren, M.D., Bray, T.A. (1964). A fast procedure for generating normal random variables. *Communications of the ACM* 7, 4–10.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics* 21, 1087–1091.

Meyn, S.P., Tweedie, R.L. (1993). *Markov Chains and Stochastic Stability*. Springer-Verlag, London.

Mira, A., Moller, J., Roberts, G.O. (2001). Perfect slice samplers. *Journal of the Royal Statistical Society B* 63, 593–606.

Murdoch, D.J. (2000). Exact sampling for Bayesian inference: Unbounded state spaces. In: Madras, N. (Ed.), *Monte Carlo Methods*. *Fields Institute Communications*, vol. 6. American Mathematical Society, Providence, RI, pp. 111–121.

Nelsen, R.B. (1999). *An Introduction to Copulas*. Springer-Verlag, Berlin.

Propp, J.G., Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9, 223–252.

Rachev, S.T. (1991). *Probability Metrics and the Stability of Stochastic Models*. Wiley, New York.

Rényi, A., Szekeres, G. (1967). On the height of trees. *Journal of the Australian Mathematical Society* 7, 497–507.

Rösler, U., Rüschendorf, L. (2001). The contraction method for recursive algorithms. *Algorithmica* 29, 3–33.

Sato, K. (2000). *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, Cambridge.

Scheffé, H. (1947). A useful convergence theorem for probability distributions. *Annals of Mathematical Statistics* 18, 434–458.

Schmeiser, B.W., Lal, R. (1980). Squeeze methods for generating gamma variates. *Journal of the American Statistical Association* 75, 679–682.

Schmeiser, B.W., Babu, A.J.G. (1980). Beta variate generation via exponential majorizing functions. *Operations Research* 28, 917–926.

Stadlober, E. (1988). Sampling from Poisson, binomial and hypergeometric distributions: Ratio of uniforms as a simple fast alternative. Habilitationsschrift, Institute of Statistics, Technical University of Graz, Austria.

Stadlober, E. (1989). Binomial random variate generation: A method based on ratio of uniforms. *American Journal of Mathematical and Management Sciences* 9, 1–20.

Stadlober, E. (1990). The ratio of uniforms approach for generating discrete random variates. *Journal of Computational and Applied Mathematics* 31, 181–189.

Stoyanov, J. (2000). Krein condition in probabilistic moment problems. *Bernoulli* 6, 939–949.

Swendsen, R., Wang, J. (1987). Non-universal critical dynamics in Monte Carlo simulation. *Physical Review Letters* 58, 86–88.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics* 22, 1701–1762.

Ulrich, G. (1984). Computer generation of distributions on the *m*-sphere. *Applied Statistics* 33, 158–163.

Vervaat, W. (1979). On a stochastic difference equation and a representation of non-negative infinitely divisible random variables. *Advances in Applied Probability* 11, 750–783.

von Neumann, J. (1951). Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards Applied Mathematics Series* 12, 36–38, also in: *Collected Works*, vol. 5. Pergamon Press, Oxford, 1963, pp. 768–770.

Walker, A.J. (1974). New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters* 10, 127–128.

Walker, A.J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software* 3, 253–256.

Wheeden, R.L., Zygmund, A. (1977). *Measure and Integral*. Dekker, New York, NY.

Wilson, D.B. (2000). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In: Madras, N. (Ed.), *Monte Carlo Methods*. *Fields Institute Communications*, vol. 6. American Mathematical Society, Providence, RI, pp. 141–176.

Zechner, H., Stadlober, E. (1993). Generating beta variates via patchwork rejection. *Computing* 50, 1–18.

Zolotarev, V.M. (1959). On analytic properties of stable distribution laws. *Selected Translations in Mathematical Statistics and Probability* 1, 207–211.

Zolotarev, V.M. (1966). On the representation of stable laws by integrals. *Selected Translations in Mathematical Statistics and Probability* 6, 84–88.

Zolotarev, V.M. (1981). Integral transformations of distributions and estimates of parameters of multidimensional spherically symmetric stable laws. In: *Contributions to Probability*. Academic Press, New York, pp. 283–305.

Zolotarev, V.M. (1986). *One-Dimensional Stable Distributions*. American Mathematical Society, Providence, RI.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13005-4

Chapter 5

Multivariate Input Processes

Bahar Biller

Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA
E-mail: billerb@andrew.cmu.edu

Soumyadip Ghosh

Mathematical Sciences Department, IBM T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA
E-mail: ghoshs@us.ibm.com

Abstract

Representing uncertainty in a simulation study is referred to as input modeling, and is often characterized as selecting probability distributions to represent the input processes. This is a simple task when the input processes can be represented as sequences of independent random variables with identical distributions. However, dependent and multivariate input processes occur naturally in many service, communications, and manufacturing systems. This chapter focuses on the development of multivariate input models which incorporate the interactions and interdependencies among the inputs for the stochastic simulation of such systems.

1 Introduction

An important step in the design of stochastic simulation is input modeling, which refers to modeling the uncertainty in the input environment of the system being studied. Input modeling is often thought of as limited to selecting appropriate univariate probability distributions to represent the primitive inputs of a simulation, and this would indeed be true if the relevant input processes could be represented as independent sequences of identically distributed random variables. When such univariate models are appropriate, a number of generation procedures and tools that support automated input modeling are available; Chapter 4 provides a good overview.

Often, however, univariate models fail to adequately capture the effect of dependencies between input processes that occur naturally in different forms in many service, communications, and manufacturing systems; Melamed et al. (1992) and Ware et al. (1998) give good examples. The ability to capture these dependencies is crucial because inaccurate input models often lead to performance estimates that are seriously in error. This is illustrated powerfully in a straightforward example provided by Livny et al. (1993), who examine the impact of autocorrelated interarrival times on the mean waiting time of a single-server queue. They simulate an $M/M/1$ queue process with independent sequences of exponential interarrival and service times and observe that the estimated mean waiting time changes dramatically with the introduction of autocorrelation in the interarrival process, showing increases by a factor of two orders of magnitude when the autocorrelation is set high. This shows that the independence assumption could lead to very poor estimates of performance measures in the presence of dependence. It is thus imperative to develop simulation input models that incorporate dependence. Our goal here is to present a coherent narrative of the central principles that underlie many promising methods available for constructing dependent input models.

We consider models that can be broadly classified into two groups. The first group produces a sequence of independent samples of a *random vector* $\mathbf{X} = (X_1, X_2, \ldots, X_d)'$, which is a finite collection of d random components. Each component is a real-valued random variable and is associated with a univariate distribution function called its *marginal* distribution. We associate with the random vector a probability distribution function, called its *joint distribution function*, in the space $\Re^d$. This joint distribution completely defines the stochastic (and hence dependence) properties of $\mathbf{X}$. The second group of input models we consider in this chapter captures dependence that arises between subsequent samples of a sequence. Sequences of samples are frequently indexed over time and stochastic processes that exhibit this *temporal* dependence are called *time series*. In this chapter, we use the term time series to denote a sequence of random variables $\{X_t; t \geqslant 1\}$ indexed by a discrete set $t = 1, 2, \ldots$. For example, the month-to-month order quantities for a product placed by a customer can be considered as a univariate time series. The time-series process is a stochastic process with a probability distribution on the space of all possible path realizations. Such a probability distribution completely describes the dependence properties of the stochastic process $\{X_t; t \geqslant 1\}$. However, almost all statistical and engineering applications specify and analyze time series in terms of the probability distributions of the individual random elements X_t and the *autocorrelation* structure of the sequence, which (refer to Section 4.2) is a measure that tries to capture temporal dependencies in sufficient detail.

Simulation input modeling literature focuses on random vectors and time series as separate cases, although the underlying concepts are often similar. We maintain this distinction while describing the methods, but strive to present them as different avatars of the same central concept. A recent unifying work (Biller and Nelson, 2003) represents random vector and time series as special cases of the more general framework of *multivariate time series*. This denotes a time series where the elements $\mathbf{X}_t$ are finite d-dimensional random vectors with distributions in $\Re^d$, and thus represent input processes where dependence exists both over time and among the components of the input process. For example, a distributor has several warehouses and each places monthly orders for a product. The month-to-month dependence still exists, but there may also be dependence between the orders from different warehouses in the same month if they are able to share inventory or supply the same customers.

A multivariate input modeling procedure can be judged *good* or *effective* based on a number of criteria that can be grouped into two main categories, those that judge its modeling capabilities and the effectiveness of its sampling procedures respectively. Any model needs to perform well in both.

From the modeling point of view, a method should firstly be able to represent a broad class of real-life situations. For example, a procedure that models random vectors to have multivariate- or joint-normal distributions may not be appropriate in situations where the true input process is known to be either multi-modal, nonsymmetric, skewed or heavy-tailed. The goal of representing a wide variety of situations is best achieved when the procedure works directly with a complete characterization of the input process, which would be the joint distribution in the case of random vectors and in the more general case of a multivariate time series a probability distribution on the space of all possible $\Re^d$ sample path realizations of the process. Some random vector modeling procedures indeed work on this principle (Section 2) or with a more modest aim of modeling the random vector as a member of an appropriately chosen family of joint distributions (Section 3).

Modeling by matching the joint distribution, though sound from a modeling power point of view, can be quite restrictive in practice. This is primarily due to the quantity and quality of information needed to adequately fit a joint distribution to the situation at hand. The information needed could be data or expert opinion to estimate the true joint distribution function or the parameters of a chosen model distribution function. In light of this difficulty, most of the focus in input-modeling research has been on the development of methods that match only certain key properties of the input process (Section 4), mainly its marginal distributions and an aptly chosen dependence measure. This approach is not completely general, but one can be reasonably certain that the chosen properties capture the effect of dependence to a satisfactory degree in the model. Moreover, these models are presumably easier to construct from data since either fewer parameters need to be estimated or the estimates needed are easily obtained from available data.

From a sampling point of view, the sampling schema associated with a procedure should be easy to implement on a computer and fast in generating samples. Thus, it should avoid complex, computationally intensive function evaluations, which might be needed in many of the approaches in Sections 2 and 3.

One should additionally be able to easily validate the success of a procedure in representing input dependencies once a sample of data is generated. This statistical model validation is an important aspect of the input model development but it will not be covered here; texts such as Law and Kelton (2000) provide a detailed discussion of issues such as goodness of the fit and validation.

The methods discussed in this chapter are presented in descending order of control we have on the stochastic description of the multivariate input processes. In Sections 2 and 3 this discussion is essentially centered around random vector processes and their joint distributions. Much of the time series modeling literature focuses on processes that are partially specified with distributions for individual elements and autocorrelations. These methods have much in common with random vector methods that work with similar partial specifications and are therefore discussed in a unified manner in Section 4. We conclude with promising areas for input modeling research in Section 5.

2 Constructing full joint distributions

The *joint* (or *multivariate*) *cumulative distribution function* (c.d.f.) of a d-dimensional random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)'$ is a nondecreasing d-dimensional real-valued function F that takes values in $[0, 1]$ and is right-continuous in each argument. The joint distribution function completely characterizes the stochastic behavior of $\mathbf{X}$. Random vector $\mathbf{X}$ is said to have a *joint* (or *multivariate*) *probability density function* (p.d.f.) f, if a nonnegative, integrable function f exists such that

$$F(\mathbf{x}) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} \cdots \int_{-\infty}^{x_d} f(u_1, u_2, \ldots, u_d)\, \mathrm{d}u_1\, \mathrm{d}u_2 \cdots \mathrm{d}u_d.$$

In this section, we present different approaches that construct input models with fully specified joint c.d.f.'s or p.d.f.'s. Although these methods are typically used for constructing random vectors and generating data from the pre-specified joint cumulative distribution functions, they can be applied to time-series processes of fixed lengths as well.

Some methods assume that such a joint distribution is explicitly given (Sections 2.1 and 2.3) while some assume that it is available in a special form (Section 2.2). These requirements entail knowledge of a multidimensional function, which can be a major drawback especially if the dimensions involved are large. Other methods reviewed (Sections 2.4 and 2.5) take the alternative approach of obtaining nonparametric estimations of the joint distribution from available historical data. In particular, Bézier curves (Section 2.4) allow the inclusion of various types of expert opinions into the model development using visualization tools, while kernel density estimators (Section 2.5) provide a generalization and improvement over histograms.

2.1 Acceptance/rejection method

The acceptance/rejection method has been used extensively for generating univariate data, but its impact on multivariate data generation has been muted.

This is partly due to the relatively limited attention researchers have paid to multivariate generation, but also because significant practical difficulties exist in the implementation of the method.

The acceptance/rejection principle has a long history; Marsaglia and Bray (1964) is an early reference. To generate random vector $\mathbf{X}$ from a multivariate joint density function $f(\cdot)$, first a joint density $h(\mathbf{x})$ is selected such that $ch(\mathbf{x})$ dominates $f(\mathbf{x})$, i.e., $ch(\mathbf{x}) \geqslant f(\mathbf{x})$, for any $\mathbf{x}$ in the domain D of the function f, where c is some positive constant. A random vector sample generated from $h(\mathbf{x})$ is then accepted as a sample from $f(\mathbf{x})$ with probability $f(\mathbf{x})/[ch(\mathbf{x})]$. It is straightforward to show that this procedure generates vector samples with density f; a proof of this result is available in Appendix 8A of Law and Kelton (2000).

Two important challenges arise in implementing this method: finding a dominating density h which is easy to sample, and computing the constant $c = \sup_{\mathbf{x}\in D} f(\mathbf{x})/h(\mathbf{x})$. If D is bounded, then one can choose the multivariate uniform density function as the dominating h. Otherwise, a reasonable choice is to let h be the joint distribution of independent random variables with the same marginals as those of f. However, for these choices of h, the constant c (which also represents the expected number of h-samples needed to obtain an f-sample, and hence the average time needed to generate one f-sample) increases rapidly as the dimension increases. A more complicated h might be more efficient, but constructing h, generating data from h and computing c may be more difficult and expensive. Gilks and Wild (1992), Hörmann (1995) and Leydold (1998) suggest the transformed-density-rejection method to construct a dominating function, in which one uses a monotonic function to transform f into a concave function, takes the minimum of several tangent hyper-planes, and then transforms it back into the original scale.

The choice of the dominating function h is by no means a trivial one, and one needs to exercise great care. Lack of a codified procedure to obtain efficiently sampled dominating functions makes this method unappealing even in cases of moderate dimensional random vectors.

2.2 Conditional distributions

These methods factorize the joint distribution into a set of conditional and marginal distributions that easily yields samples with the desired joint distributional properties. The key idea is to reduce the problem of generating a d-dimensional random vector into a series of smaller multivariate, and often d univariate, generation problems. Thus, these methods can utilize the vast body of techniques available for univariate generation problems.

There are a large number of ways in which joint distributions can be specified in terms of conditional and marginal distributions, and such a specification can arise naturally in many systems through its dynamics. Gelman and Speed (1993) provide a discussion of the combinations of marginal and conditional distributions that ensure the existence of a (unique) joint distribution. We shall outline two such approaches that we feel are attractive from a random vector generation point of view.

The first approach (Rosenblatt, 1952) factorizes the joint density function $f(\mathbf{x})$ of the d-dimensional random vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)'$ into a marginal and $(d-1)$ conditional distributions as follows:

$$f(\mathbf{x}) = f_{X_1}(x_1) \prod_{m=2}^{d} f_{X_m|X_1,\ldots,X_{m-1}}(x_m|x_1, \ldots, x_{m-1}), \tag{1}$$

where f_{X_1} is the marginal density of the first component X_1 and $f_{X_m|X_1,\ldots,X_{m-1}}$ is the conditional density of the mth component X_m, given the first $m-1$ components $X_1, \ldots, X_{m-1}$. Any such set of conditional distributions always defines a valid joint distribution via (1). Notice that as long as the structure of the factorization remains the same, the order of the conditioning has no impact on the resulting input model: the first factor on the right-hand side of the factorization might as well correspond to the dth component.

Using the factorization (1), random vector $\mathbf{X}$ can be sampled by first generating $X_1 = x_1$ from the marginal distribution f_{X_1} of X_1, then generating $X_2 = x_2$ from the conditional distribution of X_2 given $X_1 = x_1$, $f_{X_2|X_1}$, and so forth through the d components. Despite its simple logic, this method may be difficult to apply as conditional distributions are not easy to derive except for some special cases. Such easily factored multivariate distributions include the multivariate normal (Tong, 1990), the Cauchy and the Burr (Johnson and Kotz, 1972). In cases where the joint distribution is not known, the conditional distributions have to be obtained either from an estimate of the joint distribution or estimated directly.

Another class of conditional distribution based methods that has drawn some academic interest in recent years can be of potential interest to the simulation practitioner. Arnold et al. (2001) provide a good survey of this approach, where the joint distribution is specified in terms of the conditional distribution of the mth component given all the other components $X_1, \ldots, X_{m-1}, X_{m+1}, \ldots, X_d$, $m = 1, \ldots, d$. We denote the associated conditional distribution by $F_{m|-m}(x_m|x_1, \ldots, x_{m-1}, x_{m+1}, \ldots, x_d)$. The sample generation procedure is simple. The $(n+1)$st sample $\mathbf{x}^{n+1}$ is obtained using $\mathbf{x}^n$ of the previous step as follows:

(i) x_1^{n+1} is sampled from $F_{1|-1}(\cdot|x_2^n, \ldots, x_d^n)$.
(ii) For $m = 2, \ldots, d-1$, x_m^{n+1} is sampled from $F_{m|-m}(\cdot|x_1^{n+1}, \ldots, x_{m-1}^{n+1}, x_{m+1}^n, \ldots, x_d^n)$.
(iii) x_d^{n+1} is sampled from $F_{d|-d}(\cdot|x_1^{n+1}, \ldots, x_{d-1}^{n+1})$.

This procedure is known as the Gibbs sampling technique (Geman and Geman, 1984) and is primarily driven by the theory of Markov chains. Under fairly reasonable conditions, the joint distribution of the sample points $\mathbf{x}^n$, $n \geqslant 1$, can be shown to converge to a limiting stationary joint distribution geometrically fast (under a certain norm). This limiting distribution is unique if all of the conditional distributions have the same support (i.e., the subset over which the functions are nonzero in $\Re$). However, the convergence to a limiting distribution is not guaranteed for any arbitrary set of conditional distributions, unlike the earlier approach, where the factorization (1) ensures the correct joint distribution produced. Gelman and Speed (1993) discuss sufficient conditions for the convergence in the bivariate case. Although compatible conditional distributions can be chosen with relative ease in lower dimensions, this problem will likely become intractable in higher dimensions.

Independent of the approach used, it gets harder to derive the conditional distributions from a joint distribution as the dimension of the input process increases. Additionally, conditional distributions are hard to understand in higher dimensions; therefore using expert opinion in the context of these methods with increasing dimensions will be difficult. These issues limit the usage of this method in higher dimensional random vector sampling beyond special cases where the conditional distributions arise naturally.

2.3 Method of copulas

The distinguishing feature of the method of this section is its use of a family of distributions called *copulas*. A copula may be thought of in two equivalent ways: as a function that maps points in the unit hypercube in $\Re^d$ to values in the unit interval or as a multivariate joint distribution function with standard uniform marginal distributions. Sklar (1959) shows that every joint distribution H with marginals F_i, $i = 1, \dots, d$, can be written as

$$H(x_1, x_2, \dots, x_d) = C\bigl(F_1(x_1), F_2(x_2), \dots, F_d(x_d)\bigr), \tag{2}$$

where C is a copula that is uniquely defined if the marginals F_i are continuous, and is unique to an equivalence class in the case of discrete marginals. In either case, C can be interpreted as the dependence structure of H. Thus, we can transform the problem of estimating the joint distribution from a given data set to that of estimating a function in the unit hypercube. The copula can be determined empirically or can belong to an appropriately chosen parametric family. However, the central issue of estimating a function in d variates remains, and hence these methods can very often be haunted by the curse of dimensionality.

The main advantage of a copula is that it remains invariant under strictly increasing transformations of its component variables, simplifying the random vector generation. To sample a random vector $\mathbf{X} = (X_1, X_2, \dots, X_d)'$ with joint distribution H as defined in (2), we first generate a multivariate uniform vector $\mathbf{U} = (U_1, U_2, \dots, U_d)'$ from copula C and then set the random vector $\mathbf{X}$ to be $(F_1^{-1}(U_1), \dots, F_d^{-1}(U_d))'$, where $F_i^{-1}(u) = \inf\{x\colon F_i(x) \geqslant u\}$ is the generalized inverse of F_i. Thus, copulas provide an easy method to model and generate random vectors when the modeler believes that the dependence between the components of the random vector can be expressed independently of the marginal distributions. Later in Sections 4.3.2 and 4.3.3, we shall discuss the methods that use copulas with a special structure as a base for a fast sampling procedure that match partial specifications of random vectors.

Normal copulas are commonly used in the literature for summarizing the dependence structure of a random vector since one can calibrate them to yield pre-specified pair-wise product-moment or rank correlations (Section 4.1); see Clemen and Reilly (1999) for an example. This kind of dependence modeling is also becoming increasingly popular in cases where linear correlation falls short of capturing complex interactions and interdependencies among the model inputs. Further, numerous other parametric families of copulas can be coupled with arbitrary marginal distributions without worries about consistency. Joe (1997) and Nelsen (1999) provide a survey of some extensively studied parametric families of copulas.

2.4 Bézier distributions

This subsection presents a graphical, interactive technique for modeling simulation input processes whose distributions are based on Bézier curves used extensively in computer graphics to approximate smooth functions on bounded intervals. The distinguishing feature of this type of modeling is that since low dimensional Bézier curves can be easily visually modified, various types of expert opinions can be incorporated into the model development using visualization tools. Wagner and Wilson (1995, 1996) develop such tools for the univariate and bivariate cases and show that these are quite effective, especially when few historical data points are available from the existing system. This method is limited to the estimation of continuous joint distributions with bounded support.

We present the key idea behind the construction of Bézier distributions considering the simplest case of representing a univariate, independent process with a marginal distribution denoted by F. To estimate the Bézier approximation $\widehat{F}$ for the true distribution function F, we first define a Bézier curve in two dimensions via the parametric bivariate function

$$\begin{aligned}\mathbf{P}(t) &= \left[P_{\mathbf{x}_1}(t; n, \mathbf{x}_1), P_{\mathbf{x}_2}(t; n, \mathbf{x}_2)\right] \\ &= \sum_{i=0}^{n} B_{n,i}(t)\mathbf{p}_i,\end{aligned}$$

where $t \in [0,1]$, $\mathbf{x}_1 = (x_{10}, x_{11}, \ldots, x_{1n})'$, $\mathbf{x}_2 = (x_{20}, x_{21}, \ldots, x_{2n})'$ and $\{\mathbf{p}_i = (x_{1i}, x_{2i})'; i = 0, 1, \ldots, n\}$ are $n+1$ bivariate control points. The control points can be thought of as magnets attracting the curve $\mathbf{P}$ to themselves. The attraction is quantified in terms of the Bernstein polynomials $\{B_{n,i}(t); i = 0, 1, \ldots, n\}$ (Farin, 1990). The attraction from the ith control point is the strongest at i/n, where $B_{n,i}(t)$ attains its maximum value. The curve passes through $\mathbf{p}_i$ if $B_{n,i} = 1$, which is true for the endpoints $\mathbf{p}_0$ and $\mathbf{p}_n$.

We can now define the Bézier approximation $\widehat{F}$ for F via the curve $\mathbf{P}$ as

$$x(t) = P_{\mathbf{x}_1}(t; n, \mathbf{x}_1) = \sum_{i=0}^{n} B_{n,i}(t) x_{1i},$$

$$\widehat{F}\bigl[x(t)\bigr] = P_{\mathbf{x}_2}(t; n, \mathbf{x}_2) = \sum_{i=0}^{n} B_{n,i}(t) x_{2i}$$

for all $t \in [0, 1]$. Further, the density function estimate $\hat{f}$ is given parametrically for $x(t)$ as

$$\hat{f}\bigl[x(t)\bigr] = \frac{\sum_{i=0}^{n-1} B_{n-1,i}(t)\Delta x_{1i}}{\sum_{i=0}^{n-1} B_{n-1,i}(t)\Delta x_{2i}}$$

for all $t \in [0, 1]$, where $\Delta x_{ji} = x_{j(i+1)} - x_{ji}$ for $i = 0, 1, \ldots, n-1$ and $j = 1, 2$. It is straightforward to generate a Bézier random variate with this density via inversion. After a uniformly distributed random number $u \in [0, 1]$ is obtained, the equality $P_{\mathbf{x}_2}(t_u; n, \mathbf{x}_2) = u$ is solved for $t_u \in [0, 1]$ and the random variate x is set to $P_{\mathbf{x}_1}(t_u; n, \mathbf{x}_1)$.

Wagner and Wilson (1995) provide two approaches for obtaining the Bézier density estimate $\hat{f}$ when the functional form of the true distribution is not known. The first approach finds a Bézier approximation whose distributional properties match expert opinion using a visualization software. This software allows the user to choose appropriate control points, and then makes sure that $\hat{f}$ satisfies the density validity conditions. The second approach is used when historical data are available. An empirical distribution F_n can be constructed from the data sample and one can then minimize the distance between F_n and $\widehat{F}$ in any functional distance norm subject to the constraints that $\widehat{F}$ has the extreme values 0 and 1, and that $\hat{f}$, the density function defined above, is nonnegative.

The extension of the key idea to the characterization of the multivariate case is straightforward. The parametric function $\mathbf{P}$ in $d + 1$ dimensions,

$$\begin{aligned}\mathbf{P}(t) = \bigl[&P_{\mathbf{x}_1}(t; n, \mathbf{x}_1), P_{\mathbf{x}_2}(t; n, \mathbf{x}_2), \ldots, \\ &P_{\mathbf{x}_d}(t; n, \mathbf{x}_d), P_{\mathbf{x}_{d+1}}(t; n, \mathbf{x}_{d+1})\bigr],\end{aligned}$$

is used to derive a d-dimensional joint distribution function and the density function estimate can again be obtained in several ways. However, parameter estimation would become increasingly difficult. Using visualization tools becomes noticeably harder even in the bivariate case (Wagner and Wilson, 1996) and can be ruled out in higher dimensions. Wagner and Wilson (1996) suggest some simple easily solvable optimization formulations to obtain a proper fit in the bivariate case. These can be extended to higher dimensions, but care must be given to the formulation to ensure feasibility and computational ease. Nevertheless, to the best of our knowledge, the extension of these methods to higher-dimensional generation has not been tried and it deserves attention.

2.5 Kernel density estimation

These methods estimate joint densities using kernel density estimates. They are also well suited for efficient multivariate generation in cases where the kernel estimators are carefully chosen. We note that generating random variates from a density estimate by using kernel estimates is the same as the smoothed bootstrap, the subject of Chapter 14.

The literature credits Rosenblatt (1956) for proposing this method and Parzen (1962) for providing an early framework. We refer the reader to Devroye and Györfi (1985) and Silverman (1982) for a good introduction. These methods do not seem to be very popular in the simulation input modeling literature, and this is perhaps due to the fact that selecting the right parameters to obtain good estimates of joint distributions with these methods can increasingly be an art. Some recent attention to these methods, however, suggest that they have the potential to perform very well for problems with moderate dimensions.

The kernel density estimate of an unknown d-dimensional joint density function f is obtained from an observed d-dimensional data sample of size n by averaging an appropriately scaled *noise* density centered around each of the data points. (Although we refer here to density estimates, this procedure can be applied to estimate c.d.f.'s using kernel distributions.) The noise is chosen to be a multivariate random vector whose d-variate probability density function is called the *kernel*, denoted by $K_{\mathbf{H}}$ and defined as

$$K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K\bigl(\mathbf{H}^{-1/2}\mathbf{x}\bigr),$$

where $\mathbf{H}$ is a symmetric positive definite $d \times d$ matrix called the bandwidth matrix. The density function K is assumed to be symmetric around the origin, and we shall henceforth consider this condition to be a part of the definition of the kernel density function. In its most general form, the d-dimensional kernel density estimator is (Deheuvels, 1977)

$$\hat{f}_n(\mathbf{x}; \mathbf{H}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{X}_i).$$

This probability density function is the equiprobable mixture of n noise distributions, each centered around one of the data points. Sampling from this density estimate is achieved by resampling from the original n data points, each being equally likely, and then adding some noise to the selected data point in the form of a sample from the scaled kernel $K_{\mathbf{H}}$ associated with it. Notice that the generation procedure does not need an explicit calculation of the kernel density estimate.

Under certain conditions on the bandwidth parameter $\mathbf{H}$ and the target p.d.f. f, the kernel density estimate $\hat{f}_n$ converges asymptotically to the true density in the integrated mean squared error sense as n grows. Thus, the generation procedure produces samples asymptotically consistent with f, and the kernel density and the bandwidth parameters that minimize the integrated mean squared error is the best choice for the given n-sized sample. However, the selection of such a pair has its challenges in practice. In the univariate case, where the minimization problem has been analytically solved (Wand and Jones, 1995), it is often better from a generation-effort perspective to choose a near-optimal kernel that is straightforward to sample. Although the literature sees a lot of attention paid to this issue there is no generally accepted procedure for specifying the bandwith and kernel parameters. We refer the reader to Wand and Jones (1995) for a good overview and Silverman (1982) for quick-and-dirty estimates of these parameters. These estimates can be far from optimal for many distributions. In such a case, Hörmann et al. (2001) suggest the use of the normal kernel together with a variance-correction algorithm. It is based on the fact that fitting a multivariate normal distribution is optimal if the unknown distribution is normal (with $|\mathbf{H}| \to \infty$), and naive resampling is optimal if the unknown distribution is not continuous (with $|\mathbf{H}| = 0$). In other cases, it would intuitively seem that intermediate values of $|\mathbf{H}|$ lead to a better approximation of the unknown distribution. So, even if the guess of $|\mathbf{H}|$ is far from optimal, it is still very likely that it is better than using naive resampling or fitting the normal distribution for most continuous distributions.

Although the asymptotic approximation properties of the kernel density estimation procedure are theoretically well understood as compared to many other density estimation procedures, the problem of selecting an efficient pair of kernel density and bandwidth parameters in the d-dimensional case remains to be addressed. Preliminary evidence indicates that the method performs satisfactorily in moderate dimensional input processes.

3 Parametric families of joint distributions

As underscored in the previous section, the quantity and the quality of expert opinion or data needed to satisfactorily model a multivariate input process with a fully specified joint distribution might turn the input model development into a formidable task even in moderate dimensions. A classical approach around this problem is to assume a parameterized form for the unknown joint distribution and then determine the parameter values based on any available expert opinion or historical data. However, many of the standard distributions available in literature are insufficient for Monte Carlo applications. Johnson (1987) provides a comprehensive list of these insufficiencies, where he notes that often these families are insufficiently flexible and require formidable computational power to fit and generate samples. These limitations should not be viewed as grounds to abandon these specific distributions entirely, and clearly ample opportunity exists to make significant contributions to this approach.

Many of the parametric families of the multivariate distributions are tied directly to the multivariate normal distribution, which has played a dominant role in both theoretical and applied statistics since the time of Laplace. The normal curve cannot provide adequate representation for many of the input processes encountered in practice, and there exists a significant body of work exploring reasonable departures from the multivariate normal distribution. For example, *(elliptically) symmetric distributions* generalize the symmetric form of the multivariate normal distribution to distributions that possess (elliptically) symmetric contours, while maintaining the advantage of being easy to sample. Fang et al. (1990), Joe (1997) and Johnson (1987) provide a good introduction to the properties of symmetric distributions.

The constraint that the underlying joint distributions possess symmetric contours is also found to be quite restrictive in practice. A class of distributions called the systems of skew frequency curves has been developed to overcome the limitation of the multivariate normal distribution in representing skewed processes. Recall that the probability density function of a multivariate normal random variable is completely determined by the first two moments; therefore, it is not possible to have a skewness factor that could have an impact on the shape of the distributions. The systems of skew frequency curves are constructed from a multivariate normal random vector by applying component-wise transformations and include the t-distribution, the exponential power distribution, and the log-normal distribution. Similar distributions in common use are the Pearson-type distributions proposed by Pearson (1895) and Charlier (1906), and those proposed by Edgeworth (1898), who uses transformations which can be represented by polynomials.

In the last few years, a translation system developed by Johnson (1949a) has been one of the most popular flexible system of distributions used in simulation applications (e.g., see Stanfield et al., 1996; Mirka et al., 2000; Biller and Nelson, 2003). Johnson (1987) is a good source of information on many members of the so-called Johnson translation system. We start with the univariate system, for which a random variable X is defined by a cumulative distribution function of the form

$$F(X) = \Phi\left\{\gamma + \delta g\left[\frac{X-\xi}{\lambda}\right]\right\}, \tag{3}$$

where γ and δ are shape parameters, ξ is a location parameter, λ is a scale parameter, Φ is the standard univariate normal distribution function and $g(\cdot)$ is one of the following transformations:

$$g(y) = \begin{cases} \log(y) & \text{for the } S_{\mathrm{L}} \text{ (log-normal) family,} \\ \log\bigl(y + \sqrt{y^2+1}\bigr) & \text{for the } S_{\mathrm{U}} \text{ (unbounded) family,} \\ \log\bigl(\frac{y}{1-y}\bigr) & \text{for the } S_{\mathrm{B}} \text{ (bounded) family,} \\ y & \text{for the } S_{\mathrm{N}} \text{ (normal) family.} \end{cases} \tag{4}$$

There is a unique family (choice of g) for each feasible combination of the skewness and the kurtosis that determine the parameters γ and δ. Any mean and (positive) variance can be attained by a g in (4) by manipulating the parameters λ and ξ. Within each family, a distribution is completely specified by the values of the parameters $[\gamma, \delta, \lambda, \xi]$ and the range of X depends on the family of interest. A detailed illustration for the shapes of the Johnson-type probability density functions can be found in Johnson (1987).

The multivariate version of this system is obtained by using the following d-dimensional normalizing translation (Johnson, 1949b):

$$\mathbf{Z} = \boldsymbol{\gamma} + \boldsymbol{\delta}\mathbf{g}\big[\boldsymbol{\lambda}^{-1}(\mathbf{X} - \boldsymbol{\xi})\big] \sim \mathrm{N}_d(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{Z}}), \tag{5}$$

where $\mathbf{Z}$ is a d-dimensional normal random variable with mean $\mathbf{0}$ and a $d \times d$ covariance matrix $\boldsymbol{\Sigma}_{\mathbf{Z}}$, $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_d)'$, $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_d)'$ are the d-dimensional vectors of shape and location parameters, $\boldsymbol{\delta} = \mathrm{diag}(\delta_1, \delta_2, \ldots, \delta_d)$ and $\boldsymbol{\lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ are the diagonal matrices whose entries are the shape and scale parameters and the transformation $\mathbf{g}(\cdot)$ is defined as

$$\mathbf{g}\big[(y_1, y_2, \ldots, y_d)\big] = \big[g_1(y_1), g_2(y_2), \ldots, g_d(y_d)\big],$$

where $g_i(y_i)$, $i = 1, 2, \ldots, d$, are defined as in (4). This characterization ensures that the marginal distribution of each X_i is a univariate Johnson distribution.

Modeling using the Johnson family of distributions can be advantageous since they have the ability to represent any set of (finite) first four moments. Moreover, simulation output performance measures are other insensitive to the specific input distribution chosen when enough moments of the distribution are correctly captured. Examples include diffusion approximations and many queuing applications (Burman, 1981). Moreover, the Johnson translation system represents a wide variety of unimodal and bimodal distributional shapes, though shapes with three or more nodes cannot be represented. Further, it allows for the marginal distributions to be different, while most parametric families require that they be from the same family.

The distributions of the Johnson translation system are easy to sample using the normalizing translation defined in (5) once the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{Z}}$ of the multivariate normal random variable $\mathbf{Z}$, and the marginal distribution parameters $\mathbf{g}$, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$ are given. Estimating these distribution parameters from available data can however pose significant challenges. The marginal parameters can be estimated from the corresponding univariate marginal information (DeBrota et al. (1988) provide an efficient numerical procedure), but estimating the normal covariance matrix $\boldsymbol{\Sigma}_{\mathbf{Z}}$, for instance from the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$ of the Johnson random variable $\mathbf{X}$, can be tricky. If $\boldsymbol{\Sigma}_{\mathbf{Z}}$ were being estimated from $\boldsymbol{\Sigma}_{\mathbf{X}}$, then Theorems 1 and 2 in Cario and Nelson (1997) show that under mild conditions on the marginals the input correlation $\boldsymbol{\Sigma}_{\mathbf{X}}(i, j)$ is a continuous nondecreasing function of the base correlation $\boldsymbol{\Sigma}_{\mathbf{Z}}(i, j)$ and does not depend on the other base correlations. This suggests that a straightforward numerical search procedure should yield the correct $\boldsymbol{\Sigma}_{\mathbf{Z}}$ for the given $\boldsymbol{\Sigma}_{\mathbf{X}}$. Such a procedure might however not always produce positive semidefinite estimates for $\boldsymbol{\Sigma}_{\mathbf{Z}}$ (Ghosh and Henderson, 2002a), a necessary (and sufficient) condition for a matrix to be a multivariate normal covariance matrix.

Stanfield et al. (1996) suggest a Johnson-type marginal-based data-generation procedure that does not need to determine the covariance matrix of the multivariate normal random vector. To generate data for a random vector to match mean $\boldsymbol{\mu}_{\mathbf{X}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$, the authors first define a random vector $\mathbf{Y}$ whose components are independent standardized Johnson variates, i.e., $\mathrm{E}(\mathbf{Y}_i) = 0$ and $\mathrm{Var}(\mathbf{Y}_i) = 1$ and $\mathrm{Cov}(\mathbf{Y}_i, \mathbf{Y}_j) = 0$ for $i, j = 1, 2, \ldots, d$. Let $\mathbf{L}_{\mathbf{X}} = \boldsymbol{\Sigma}_{\mathbf{X}}^{1/2}$ correspond to the lower triangular Cholesky-decomposition factor of the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$ and $\boldsymbol{\sigma}_{\mathbf{X}}$ be a diagonal matrix whose entries are the standard deviations of the components of $\mathbf{X}$. Then, random vector $\mathbf{X} = \boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\sigma}_{\mathbf{X}}\mathbf{L}_{\mathbf{X}}\mathbf{Y}$ has desired mean $\boldsymbol{\mu}_{\mathbf{X}}$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$. Further, Stanfield et al. (1996) give expressions that relate the skewness and kurtosis values of $\mathbf{X}$ to those of $\mathbf{Y}$. Thus, one can start with the right parameter values for $\mathbf{Y}$ to obtain the desired first four moments in $\mathbf{X}$. The computational effort needed in finding the right parameters is relatively little. Hence, it is an attractive generation method for obtaining samples with given mean and covariance matrix. However, their approach has some limitations. First, since the marginals do not in general survive the transformation from $\mathbf{Y}$ to $\mathbf{X}$, the exact expressions and properties of the form of the joint c.d.f. and marginal distributions of $\mathbf{X}$ are not clear. Second, they note that although any given skewness factor can be achieved for the standardized Johnson components Y_i, $i = 1, 2, \ldots, d$, not all possible values of the kurtosis can be achieved. Moreover, the order in which the component parameters are determined can affect the kurtosis determination problem.

Flexible families of distributions are, of course, not limited to the Johnson translation system. Johnson et al. (1997) and Kotz et al. (2000) are good sources for multivariate discrete distributions and multivariate continuous distributions, respectively. Other distributions that have been given particular attention in the literature include Tukey's g and h transformations (Johnson, 1987), the multivariate Pearson (Parrish, 1990), and the four-parameter families designed explicitly for use in simulation experiments by Ramberg and Schmeiser (1974) and Schmeiser and Deutsch (1977). The last two families of distributions are particularly easy to use, but the one-to-one relationship between the distribution parameters and the moments is lost. They also fall short in capturing distributional characteristics such as bimodality and heavy tails.

4 Constructing partially specified joint distributions

So far, we have discussed the methods that work either with completely specified c.d.f.'s or with estimates picked from special parametric families of distributions. We have seen that both approaches can face severe practical limitations. In this section, we discuss the construction of input models that match partial specifications of the input process. Here, the input process is commonly specified in terms of marginal distributions of its components and a measure of dependence. An argument in support of modeling using only marginals and a measure of dependence relates to the use of diffusion approximations for modeling stochastic systems. In many cases the limiting diffusions depend only on the first two moments of the input distributions. Performance measures in many queuing applications (Burman, 1981) can be insensitive to moments higher than the second. Therefore, there is some insensitivity in performance measures computed from these models to the exact form of the input distributions. In general then, if this form of insensitivity is present in a model, then the approach discussed here is quite reasonable.

The methods of this section aim, rather successfully, for efficient input model development (model fitting and variate generation) in higher dimensions in return for a compromise on the modeling front. The hope is that a proper specification can capture the essence of the dependence between the components while sparing the practitioner the often arduous task of trying to estimate the full joint distribution. However, partial characterization may not necessarily uniquely or even correctly specify a joint distribution, and more than one, or sometimes no, joint distribution can be defined that satisfy the desired characterization. We consider a partial specification as *feasible* if a multivariate process (or equivalently a probability distribution in the appropriate space), which has the specified properties, exists. The methods presented in this section should ideally be guaranteed to work with any feasible partial specification.

The most commonly specified dependence measure quantifies linear correlation between the components, as correlation matrices for random vectors and a series of correlation matrices called the correlogram for time series. Section 4.1 reviews some relevant properties of correlation and other measures of dependence used in the simulation input model development. Time series modeling literature focuses primarily on processes that are partially specified with distributions for individual elements and autocorrelations. Therefore, we provide a brief overview of basic time series notations and some important models in Section 4.2. In the remainder of the section, we present key approaches that model and generate input processes (random vectors and time series) specified with almost any set of arbitrary marginal distributions and correlation matrices.

4.1 Measures of dependence

Most of the dependence measures we discuss in this section are *pair-wise* measures, in that they are used to quantify the dependence between pairs of components. We refer the reader to Nelsen (1999) for a complete treatment of other dependence measures.

The most widely used and understood measure in engineering applications is the *product-moment covariance*. The (i, j)th product-moment covariance between random variables X_i and X_j (whose variances are necessarily finite) is defined as

$$\mathrm{Cov}(X_i, X_j) = \mathrm{E}\big[(X_i - \mu_i)(X_j - \mu_j)\big] \quad \text{for } 1 \leqslant i \leqslant j \leqslant d,$$

where $\mathrm{E}(X_i) = \mu_i$ and $\mathrm{E}(X_j) = \mu_j$. A related measure is the product-moment correlation (the sample analog is called the Pearson correlation) of X_i and X_j, defined as

$$\rho(i, j) = \mathrm{Cor}(X_i, X_j) = \frac{\mathrm{Cov}(X_i, X_j)}{\sigma_i \sigma_j}, \tag{6}$$

where $\mathrm{Var}(X_i) = \sigma_i^2$ and $\mathrm{Var}(X_j) = \sigma_j^2$. Correlation values, limited to be in a subset of $[-1, 1]$, are preferred over covariances because of their independence from scale parameters of the marginal distributions of X_i and X_j. We shall discuss correlations in the sequel, but each measure can be recovered from the other given the marginal distributions and hence can be treated equivalently.

Product-moment correlation is a measure of *linear* dependence between two components, in that it takes on the maximum magnitude of 1 if and only if a linear relationship exists between the components. This can be a modeling limitation, and can be especially disadvantageous in cases where strong nonlinear dependencies exist since the product-moment correlation value fails to register this.

Some efficient procedures try to incorporate additional dependence information beyond correlations. The algorithm developed by Schmeiser and Lal (1982) is worth mentioning here: it models and constructs bivariate gamma processes using specified correlations, and can also incorporate regression curves of the form $\mathrm{E}[X_1|X_2 = x_2]$ and $\mathrm{E}[X_2|X_1 = x_1]$. Unfortunately, this algorithm relies heavily on the properties of the gamma random variables, and a generalization to other marginal distributions is not obvious.

The correlations (covariances) between the components of a random vector $\mathbf{X}$ are collectively represented by the *correlation (covariance) matrix*. The (i, j)th element of this matrix represents the correlation between components X_i and X_j. By definition, the correlation matrix of any random vector is symmetric, positive semidefinite, and has unit diagonal elements. We refer the reader to Ghosh and Henderson (2002a) for a discussion on the implications of the positive semidefiniteness of a correlation matrix. Biller and Nelson (2005) establish similar requirements for correlograms of time-series processes. However, positive semidefiniteness is not sufficient to guarantee feasibility of a marginal and covariance (or correlation) matrix specification. The marginal distributions also play a crucial role in determining the feasibility of correlation matrices for random vectors and correlograms for multivariate time series. As an example, consider a bivariate input process with a standard exponential and a standard uniform as marginals. The magnitude of correlation between them cannot exceed 0.866. Indeed, a correlation value of 1 would imply a linear relationship between the two, which clearly cannot be the case. In general then, the set of feasible correlation matrices is a strict subset of the set of correlation matrices.

The definition of the product-moment correlation in (6) implies that it is defined only when the variances of the components of the input process are finite. Thus, it is not an appropriate measure of dependence for very heavy-tailed inputs. A dependence measure that avoids this pitfall is the *Spearman rank correlation* (see Jong-Dev, 1984), defined as

$$r(i,j)=\frac{\mathrm{E}[F_i(X_i)F_j(X_j)]-\mathrm{E}[F_i(X_i)]\,\mathrm{E}[F_j(X_j)]}{\sqrt{\mathrm{Var}(F_i(X_i))\,\mathrm{Var}(F_j(X_j))}}. \tag{7}$$

(The transform $F_i(X_i)$ is called the *probability integral* transform of X_i.) The random variables $F_i(X_i)$ and $F_j(X_j)$ are bounded within [0, 1] and hence rank correlations always exist.

In an early work on rank correlations, Hotelling and Pabst (1936), citing previous work by Karl Pearson, show that for jointly normal random variables, the product-moment correlation ρ is related to the rank correlation r by

$$\rho=2\sin\left(\frac{\pi}{6}r\right). \tag{8}$$

The main advantage of rank correlations over product-moment correlations is that they are invariant under transformations of the input random variables. As an example, consider two log-normal random variables denoted by X_1 and X_2. The rank correlation between $\log(X_1)$ and $\log(X_2)$ is the same as the rank correlation between X_1 and X_2. Thus, the use of the rank correlation provides a natural way to separate the characterization of the individual marginal distributions from that of the dependence among the variables, simplifying the sample generation. A random vector $\mathbf{X}=(X_1,X_2,\ldots,X_d)'$ with marginals F_i, $i=1,2,\ldots,d$, and rank correlation matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$ can be sampled by first generating a random vector $\mathbf{U}=(U_1,U_2,\ldots,U_d)'$ with uniform marginals and the same rank correlation matrix $\boldsymbol{\Sigma}_{\mathbf{X}}$, and then applying the inverse probability transform to each variate individually: $X_i=F_i^{-1}(U_i)$ for $i=1,2,\ldots,d$.

Nelsen (1999) describes other monotonic-transformation-invariant measures of dependence that have been used in various applications. For example, the *orthant probability* measures the probability that both variables are above or both below their medians. It is alternatively called the *median deviation concordance* probability. It is invariant to monotonic transformations of the variates. For the bivariate normal distribution with correlation ρ, this probability is $1/2+\sin^{-1}(\rho)/\pi$ (Stuart and Ord, 1987). Another example of a widely known monotonic-transformation-invariant measure is Kendall's τ and is typically defined using independent and identically distributed random vectors, say (X_1,Y_1) and (X_2,Y_2). It is defined as the probability of concordance minus the probability of discordance

$$\begin{aligned}\tau(X,Y)=\mathrm{P}\big[(X_1-X_2)(Y_1-Y_2)>0\big]\\ -\mathrm{P}\big[(X_1-X_2)(Y_1-Y_2)<0\big].\end{aligned}$$

Clearly, concordance between two random variables arises if large values of one variable occur with the large values of the other and small values occur with small values of the other. Both the orthant probability and Kendall's τ have been studied for the special case of uniform marginals and their extension to more general cases remains an open area of interest.

4.2 Multivariate time series

Statistical time series analysis literature focuses attention chiefly on studying models that specify marginal distributions for individual elements and model the dependence within the series in terms of the correlogram, which is a system of lag-indexed product-moment correlations of the sequence. Assuming the stochastic process is stationary, the correlogram can be defined to be the set of correlations of the form $\boldsymbol{\Sigma}_X(h) = \mathrm{Cor}(\mathbf{X}_t, \mathbf{X}_{t+h})$, $h \geqslant 1$. Here, $\boldsymbol{\Sigma}_X(h)$ might correspond to a scalar or a matrix depending on whether the $\{\mathbf{X}_t; t \geqslant 1\}$ is a univariate or a multivariate time-series process.

The classical model in the multivariate time-series literature is the $d \times 1$ vector linear process given by

$$\mathbf{X}_t = \sum_{i=0}^{\infty} \boldsymbol{\psi}_i \mathbf{Y}_{t-i}, \quad t \geqslant 1. \tag{9}$$

The series $\{\boldsymbol{\psi}_i; i \geqslant 1\}$ is a sequence of $d \times d$ matrices such that $\sum_{i=0}^{\infty} \| \boldsymbol{\psi}_i \| < \infty$, where $\| \cdot \|$ denotes the usual eigenvalue norm, and $\{\mathbf{Y}_t; t = 0, \pm 1, \pm 2, \ldots\}$ is a sequence of independent and identically distributed d-dimensional white noise vectors with mean zero and covariance matrix $\boldsymbol{\Sigma}_\mathbf{Y}$ such that

$$\mathrm{E}[\mathbf{Y}_t] = \mathbf{0}_{(d\times 1)} \quad \text{and} \quad \mathrm{E}\bigl[\mathbf{Y}_t \mathbf{Y}'_{t+h}\bigr] = \begin{cases} \boldsymbol{\Sigma}_\mathbf{Y} & \text{if } h = 0, \\ \mathbf{0}_{(d\times d)} & \text{otherwise.} \end{cases}$$

Hannan (1970) shows that if the $\mathbf{X}_t$'s are Gaussian with mean zero and have an absolutely continuous spectrum, then there exists a sequence of independent and identically distributed normal mean-zero random vectors $\mathbf{Y}_t$, $t = 0, \pm 1, \pm 2, \ldots$, and a sequence of matrices $\{\boldsymbol{\psi}_i; i \geqslant 0\}$, such that the processes $\mathbf{X}_t$ and $\sum_{i=0}^{\infty} \boldsymbol{\psi}_i \mathbf{Y}_{t-i}$ are stochastically equal. Furthermore, if the $\mathbf{Y}_t$'s are Gaussian, then so are the $\mathbf{X}_t$'s in (9). If we choose a $d \times d$ matrix $\boldsymbol{\psi}$ for which $\sum_{i=0}^{\infty} \| \boldsymbol{\psi}^i \| < \infty$ holds and set $\boldsymbol{\psi}_i = \boldsymbol{\psi}^i$ for $i \geqslant 0$, then $\mathbf{X}_t$ simplifies to a stationary first-order vector autoregressive process that is written as $\mathbf{X}_t = \boldsymbol{\psi}\mathbf{X}_{t-1} + \mathbf{Y}_t$. This process is very well suited to modeling multivariate time series with normal marginal distributions. We refer the reader to Lütkepohl (1993) for a comprehensive review of the distributional properties of the multivariate autoregressive time-series processes.

However, there are many physical situations in which the marginals of the time-series input processes are nonnormal. If the input process is non-Gaussian, then the decomposition in (9) may not exist and the statistical inference procedures developed for processes satisfying (9) do not apply. Over the past decade, a considerable amount of research effort has been devoted to develop models of time series with exponential, gamma, geometric, or general discrete marginal distributions. We refer the reader to Lewis et al. (1989) and Block et al. (1990) for example work on univariate and bivariate autoregressive moving-average models with gamma marginals and exponential and geometric marginals, respectively. The primary shortcoming of the approach taken to construct these input processes is that a different model is required for each marginal distribution of interest and the sample paths of these processes, while adhering to the desired marginal distribution and autocorrelation structure, sometimes have unexpected features. In the succeeding subsection, we discuss a more general approach for modeling nonnormal time series.

4.3 Transformation-based methods

The methods covered in this section are in essence an extension of the transformation-based univariate generation procedure of Chapter 4, where any arbitrary distribution F is sampled using the inverse transformation of the uniform random variable $U(0, 1]$, $F^{-1}(U)$. This procedure works because the random variable $F^{-1}(U)$ has the cumulative distribution function F, and is extensively used since univariate uniform variables are very easily generated. The key idea remains the same in the multivariate setting, where we apply component-wise inverse marginal transformations to a base multivariate process with $U(0, 1]$ marginal distributions. Consider the generic case of modeling a d-variate stationary time series $\{\mathbf{X}_t; t \geqslant 1\}$ whose component time series $\{X_{i,t}; i = 1, 2, \ldots, d, t \geqslant 1\}$ has a marginal distribution denoted by F_i. For $i = 1, 2, \ldots, d$, we let $\{U_{i,t}; t \geqslant 1\}$ be the ith component of the d-variate time-series process $\{\mathbf{U}_t; t \geqslant 1\}$ with uniform marginals. We obtain the ith time series via the transformation

$$X_{i,t} = F_i^{-1}(U_{i,t}). \tag{10}$$

In the remainder of the section, we shall refer to the time series $\{\mathbf{U}_t; t \geqslant 1\}$ as the base process.

Two important issues need to be addressed: the construction of the base process $\mathbf{U}_t$ has to enable fast and efficient sampling and the dependence structure of the base process has to achieve the desired dependence structure of the input process. Both of these issues are tackled in different ways by the methods we shall discuss below. In Section 4.3.1, we present methods that construct the uniform base process by using transformations of a multivariate process with normal marginals. They can be used in the context of random vectors as well as times series, and can match both product-moment and rank correlation specifications. Section 4.3.2 presents a method for modeling uniform random vectors using chessboard copulas, while the method in Section 4.3.3 uses vine copulas. These copula construction procedures are typically used to match rank correlation specifications. Finally, Section 4.3.4 presents transform-expand-sample univariate time series processes whose base process is a sequence of autocorrelated uniforms generated in an autoregressive manner.

4.3.1 ARTA, NORTA and VARTA processes

The Autoregressive-To-Anything (ARTA) process, developed by Cario and Nelson (1996), defines a univariate time series with uniform marginals on (0, 1] via the transformation $U_t = \Phi(Z_t)$, where the base process $\{Z_t; t \geqslant 1\}$ is a stationary, standard, Gaussian autoregressive process of order p with the representation

$$Z_t = \sum_{h=1}^{p} \alpha_h Z_{t-h} + Y_t, \quad t = p+1, p+2, \ldots.$$

The α_h, $h = 1, \ldots, p$, are fixed autoregressive coefficients that uniquely determine the autocorrelation structure of the base process, $\rho_Z(h)$, $h = 1, \ldots, p$, and Y_t, $t = p+1, p+2, \ldots,$ are independent and identically distributed Gaussian random variables with mean zero and variance σ_Y^2. The univariate time series $\{X_t; t \geqslant 1\}$ is obtained via the transformation

$$X_t = F^{-1}(U_t) = F^{-1}\bigl[\Phi(Z_t)\bigr] \tag{11}$$

which ensures that X_t has distribution F. Therefore, the central problem is to select the autocorrelation structure, $\rho_Z(h)$, $h = 1, \ldots, p$, for the base process Z_t that gives the desired autocorrelation structure, $\rho_X(h)$, $h = 1, \ldots, p$, for the input process X_t. It is easily shown that the value of the lag-h base correlation $\rho_Z(h)$ depends only on the corresponding input correlation $\rho_X(h)$. The correlation structure determination step is thus equivalent to solving p independent correlation-matching problems.

A general, and related, method for obtaining random vectors with arbitrary marginal distributions and correlation matrix is described by Cario and Nelson (1997). It can be considered as broadening the ARTA process beyond a common marginal distribution. The central idea is to transform a standard multivariate normal vector into the desired random vector, which is referred as having a Normal-To-Anything (NORTA) distribution. Specifically, we let

$$\mathbf{X} = \bigl[F_1^{-1}\bigl(\Phi(Z_1)\bigr), F_2^{-1}\bigl(\Phi(Z_2)\bigr), \ldots, F_d^{-1}\bigl(\Phi(Z_d)\bigr)\bigr]', \tag{12}$$

where the base vector $\mathbf{Z} = (Z_1, \ldots, Z_d)'$ is a standard multivariate normal vector with correlation matrix $\mathbf{\Sigma_Z}$ and $F_1, F_2, \ldots, F_d$ are the desired marginal distributions. The base process setup problem then boils down to finding the base correlation matrix $\mathbf{\Sigma_Z}$ for $\mathbf{Z}$ that transforms to the desired correlation matrix $\mathbf{\Sigma_X}$ for $\mathbf{X}$. As in the earlier case, each $\mathbf{\Sigma_Z}(i, j)$ depends only on the corresponding component $\mathbf{\Sigma_X}(i, j)$, and hence a total of $d(d-1)/2$ independent correlation-matching problems have to be solved.

Recently, Biller and Nelson (2003) have pulled together the theory behind ARTA and NORTA processes and extended it to multivariate time series. They construct the Vector-Autoregressive-To-Anything (VARTA) process, which is a stationary, d-variate, vector time series $\{\mathbf{X}_t; t \geqslant 1\}$, by taking the inverse transformation of a standardized Gaussian vector autoregressive process of order p

$$\mathbf{Z}_t = \sum_{h=1}^{p} \boldsymbol{\alpha}_h \mathbf{Z}_{t-h} + \mathbf{Y}_t, \quad t = p+1, p+2, \ldots,$$

where the $\boldsymbol{\alpha}_i$, $i = 1, 2, \ldots, p$, are fixed $d \times d$ autoregressive coefficient matrices and $\{\mathbf{Y}_t; t \geqslant p+1\}$ is a sequence of independent and identically distributed d-variate multivariate normal random vectors with mean zero and a $d \times d$ covariance matrix $\boldsymbol{\Sigma}_Y$ chosen as $\boldsymbol{\Sigma}_Z(0) - \sum_{h=1}^{p} \boldsymbol{\alpha}_h \boldsymbol{\Sigma}'_Z(h)$. This selection of $\mathbf{Y}_t$ ensures that each $Z_{i,t}$ is marginally standard normal. For a fully specified VARTA process, there are $pd^2 + d(d-1)/2$ individual correlation matching problems to solve. Note that if $d = 1$, then the VARTA process reduces to an ARTA process; and if $d > 1$, but $p = 0$, the VARTA process corresponds to a NORTA vector.

The setup problem central to the ARTA, NORTA and VARTA processes is to solve correlation-matching problems of the form $c(\rho_Z) = \rho_X$ for ρ_Z, where c is a complex nonlinear function that requires the evaluation of double integrals. For the VARTA process, we can express input product-moment correlation $\rho_{\mathbf{X}}(i, j, h) = \mathrm{Cor}[X_{i,t}, X_{j,t+h}]$ as

$$\begin{aligned}
&\mathrm{Cor}\big[F_i^{-1}[\Phi(Z_{i,t})], F_j^{-1}[\Phi(Z_{j,t+h})]\big] \\
&\quad = \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_i^{-1}[\Phi(z_{i,t})]F_j^{-1}[\Phi(z_{j,t+h})] \right. \\
&\qquad\qquad \left. \times\, \vartheta_{\rho_{\mathbf{Z}}(i,j,h)}\big(z_{i,t}, z_{j,t+h}\big)\, \mathrm{d}z_{i,t}\, \mathrm{d}z_{j,t+h} - \mu_i\mu_j\right)(\sigma_i\sigma_j)^{-1} \\
&\quad = c_{ijh}\big[\rho_{\mathbf{Z}}(i, j, h)\big], \qquad (13)
\end{aligned}$$

where μ_i and σ_i^2 are the mean and the variance of the ith component of the input process and $\vartheta_{\rho_{\mathbf{Z}}(i,j,h)}$ is the standard bivariate normal probability density function with correlation $\rho_{\mathbf{Z}}(i, j, h)$. Clearly, $\rho_{\mathbf{X}}(i, j, h)$ depends only on $\rho_{\mathbf{Z}}(i, j, h)$. The function $c_{ijh}(\rho)$ is nondecreasing, lies on the origin for $\rho = 0$ and satisfies $|c_{ijh}(\rho)| \leqslant |\rho|$ for any $\rho \in [-1, 1]$ (Lancaster, 1957). Moreover, under mild conditions on the marginal distributions F_i and F_j, the function c_{ijh} is continuous (see Theorems 3.4 and 3.5 in Cario and Nelson, 1996; Biller and Nelson, 2003). Similar relations hold between the rank correlation $r_{\mathbf{X}}$ of the input process and the product-moment correlation $\rho_{\mathbf{Z}}$ of the base normal process.

The correlation-matching problems (13) can be solved analytically, numerically or using a simulation-based approach. It is possible to find the base dependence $\rho_{\mathbf{Z}}(i, j, h)$ values analytically only in special cases, e.g., if the marginal distributions F_i are uniform on $(0, 1]$, then the product-moment (and rank) correlation matching problem (13) reduces to $\rho_{\mathbf{X}}(i, j, h) = (6/\pi)\sin^{-1}[\rho_{\mathbf{Z}}(i, j, h)/2]$ (Kruskal, 1958). The astute reader will notice that this is the inverse of the relation in (8).

The idea behind the numerical approach is to take some initial base correlations, transform them into the implied correlations for the specified pair of marginals (using a numerical integration technique), and then employ a numerical root-finding method until we find a base correlation that approximates the desired correlation within a pre-specified level of accuracy. The properties of the function c_{ijh} ensure that a numerical search finds a value for the base correlation $\rho_{\mathbf{Z}}(i, j, h)$ efficiently. Both Cario and Nelson (1998) and Biller and Nelson (2003) provide such a numerical search method. The efficiency of the numerical search methods varies with the marginals and the $\rho_{\mathbf{X}}$ values: when the integrands are smooth, the numerical approach is very efficient, and may be inaccurate otherwise. Specialized numerical integration tricks might be required to overcome problematic situations.

The simulation-based approach can be used in cases where the form of the marginal distributions makes numerical evaluations of (13) prohibitively expensive. Equation (13) can be evaluated using sampling techniques as follows: A sample of pairs of bivariate standard normal random variates with correlation $\varphi_{\mathbf{Z}}(i, j, h)$ are generated. Each pair is then transformed via (11) into samples with the desired marginals, and the sample correlation estimate $\varphi_{\mathbf{X}}(i, j, h)$ is obtained from these transformed pairs. The key idea is that if $\varphi_{\mathbf{X}}(i, j, h)$ is sufficiently close to the desired input correlation $\rho_{\mathbf{X}}(i, j, h)$, then the base correlation $\varphi_{\mathbf{Z}}(i, j, h)$ is a good approximation to the base correlation $\rho_{\mathbf{Z}}(i, j, h)$ that solves the correlation-matching problem $c_{ijh}(\rho_{\mathbf{Z}}(i, j, h)) = \rho_{\mathbf{X}}(i, j, h)$. After a number of such sampling evaluations of (13), stochastic root finding algorithms are applied to search for the correlation of interest within a predetermined precision. This approach is very general and is a sensible alternative when the dimension of the problem is moderate and a diverse collection of marginal distributions have to be considered.

Throughout the previous discussion, we have assumed that a multivariate process exists with marginal distributions F_i, for $i = 1, 2, \ldots, d$, and an autocorrelation structure characterized by $\mathbf{\Sigma}_{\mathbf{X}}(h)$, $h = 0, 1, \ldots, p$. We need to exercise some care here because not all combinations of F_i, $i = 1, 2, \ldots, d$, and $\mathbf{\Sigma}_{\mathbf{X}}(h)$, $h = 0, 1, \ldots, p$, are feasible, as discussed in Section 4.1. Any multivariate process constructed by the transformation (11) or (12) from a base normal process with an appropriately chosen positive semidefinite base autocorrelation structure $\mathbf{\Sigma}_{\mathbf{Z}}(h)$ is automatically assured to be feasible. Unfortunately, the converse does not necessarily hold; that is, there exist sets of marginals with a feasible autocorrelation structure that cannot be represented by the VARTA transformation. This problem occurs because $\mathbf{\Sigma}_{\mathbf{Z}}(h)$ corresponding to a given feasible $\mathbf{\Sigma}_{\mathbf{X}}(h)$ structure is not positive semidefinite. This problem was postulated for the random vector case in Li and Hammond (1975) and was proved to exist in Ghosh and Henderson (2002a). Ghosh and Henderson (2002b) report that, in the case of random vectors, the failure of the transformation-based methods is relatively rare in moderate dimensions and that the method fails when the correlations lie either on the boundary or in close proximity to the set of correlations achievable for the specified marginals of the process. They note, however, that the problem does become increasingly evident as the dimension increases.

If, after solving the correlation-matching problems, the estimate for the base correlations $\mathbf{\Sigma}_Z$ is not positive definite, then the procedures proposed by Lurie and Goldberg (1998) and by Ghosh and Henderson (2002a) can be applied to find a symmetric, positive definite approximation of the base correlation matrix. The hope is that a multivariate process constructed in this manner will have a correlation structure close to the desired one, since the function c_{ijh} defined by (13) is known to be continuous under fairly general conditions.

4.3.2 Chessboard distributions

Ghosh and Henderson (2002a) study a class of copulas that they call chessboard distributions to model a random vector with uniform marginals given the correlation matrix. These distributions are closest in nature to the piecewise-uniform copulas developed by Mackenzie (1994). The construction of a chessboard distribution starts with the division of the d-dimensional unit hypercube into a grid of equally sized cells with sides of length $1/n$, where parameter n is said to index the level of discretization. The joint density of this copula is defined to be constant over each cell. Thus, the density over cell $C(j_1, \ldots, j_d)$, $j_i = 1, 2, \ldots, n$ for each $i = 1, \ldots, d$, is defined as $f(x) = n^d q(j)$, where $j = (j_1, \ldots, j_d)$ and $q(j)$ is the probability that the random vector takes values in the cell $C(j)$. Ghosh and Henderson (2002a) find the values of the $q(j)$s that match a given correlation matrix by solving a linear program. They additionally impose conditions on the values of the $q(j)$s to ensure that f is a d-variate density function.

After appropriate $q(j)$ values are determined, the data-generation procedure works as follows: an index vector $j = (j_1, \ldots, j_d)$ is generated from the discrete distribution represented by the $q(j)$s, and then the random vector sample is chosen uniformly from the chosen cell $C(j)$. Efficient algorithms are available to sample from the discrete distribution q, e.g., the alias method that is developed by Walker (1977) and discussed in detail in Law and Kelton (2000).

This correlation-matching procedure was first introduced in Ghosh and Henderson (2002a) to study feasibility of correlation matrices for any arbitrary set of marginals. They show that this procedure matches almost any (in a precise sense) feasible correlation matrix. The framework is advantageous because one can impose additional constraints on the joint distribution (Ghosh and Henderson, 2001). Thus, it gives the user a degree of control over the joint distribution.

Chessboard copulas can be used to sample random vectors with specified arbitrary continuous marginals and rank correlation matrices via the transformation (10). This approach can be extended to match arbitrary marginals with a specified product-moment correlation matrix (Ghosh, 2004).

However, this modeling technique suffers significant limitations. The chessboard distributions tend to be sparse in nature as the fraction of cells with positive valued $q(j)$s gets smaller with increasing dimensions. Moreover, the size of the linear programs (indexed by the discretization parameter n) that are solved to match a given correlation matrix can be very large. Ghosh (2004) develops bounds on the required discretization level n. Nevertheless, the chessboard distributions can be of significant help in modeling random vectors of low to moderate dimensions, and are especially attractive since the constructed joint distributions can be controlled.

4.3.3 Vine copula method

An alternative copula-based input model has recently been suggested by Cooke and his co-authors (Kurowicka and Cooke, 2002; Bedford and Cooke, 2002). This method is founded on Bayesian statistics literature and utilizes graph theoretic concepts. It generates copula samples quickly and efficiently, and can be used as a base for a fast sampling procedure that matches a set of arbitrary marginals and a rank correlation matrix.

To define a d-variate copula, the first step is to construct a graph structure called a *vine* on the d components. A vine is a nested set of spanning trees where the edges of tree j are the nodes of tree $j + 1$, starting with a tree on a graph with the d components as nodes. The number of nested trees is then $d - 2$. A *regular* vine on the d components is a vine in which two edges in tree j are joined by an edge in $j + 1$ only if they share a common node in the jth tree for $j = 1, 2, \dots, d - 2$. There are thus a total of $d(d - 1)/2$ edges in a regular vine.

Each edge is associated with a *conditional rank correlation* that takes values from the $[-1, 1]$ interval. A conditional rank correlation $r[X_i, X_j|S]$ is defined to be the rank correlation between the conditioned random variables $X_i|S$ and $X_j|S$, where S is a collection of other components. The set S of conditioning components is determined in a systematic manner (Cooke, 1997) and grows as we go deeper into the vine, with the edge of the $(d-2)$nd single-edge tree associated with a conditional rank correlation $r[X_u X_v|\{X_i, i \neq u \text{ or } v\}]$, for some u and v.

Cooke (1997) provides a method to sample from the vine. A bivariate copula is associated with each edge of the vine, and has a correlation equal to the conditional correlation value associated with the edge. Sampling starts with d independent uniform samples and proceeds by traversing the vine in a specific order and applying successive inversions of the conditional distributions derived from the bivariate copulas of each edge. Cooke (1997) associates minimum information copulas (see Meeuwissen and Bedford (1997) for a definition) with the vine edges and shows that the joint distribution of the random vector generated by this procedure is unique and possesses certain favorable properties.

Bedford and Cooke (2002) associate another class of bivariate copulas, the elliptical copulas, with the edges of the vine and their conditional correlations. This case has the advantage that the conditional correlations of the edges can now be related by a set of recursive equations to the set of correlations induced between the components of the random vector generated by the vine-traversing procedure. This implies that a set of conditional rank correlations of a regular vine can be derived for any desired correlation matrix, and thus the method can be used to match a given $d \times d$ rank correlation matrix to uniform marginals. Additionally, conditional distributions of elliptical copulas can be derived in explicit form and are very easy to invert. Thus vine-based sampling, which involves applying $d(d-1)/2$ of these inverses to d independent uniform variates, is fast and efficient. Kurowicka and Cooke (2002) show that this procedure can match any feasible correlation matrix with uniform marginals in the trivariate case. The method can come very close to achieving any feasible correlation matrix in higher dimensions (Cooke, 2003). However, it is not known whether they can achieve all feasible correlation matrices in dimensions greater than three.

4.3.4 TES processes

The Transform-Expand-Sample (TES) process, developed by Melamed (1991), generates a univariate time series with general marginals using as base process a sequence of autocorrelated uniforms generated in an autoregressive manner. Arbitrary marginals are obtained via a transformation similar to (10). TES processes play a significant role in univariate time series modeling and approach the generality of the ARTA process in generating stationary univariate time-series processes. Unfortunately, extensions to random vector and multivariate time-series processes are not available.

The TES process U_t can attain the full range of feasible lag-one autocorrelations for a given marginal distribution, and is defined via processes U_t^+ and U_t^- that are defined as follows:

$$U_t^+ = \begin{cases} U_0^+, & t = 0, \\ \langle U_{t-1}^+ + V_t \rangle, & t = 1, 2, \ldots, \end{cases}$$

$$U_t^- = \begin{cases} U_t^+, & t \text{ is even}, \\ 1 - U_t^+, & t \text{ is odd}, \end{cases}$$

where the random variables U_0^+ and V_t, $t \geqslant 1$, have $(0, 1]$-uniform distributions, V_t is independent of $\{U_0^+, V_1, \ldots, V_{t-1}\}$, and the notation $\langle x \rangle$ denotes modulo-1 arithmetic. The process U_t^+ covers the positive lag-one range, while the process U_t^- covers the negative lag-one range and is defined in terms of the U_t^+ process. The autocorrelation structure depends only on the distribution of V_t and can be manipulated by modifying the distribution of V_t without changing the marginal distribution of U_t. However, altering the distribution of V_t typically changes the autocorrelations of the process at all lags, but the user has no control over this behavior.

A disadvantage of the TES process is that extreme jumps may appear in the sample path due to the modulo-1 arithmetic. For example, U_{t-1} can be very close to 1 when U_t is very close to 0. A stitching transformation, $S_\xi(U_t)$ parameterized by $\xi \in [0, 1]$, mitigates this effect. The process $\{S_\xi(U_t); t \geqslant 1\}$ still has (0, 1]-uniform marginals, but no longer has extreme jumps. Unfortunately, stitching does not preserve the autocorrelation structure and the change cannot be expressed as a simple function of ξ. The distribution of V_t has to be modified until the autocorrelations of the constructed process match the pre-specified values. This forms one of the main challenges of using the TES process, and the ARTA process is perhaps a better option when the objective is to construct a time-series input process with pre-specified marginal distribution and autocorrelations through lag p. An ARTA process can be fitted to the desired set of $p \geqslant 1$ autocorrelations without user intervention, for any marginal distribution, and has smoother sample paths.

4.4 *Mixture methods*

Mixture methods sample random vectors with arbitrary marginals and a desired correlation matrix by probabilistically mixing samples from joint distributions with pre-determined correlation matrices. While this approach has been utilized with various configurations of pre-determined matrices, Hill and Reilly (1994) present what is perhaps the most elegant version.

Let $\mathbf{\Sigma_X} = [\rho_{ij}]_{(d\times d)}$ be the correlation matrix desired for a d-variate random vector $\mathbf{X}$ with marginals F_i, $i = 1, 2, \ldots, d$. Let $\underline{\rho}_{ij}$ and $\bar{\rho}_{ij}$ be the minimum and maximum correlations attainable between X_i and X_j. Whitt (1976) shows that these values are obtained when the bivariate vector (X_i, X_j) is generated as $(F_i(U), F_j(1 - U))$ and $(F_i(U), F_j(U))$, respectively, where U is a uniform random variable between 0 and 1. Hill and Reilly (1994) define an *extremal distribution* as a joint distribution that has $\rho_{ij} = \bar{\rho}_{ij}$ or $\rho_{ij} = \underline{\rho}_{ij}$ for every $i < j \leqslant d$. Associate a $(d(d-1)/2)$-vector $\boldsymbol{\delta}$ with each extremal distribution, where δ_{ij} takes value 1 if $\rho_{ij} = \bar{\rho}_{ij}$ and 0 otherwise. Hill and Reilly (1994) note that knowledge about any $d - 1$ components of $\boldsymbol{\delta}$ determines the rest of the vector. This comes from the result quoted earlier from Whitt (1976). Hence, there are potentially 2^{d-1} extremal correlation matrices for a d-dimensional $\mathbf{X}$.

Define $\mathbf{\Sigma}_0$ to be the zero-correlation matrix that corresponds to a random vector with uncorrelated components, and let $\mathbf{\Sigma}_\ell$, $\ell = 1, 2, \ldots, 2^{d-1}$, be the extremal correlation matrices. Hill and Reilly's method tries to match a desired correlation matrix $\mathbf{\Sigma_X}$ to a point in the convex hull of the set $\{\mathbf{\Sigma}_\ell; \ell = 0, 1, \ldots, 2^{d-1}\}$. The problem, formally, is to determine a set of composition probabilities λ_ℓ, $\ell = 0, 1, \ldots, 2^{d-1}$, that satisfy

$$\sum_{\ell=0}^{2^{d-1}} \lambda_\ell \left[\delta_{ij}^{\ell}\bar{\rho}_{ij} + \left(1 - \delta_{ij}^{\ell}\right)\underline{\rho}_{ij}\right] = \rho_{ij}, \quad \forall i < j,\ i, j = 1, \ldots, d, \tag{14}$$

$$\sum_{\ell=0}^{2^{d-1}} \lambda_\ell = 1, \tag{15}$$

$$\lambda_\ell \geqslant 0, \quad \ell = 0, \ldots, 2^{d-1}, \tag{16}$$

where δ_{ij} is as defined earlier. Once the λ_ℓ values are determined, the constructed joint distribution (a probabilistic mixture of the extremal distributions and the independent distribution) can be sampled by first generating an index ℓ from the discrete distribution $\{\lambda_\ell; \ell = 0, \ldots, 2^{d-1}\}$ and then producing a sample from the distribution that corresponds to the chosen index.

In many cases, there are an infinite number of solutions to the Equations (14)–(16). Hill and Reilly (1994) suggest formulating and solving a linear program with an appropriate objective function with Equations (14)–(16) as constraints to select a composite probability density function with desirable characteristics. Additional constraints might also be included; for instance, the frequency of sampling independent random vectors can be minimized by setting $\lambda_0 = 0$.

This approach suffers some major drawbacks. Extremal distributions have a distinctive structure; for instance, in the d-uniform random vector case, each extremal distribution places the entire mass uniformly on a chosen diagonal of the $\Re^d$-unit hypercube. Thus, this mixing procedure constructs distributions that can be quite unnatural to the environment being modeled. The correlation-matching problem defined via (14)–(16) grows exponentially with dimension, which makes the linear programs harder to solve in higher dimensions. The composition finding problem might also have no solution for some feasible correlation matrices. This happens when the set of all feasible correlation matrices for a given set of marginals does not lie within the convex polytope defined by (14)–(16). Ghosh and Henderson (2002a) give an example where this occurs. In light of these difficulties, this method is better suited for low dimensional random vectors.

5 Conclusion

A message the reader would consistently get from this chapter is that multivariate input process modeling is far from being a complete discipline. It is, if anything, a frontier of research in input modeling methodology that awaits conquering. Interesting problems which promise both methodological content as well as practical impact abound.

In terms of developing methods that purport to handle the true joint distribution of random vector input processes or approximations to it, much of the focus has been on developing parametric approximation approaches or on methods that exploit specific properties of these distribution functions. However, nonparametric approximation methods can have a significant impact and need to be explored more. For instance, a broad and deeply researched literature on procedures like kernel density estimation await our attention and adaptation.

We see that much of the focus has been on constructing methods that match partially specified multivariate processes. The majority of these methods handle dependence in terms of linear correlations in conjunction with marginal distributions. This approach has several limitations. First, most of the available methods are not guaranteed to work for all feasible correlation and marginals specifications. This effect is muted in lower dimensions, but can pose significant challenges as the dimension of the input process grows. Thus, there is a need for methods that can demonstrably work for any given feasible combination. Second, since correlations are single-valued linear dependence measures, they can miss much of the more complex interactions between various components of a multivariate process. Therefore, there is a clear need for models that capture more dependence information than do linear correlations. The ability to control the distribution of the generated multivariate process, for instance in the form of simple constraints on the joint distribution function, is desirable. We believe this objective of capturing more complex interactions and interdependencies will get an increasing impetus once more guidance along these lines become available from practitioners.

The presentation in this chapter has mostly focused on the modeling and variate-generation aspect of the input model development for stochastic simulation. Another important aspect of the simulation input model development is fitting models to historical data. Although it is common practice to assume that the desired marginal distributions and dependence structure are given in the current input modeling research, it is critically important to develop methods for fitting input models when only raw data generated by an unknown process are available. Recently, Biller and Nelson (2005) propose an automatic and statistically valid algorithm to fit stochastic models to dependent univariate time series for simulation. Development of similar methods for multivariate time-series input processes promises to be a very interesting research area.

References

Arnold, B.C., Castillo, E., Sarabia, J.M. (2001). Conditionally specified distributions. *Statistical Science* 16, 249–274.

Bedford, T., Cooke, R.M. (2002). Vines: A new graphical model for dependent random variables. *The Annals of Statistics* 30 (4), 1031–1068.

Biller, B., Nelson, B.L. (2003). Modeling and generating multivariate time-series input processes using a vector autoregressive technique. *ACM Transactions on Modeling and Computer Simulation* 13 (3), 211–237.

Biller, B., Nelson, B.L. (2005). Fitting arta processes for simulation. *Operations Research* 53 (3), 549–559.

Block, H.W., Langberg, N.A., Stoffer, D.S. (1990). Time series models for non-Gaussian processes. In: Block, W.H., Sampson, A.R., Savits, T.H. (Eds.), *Topics in Statistical Dependence*. Institute of Mathematical Statistics, Hayward, CA, pp. 69–83.

Burman, D. (1981). Insensitivity in queueing systems. *Advances in Applied Probability* 13, 846–859.

Cario, M.C., Nelson, B.L. (1996). Autoregressive to anything: Time-series input processes for simulation. *Operations Research Letters* 19, 51–58.

Cario, M.C., Nelson, B.L. (1997). Modeling and generating random vectors with arbitrary marginal distributions and correlation matrix. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.

Cario, M.C., Nelson, B.L. (1998). Numerical methods for fitting and simulating autoregressive-to-anything processes. *INFORMS Journal on Computing* 10, 72–81.

Charlier, C.V.L. (1906). Uber die Darstellung willkurlicher Funktionen. *Arkiv for Matematik, Astronomi och Fysik* 2 (20), 8–20.

Clemen, R.T., Reilly, T. (1999). Correlations and copulas for decision and risk analysis. *Management Science* 45, 208–224.

Cooke, R.M. (1997). Markov and entropy properties of tree- and vine-dependent variables. In: *Proceedings of the ASA Section on Bayesian Statistical Science*. American Statistical Association, Alexandria, VA, pp. 166–175.

Cooke, R.M. (2003). The vine copula method in higher dimensions. Personal communication.

DeBrota, D.J., Roberts, S.D., Swain, J.J., Dittus, R.S., Wilson, J.R., Venkatraman, S. (1988). Input modeling with the Johnson system of distributions. In: Haigh, P.L., Comfort, J.C., Abrams, M.A. (Eds.), *Proceedings of the 1988 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 165–179.

Deheuvels, P. (1977). Estimation non parametrique de la densité par histogrammes generalisés (ii). *Publications de l'Institut de Statistique de l'Université de Paris* 22, 1–23.

Devroye, L., Györfi, L. (1985). *Nonparametric Density Estimation: The L_1 View*. Wiley, New York.

Edgeworth, F.Y. (1898). Miscellaneous applications of the calculus of probabilities, contd. *Journal of the Royal Statistical Society* 61, 119–131.

Fang, K.-T., Kotz, S., Ng, K.W. (1990). *Symmetric Multivariate and Related Distributions*. *Monographs on Statistics and Applied Probability*, vol. 36. Chapman and Hall, New York.

Farin, G. (1990). *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 2nd edition. Academic Press, New York.

Gelman, A., Speed, T.P. (1993). Characterizing a joint probability distribution by conditionals. *Journal of the Royal Statistical Society* 55 (1), 185–188.

Geman, S., Geman, D. (1984). Stochastic relaxation, Gibbs distributions and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 721–741.

Ghosh, S. (2004). Dependence in stochastic simulation models. PhD thesis, Department of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.

Ghosh, S., Henderson, S.G. (2001). Chessboard distributions. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 385–393.

Ghosh, S., Henderson, S.G. (2002a). Chessboard distributions and random vectors with specified marginals and covariance matrix. *Operations Research* 50 (5), 820–834.

Ghosh, S., Henderson, S.G. (2002b). Properties of the NORTA method in higher dimensions. In: Yücesan, E., Chen, C.H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 263–269.

Gilks, W.R., Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 41, 337–348.

Hannan, E.J. (1970). *Multiple Time Series*. Wiley, New York, NY.

Hill, R.R., Reilly, C.H. (1994). Composition for multivariate random variables. In: Sadowski, D.A., Seila, A.F., Tew, J.D., Manivannan, S. (Eds.), *Proceedings of the 1994 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 332–339.

Hörmann, W. (1995). A rejection technique for sampling from t-concave distributions. *ACM Transactions on Mathematical Software* 21, 182–193.

Hörmann, W., Leydold, J., Derflinger, G. (2001). Universal methods of non-uniform random variate generation. Technical report, Department for Applied Statistics and Data Processing, University of Economics and Business Administration, Vienna, Austria.

Hotelling, H., Pabst, M.R. (1936). Rank correlation and tests of significance involving no assumption of normality. *Annals of Mathematical Statistics* 7, 29–43.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman and Hall, London.

Johnson, M.E. (1987). *Multivariate Statistical Simulation*. Wiley, New York.

Johnson, N.L. (1949a). Systems of frequency curves generated by methods of translation. *Biometrika* 36, 149–176.

Johnson, N.L. (1949b). Bivariate distributions based on simple translation systems. *Biometrika* 36, 297–304.

Johnson, N.L., Kotz, S. (1972). *Distributions in Statistics: Continuous Distributions*. Wiley, New York.

Johnson, N.L., Kotz, S., Balakrishnan, N. (1997). *Discrete Multivariate Distributions*. Wiley, New York.

Jong-Dev, K. (1984). Measures of dependence. In: Krishnaiah, P.R., Sen, P.K. (Eds.), *Handbook of Statistics*, vol. 4. Elsevier (North-Holland), Amsterdam.

Kotz, S., Balakrishnan, N., Johnson, N.L. (2000). *Continuous Multivariate Distributions, volume 1: Models and Applications*. Wiley, New York.

Kruskal, W. (1958). Ordinal measures of association. *Journal of the American Statistical Association* 53, 814–861.

Kurowicka, D., Cooke, R.M. (2002). The vine copula method for representing high dimensional dependent distributions: Application to continuous belief nets. In: Yücesan, E., Chen, C.H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 270–278.

Lancaster, H.O. (1957). Some properties of the bivariate normal distribution considered in the form of a contingency table. *Biometrika* 44, 289–292.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Lewis, P.A.W., McKenzie, E., Hugus, D.K. (1989). Gamma processes. *Communications in Statistics – Stochastic Models* 5, 1–30.

Leydold, J. (1998). A rejection technique for sampling from log-concave multivariate distributions. *ACM Transactions on Modeling and Computer Simulation* 8, 254–280.

Li, S.T., Hammond, J.L. (1975). Generation of pseudorandom numbers with specified univariate distributions and correlation coefficients. *IEEE Transactions on Systems, Man, and Cybernetics* 5, 557–561.

Livny, M., Melamed, B., Tsiolis, A.K. (1993). The impact of autocorrelation on queueing systems. *Management Science* 39, 322–339.

Lurie, P.M., Goldberg, M.S. (1998). An approximate method for sampling correlated random variables from partially-specified distributions. *Management Science* 44, 203–218.

Lütkepohl, H.L. (1993). *Introduction to Multiple Time Series Analysis*. Springer-Verlag, New York.

Mackenzie, G.R. (1994). Approximately maximum-entropy multivariate distributions with specified marginals and pairwise correlations. PhD thesis, Department of Decision Sciences, University of Oregon, Eugene, OR.

Marsaglia, G., Bray, T.A. (1964). A convenient method for generating normal variates. *SIAM Review* 6 (3), 260–264.

Meeuwissen, A.M.H., Bedford, T.J. (1997). Minimal informative distributions with given rank correlation for use in uncertainty analysis. *Journal of Statistical Computation and Simulation* 57 (4), 143–175.

Melamed, B. (1991). TES: A class of methods for generating autocorrelated uniform variates. *ORSA Journal of Computing* 3, 317–329.

Melamed, B., Hill, J.R., Goldsman, D. (1992). The TES methodology: Modeling empirical stationary time series. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 135–144.

Mirka, G.A., Glasscock, N.F., Stanfield, P.M., Wilson, J.R. (2000). An empirical approach to characterizing trunk muscle coactivation using simulation input modeling techniques. *Journal of Biomechanics* 33, 1701–1704.

Nelsen, R.B. (1999). *An Introduction to Copulas*. *Lecture Notes in Statistics*, vol. 139. Springer-Verlag, New York.

Parrish, R.S. (1990). Generating random deviates from multivariate Pearson distributions. *Computational Statistics and Data Analysis* 9, 283–295.

Parzen, E. (1962). On the estimation of a probability density function and mode. *Annals of Mathematical Statistics* 33, 1065–1076. Available at *http://citeseer.ist.psu.edu/parzen62estimation.html*.

Pearson, K. (1895). Contributions to the mathematical theory of evolution II: Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society A* 186, 343–414.

Ramberg, J.S., Schmeiser, B.W. (1974). An approximate method for generating asymmetric random variables. *Communications of the ACM* 17, 78–82.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Annals of Mathematical Statistics* 23, 470–472.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics* 27, 832–837.

Schmeiser, B., Deutsch, S. (1977). A versatile fourparameter family of probability distributions, suitable for simulation. *AIIE Transactions* 9, 176–182.

Schmeiser, B., Lal, R. (1982). Bivariate gamma random vectors. *Operations Research* 30, 355–374.

Silverman, B. (1982). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London.

Sklar, A. (1959). Fonctions de répartition à *n* dimensions et leurs marges. *Publications de l'Institut Statistique de l'Université de Paris* 8, 229–231.

Stanfield, P.M., Wilson, J.R., Mirka, G.A., Glasscock, N.F., Psihogios, J.P., Davis, J.R. (1996). Multivariate input modeling with Johnson distributions. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1457–1464.

Stuart, A., Ord, J.K. (1987). *Kendall's Advanced Theory of Statistics*, 5th edition. Oxford University Press, New York, NY.

Tong, Y.L. (1990). *The Multivariate Normal Distribution*. Springer-Verlag, New York.

Wagner, M.A.F., Wilson, J.R. (1995). Graphical interactive simulation input modeling with bivariate Bézier distributions. *ACM Transactions on Modeling and Computer Simulation* 5, 163–189.

Wagner, M.A.F., Wilson, J.R. (1996). Recent developments in input modeling with Bézier distributions. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1448–1456.

Walker, A.J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software* 3, 253–256.

Wand, M.P., Jones, M.C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

Ware, P.P., Page, T.W., Nelson, B.L. (1998). Automatic modeling of file system workloads using two-level arrival processes. *ACM Transactions on Modeling and Computer Simulation* 8, 305–330.

Whitt, W. (1976). Bivariate distributions with given marginals. *The Annals of Statistics* 4, 1280–1289.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13006-6

Chapter 6

Arrival Processes, Random Lifetimes and Random Objects

Lawrence M. Leemis
Department of Mathematics, The College of William & Mary, USA

Abstract

This chapter considers (1) the modeling, estimation, and generation of arrival processes, (2) the generation of random lifetimes, and (3) the generation of random objects. Accurate modeling of arrival processes is critical for the development of valid discrete-event simulation models. Although the generation of random lifetimes is typically applied to reliability and survival analysis in a simulation setting, their use is widespread in other disciplines as well. The generation of "random objects" will be applied here to generating combinatorial objects (e.g., combinations and permutations), matrices, and polynomials. The wider literature on generating random objects is much more diverse, including generating random card shuffles, generating random colors, generating random geometric objects, and generating random spawning trees.

1 Arrival processes

Most discrete-event simulation models have stochastic elements that mimic the probabilistic nature of the system under consideration. The focus in this section is on the modeling and generation of arrival processes. Stochastic models for arrival processes could be used to model the arrival times of customers to a queue, the arrival times of jobs to a job shop, the arrival times of demands in an inventory system, the arrival times of bills to Congress, or the arrival times of failures to a repairable machine. These stochastic models generalize to model events that occur over time or space. A close match between the arrival process model and the true underlying probabilistic mechanism associated with arrivals to the system of interest is required for successful simulation input modeling. The general question considered here is how to model an arrival process in a discrete-event simulation. It is typically the case that a data set of arrival times has been collected on the system of interest. We begin by introducing probabilistic models for an arrival process, which are special cases of what are known as "point processes", where "events" occur at points in time.

A special case of a point process is a "counting process", where event occurrences increment a counter.

1.1 Counting processes

A continuous-time, discrete-state stochastic process is often characterized by the *counting function* $\{N(t), t \geqslant 0\}$ which represents the total number of "events" that occur by time t (Ross, 2003). In a simulation setting, these events may be arrivals to a store, the births of babies, or machine failures. A counting process satisfies the following properties:

(1) $N(t) \geqslant 0$,
(2) $N(t)$ is integer-valued,
(3) if $s < t$, then $N(s) \leqslant N(t)$, and
(4) for $s < t$, $N(t) - N(s)$ is the number of events in $(s, t]$.

Two important properties associated with some counting processes are *independent increments* and *stationarity*. A counting process has independent increments if the number of events that occur in mutually exclusive time intervals are independent. A counting process is stationary if the distribution of the number of events that occur in any time interval depends only on the length of the interval. Thus, the stationarity property should only apply to counting processes with a constant rate of occurrence of events.

Counting processes can be used in modeling events as diverse as earthquake occurrences (Schoenberg, 2003), customer arrival times to an electronics store (White, 1999), and failure times of a repairable system (Nelson, 2003).

We establish some additional notation at this point which will be used in some results and process generation algorithms that follow. Let $X_1, X_2, \ldots$ represent the times between events in a counting process. Let $T_n = X_1 + X_2 + \cdots + X_n$ be the time of the nth event. With these basic definitions in place, we now define the Poisson process, which is the most fundamental of the counting processes.

1.2 Poisson processes

A Poisson process is a special type of counting process that is a fundamental base case for defining many other types of counting processes.

Definition (Ross, 2003). The counting process $\{N(t), t \geqslant 0\}$ is said to be a *Poisson process* with rate λ, $\lambda > 0$, if

(1) $N(0) = 0$,
(2) the process has independent increments, and
(3) the number of events in any interval of length t is Poisson distributed with mean λt.

The single parameter λ controls the rate at which events occur over time. Since λ is a constant, a Poisson process is often referred to as a *homogeneous* Poisson process. The third condition is equivalent to

$$\mathrm{P}\bigl(N(t+s)-N(s)=n\bigr)=\frac{(\lambda t)^n\exp(-\lambda t)}{n!},\quad n=0,1,2,\ldots,$$

and the stationarity property follows from it.

Although there are many results that follow from the definition of a Poisson process, three are detailed in this paragraph that have applications in discrete-event simulation. Proofs are given in any introductory stochastic processes textbook. First, given that n events occur in a given time interval $(s,t]$, the event times have the same distribution as the order statistics associated with n independent observations drawn from a uniform distribution on $(s,t]$. Second, the times between events in a Poisson process are independent and identically distributed exponential random variables with probability density function $f(x)=\lambda\exp(-\lambda x)$, for $x>0$. Since the mode of the exponential distribution is 0, a realization of a Poisson process typically exhibits significant clustering of events. Since the sum of n independent and identically distributed exponential(λ) random variables is Erlang(λ, n), T_n has a cumulative distribution function that can be expressed as a summation

$$F_{T_n}(t)=1-\sum_{k=0}^{n-1}\frac{(\lambda t)^k}{k!}\mathrm{e}^{-\lambda t},\quad t>0.$$

Third, analogous to the central limit theorem, which shows that the sum of arbitrarily distributed random variables is asymptotically normal, the superposition of renewal processes converges asymptotically to a Poisson process (Whitt, 2002, p. 318).

The mathematical tractability associated with the Poisson process makes it a popular model. It is the base case for queueing theory (e.g., the $M/M/1$ queue as defined in Gross and Harris, 1985) and reliability theory (e.g., the models for repairable systems described in Meeker and Escobar, 1998). Its rather restrictive assumptions, however, limit its applicability. For this reason, we consider variants of the Poisson process which can be useful for modeling more complex arrival processes.

1.3 Variants of Poisson processes

There are many variants of the Poisson process which would be appropriate for modeling arrivals in a discrete-event simulation model. We outline several of these models in this section. These variants are typically formulated by generalizing an assumption or a property of the Poisson process. Details can be found, for example, in Resnick (1992) or Nelson (2002).

Renewal processes

A renewal process is a generalization of a Poisson process. Recall that in a Poisson process, the interarrival times $X_1, X_2, \ldots,$ are i.i.d. exponential(λ) random variables. In a renewal process, the interarrival times are independent and identically distributed random variables from any distribution with positive support. One useful classification of renewal processes (Cox and Isham, 1980) concerns the coefficient of variation σ/μ of the distribution of the times between failures. This classification divides renewal processes into underdispersed and overdispersed processes. A renewal process is *underdispersed* (*overdispersed*) if the coefficient of variation of the distribution of the times between failures is less than (greater than) 1. An extreme case of an underdispersed process is when the coefficient of variation is 0 (i.e., deterministic interarrival times), which yields a deterministic renewal process. The underdispersed process is much more regular in its event times. In the case of a repairable system with underdispersed failure times, for example, it is easier to determine when it is appropriate to replace an item in order to avoid experiencing a failure. There is extreme clustering of events, on the other hand, in the case of an overdispersed renewal process, and replacement policies are less effective.

Alternating renewal processes

An alternating renewal process is a generalization of a renewal process that is often used to model the failure and repair times of a repairable item. Let $X_1, X_2, \ldots$ be i.i.d. random variables with positive support and cumulative distribution function $F_X(x)$ that represent the times to failure of a repairable item. Let $R_1, R_2, \ldots$ be i.i.d. random variables with positive support and cumulative distribution function $F_R(r)$ that represent the times to repair of a repairable item. Care must be taken to assure that $X_1, X_2, \ldots$ are indeed identically distributed, i.e., the item is neither improving nor deteriorating. Assuming that the alternating renewal process begins at time 0 with the item functioning, then

- X_1 is the time of the first failure,
- $X_1 + R_1$ is the time of the first repair,
- $X_1 + R_1 + X_2$ is the time of the second failure,
- $X_1 + R_1 + X_2 + R_2$ is the time of the second repair, etc.

Thus the times between events for an alternating renewal process alternate between two distributions, each with positive support.

Nonhomogeneous Poisson processes

A nonhomogeneous Poisson process is also a generalization of a Poisson process which allows for an arrival rate $\lambda(t)$ (known as the *intensity* function) that can vary with time.

Definition (Ross, 2003). The counting process $\{N(t), t \geqslant 0\}$ is said to be a *nonhomogeneous Poisson process* (NHPP) with intensity function $\lambda(t)$, $t \geqslant 0$, if

(1) $N(0) = 0$,
(2) the process has independent increments,
(3) $\mathrm{P}(N(t+h) - N(t) \geqslant 2) = \mathrm{o}(h)$, and
(4) $\mathrm{P}(N(t+h) - N(t) = 1) = \lambda(t)h + \mathrm{o}(h)$,

where a function $f(\cdot)$ is said to be $\mathrm{o}(h)$ if $\lim_{h\to 0} f(h)/h = 0$.

An NHPP is often appropriate for modeling a series of events that occur over time in a nonstationary fashion. Two common application areas are the modeling of arrivals to a waiting line (queueing theory) and the failure times of a repairable system (reliability theory) with negligible repair times. The cumulative intensity function

$$\Lambda(t) = \int_0^t \lambda(\tau)\,\mathrm{d}\tau, \quad t > 0,$$

gives the expected number of events by time t, i.e., $\Lambda(t) = \mathrm{E}[N(t)]$. The probability of exactly n events occurring in the interval $(a, b]$ is given by

$$\frac{[\int_a^b \lambda(t)\,\mathrm{d}t]^n \exp\{-\int_a^b \lambda(t)\,\mathrm{d}t\}}{n!}$$

for $n = 0, 1, \ldots$ (Çinlar, 1975).

Other variants

Other variants of a Poisson process have been proposed. For brevity, we outline three such variants. Details are given in Resnick (1992). *Mixed Poisson processes* can be formulated in terms of an NHPP with cumulative intensity function $\Lambda(t)$ and a random variable L with positive support. The associated counting process $N(L\Lambda(t))$ is a mixed Poisson process. Transforming the time scale with the random variable L results in a process that does not, in general, have independent increments. Ross (2003, p. 328) provides an illustration from the insurance industry where L models the claim rate (which varies from one policyholder to the next) and $\Lambda(t)$ is linear. *Doubly stochastic Poisson processes* generalize the notion of transforming the time scale by embedding a stochastic process within another stochastic process. The random variable L from a mixed Poisson process is replaced with a stochastic process with nondecreasing paths. Markov modulated Poisson processes are also a special case of doubly stochastic processes. *Compound Poisson processes* are formulated with a homogeneous or nonhomogeneous Poisson process and a sequence of i.i.d. random variables $D_1, D_2, \ldots$. The function

$$C(t) = \begin{cases} \sum_{i=1}^{N(t)} D_i, & \text{if } N(t) > 0, \\ 0, & \text{otherwise,} \end{cases}$$

defines a process that increases by $D_1, D_2, \ldots$ at each event time. This would be an appropriate model for an automobile insurance company whose claims occur according to a Poisson process with claim values $D_1, D_2, \ldots,$ and $C(t)$ models the total claim amounts that have occurred by time t. Similarly, if $D_1, D_2, \ldots$ are i.i.d. random variables with support on the nonnegative integers, then a compound Poisson process can be used to model batch arrivals.

1.4 Estimation

Poisson, renewal and alternating renewal processes

Poisson, renewal and alternating renewal processes all have independent interarrival times, which simplifies the estimation process. Let $x_1, x_2, \ldots, x_n$ denote independent observations collected on an interarrival time of interest from the "target" arrival process. The term *target* refers to the population arrival process that we want to estimate and simulate. These values have been assumed to be continuous in the models considered thus far. One simple trace-driven input model for variate generation would be to repeatedly select one of the data values with probability $1/n$, just as in the discrete case. This corresponds to an "empirical" c.d.f.

$$\widehat{F}(x) = \frac{M(x)}{n}, \quad x = x_i, i = 1, 2, \ldots, n,$$

where $M(x)$ is the number of data values that are less than or equal to x. One problem with this estimator is that only the data values will be generated in an associated variate generation algorithm. The interpolation problem can be overcome by filling the $n - 1$ "gaps" created by the n data values, with a piecewise-linear empirical c.d.f. These $n - 1$ gaps can each have a linear c.d.f. that rises $1/(n - 1)$ between each of the sorted data values $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$. The piecewise-linear empirical c.d.f. is

$$\widehat{F}(x) = \frac{i-1}{n-1} + \frac{x - x_{(i)}}{(n-1)(x_{(i+1)} - x_{(i)})}, \quad x_{(i)} \leqslant x < x_{(i+1)},$$

for $i = 1, 2, \ldots, n - 1$. Details concerning this model are given in Law and Kelton (2000, pp. 326 and 470) and a slight variant is presented in Banks et al. (2005, p. 281).

In the parametric case, the maximum likelihood estimation technique is generally used to estimate parameters due to its desirable statistical properties. If $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_q)'$ is a vector of unknown parameters associated with the interarrival density f, then the maximum likelihood estimator (MLE) maximizes the likelihood function

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i).$$

Statistical properties of MLEs are given in Casella and Berger (2002). Details concerning selecting the appropriate parametric distribution(s) (in the renewal and alternating renewal cases), estimating parameters, and assessing goodness-of-fit are given in Law and Kelton (2000).

Nonhomogeneous Poisson processes

This subsection describes two nonparametric (trace-driven) procedures and two parametric procedures for estimating the intensity function $\lambda(t)$ or, equivalently, the cumulative intensity function $\Lambda(t) = \int_0^t \lambda(\tau)\,\mathrm{d}\tau$ from k realizations sampled from a target NHPP. The first nonparametric procedure can be used on "count" or "interval" data. The second nonparametric procedure is appropriate for "raw" data. In each of the procedures, we find point estimates and interval estimates for the cumulative intensity function.

First consider finding an estimate for the intensity function when only count data are available. The intensity function $\lambda(t)$ or cumulative intensity function $\Lambda(t)$ is to be estimated on $(0, S]$, where S is a known constant. The interval $(0, S]$ may represent the time a system allows arrivals (e.g., 9:00 a.m. to 5:00 p.m. at a bank) or one period of a cycle (e.g., one day at an emergency room). There are k representative realizations collected on the target process on $(0, S]$.

For systems with high arrival rates (e.g., busy call centers or web sites), there is often so much data that *counts* of events that occur during time subintervals are available, rather than the raw event times. Although this is less preferable that having the raw data, it is still possible to construct an estimate of the intensity function and generate variates for a discrete-event simulation model. The time interval $(0, S]$ can be partitioned into m subintervals $(a_0, a_1], (a_1, a_2], \ldots, (a_{m-1}, a_m]$, where $a_0 = 0$ and $a_m = S$. The subintervals do not necessarily have equal widths. Let $n_1, n_2, \ldots, n_m$ be the total number of observed events in the subintervals over the k realizations.

To simplify the estimation process, assume that the target NHPP has an intensity function $\lambda(t)$ that is piecewise constant on each subinterval of the partition $(a_0, a_1], (a_1, a_2], \ldots, (a_{m-1}, a_m]$. Since the average intensity function on the interval $(a_{i-1}, a_i]$ is the rate per unit time of the events that occur on that interval, the maximum likelihood estimator is the average number of events that occurred on the interval, normalized for the length of the interval (Law and Kelton, 2000, pp. 390–393; Leemis, 2004)

$$\hat{\lambda}(t) = \frac{n_i}{k(a_i - a_{i-1})}, \qquad a_{i-1} < t \leqslant a_i,$$

for $i = 1, 2, \ldots, m$. Since the intensity estimator is piecewise constant, the associated cumulative intensity function estimator is a continuous, piecewise-linear function on $(0, S]$

$$\widehat{\Lambda}(t) = \int_0^t \hat{\lambda}(\tau)\,\mathrm{d}\tau = \left(\sum_{j=1}^{i-1} \frac{n_j}{k}\right) + \frac{n_i(t - a_{i-1})}{k(a_i - a_{i-1})}, \qquad a_{i-1} < t \leqslant a_i,$$

for $i = 1, 2, \ldots, m$. (If there are no events observed on interval i, i.e., $n_i = 0$, then the intensity function estimate is zero on interval i and the cumulative intensity function estimate is constant on interval i.) This estimator passes through the points $(a_i, \sum_{j=1}^{i} n_j/k)$ for $i = 1, 2, \ldots, m$. Asymptotic properties of this estimator in the case of equal-width subintervals are considered by Henderson (2003). Variate generation via inversion is straightforward.

It is important to assess the precision of the point estimator, which is developed thus far, which is typically done via confidence intervals. Based on the fact that the number of events by time t has a Poisson$(\Lambda(t))$ distribution, an approximate, two-sided $(1-\alpha)100\%$ confidence interval for $\Lambda(t)$ is

$$\widehat{\Lambda}(t) - z_{\alpha/2}\sqrt{\frac{\widehat{\Lambda}(t)}{k}} < \Lambda(t) < \widehat{\Lambda}(t) + z_{\alpha/2}\sqrt{\frac{\widehat{\Lambda}(t)}{k}}$$

for $0 < t \leqslant S$, where $z_{\alpha/2}$ is the $1 - \alpha/2$ fractile of the standard normal distribution. The interval is always asymptotically exact at the endpoints, but only asymptotically exact for all t in $(0, S]$ when the target intensity function $\lambda(t)$ is piecewise constant over each subinterval $(a_{i-1}, a_i]$ in the arbitrary partition of $(0, S]$. In most applications, this assumption is *not* satisfied.

Now consider the case where *raw event times* are available for the k realizations collected on $(0, S]$. The meaning of n_i now changes from the count data case. We now let n_i, $i = 1, 2, \ldots, k$, be the number of observations in the ith realization, $n = \sum_{i=1}^{k} n_i$, and let $t_{(1)}, t_{(2)}, \ldots, t_{(n)}$ be the order statistics of the superposition of the event times in the k realizations, $t_{(0)} = 0$ and $t_{(n+1)} = S$.

The standard step-function estimator for the cumulative intensity function $\Lambda(t)$ takes upward steps of height $1/k$ only at the event times in the superposition of the k realizations collected from the target process. This means that a variate generation algorithm will produce realizations that only contain event times that are equal to the ones that were collected in one of the k realizations. This is known as the "interpolation" problem, which can be overcome by using a piecewise-linear cumulative intensity function estimator.

There are $n + 1$ "gaps" on $(0, S]$ created by the superposition $t_{(1)}, t_{(2)}, \ldots, t_{(n)}$. Setting $\widehat{\Lambda}(S) = n/k$ yields a process where the expected number of events by time S is the average number of events in k realizations, since $\Lambda(S)$ is the expected number of events by time S. The piecewise-linear cumulative intensity function estimate rises by $n/[(n+1)k]$ at each event time in the superposition. Thus the piecewise-linear estimator of the cumulative intensity function between the time values in the superposition is

$$\widehat{\Lambda}(t) = \frac{in}{(n+1)k} + \left[\frac{n(t - t_{(i)})}{(n+1)k(t_{(i+1)} - t_{(i)})}\right], \quad t_{(i)} < t \leqslant t_{(i+1)},$$

for $i = 0, 1, 2, \ldots, n$. This estimator passes through the points $(t_{(i)}, in/(n+1)k)$, for $i = 1, 2, \ldots, n+1$. This estimator was developed in Leemis (1991) and extended to nonoverlapping intervals in Arkin and Leemis (2000).

A $(1-\alpha)100\%$ asymptotically exact (as $k \to \infty$) confidence interval for $\Lambda(t)$ is given by

$$\widehat{\Lambda}(t) - z_{\alpha/2}\sqrt{\frac{\widehat{\Lambda}(t)}{k}} < \Lambda(t) < \widehat{\Lambda}(t) + z_{\alpha/2}\sqrt{\frac{\widehat{\Lambda}(t)}{k}}$$

for $0 < t \leqslant S$, where $z_{\alpha/2}$ is the $1-\alpha/2$ fractile of the standard normal distribution.

The estimation procedure given here is nonparametric and does not require any arbitrary decisions (e.g., parameter values) from the modeler. The fact that each of the piecewise segments rises the same height can be exploited to create an efficient variate generation algorithm via inversion. As n increases, the amount of memory required increases, but the amount of execution time required to generate a realization depends only on the ratio n/k, the average number of events per realization. Thus, collecting more realizations (resulting in narrower confidence intervals) increases the amount of memory required, but does not impact the expected execution time for generating a realization.

To summarize, this piecewise-linear cumulative intensity function estimator with raw data is in some sense ideal in that

- it uses raw data which avoids the loss of accuracy imposed by breaking $(0, S]$ into arbitrary time subintervals,
- the point and interval estimates of $\Lambda(t)$ are closed form and easily computed,
- the point estimate of $\Lambda(t)$ is consistent, i.e., $\lim_{k\to\infty} \widehat{\Lambda}(t) = \Lambda(t)$ with probability 1, which is proved using the strong law of large numbers,
- the interval estimate of $\Lambda(t)$ is asymptotically exact as $k \to \infty$, which is proved using the central limit theorem and Slutsky's theorem, and
- the variate generation algorithm which can be used to simulate a realization from $\widehat{\Lambda}(t)$ is efficient, monotone, and synchronized (see Chapter 4 for definitions of these terms).

One downside over a parametric approach is that the variate generation algorithm is storage intensive, since all arrival times collected must be stored in memory.

We now shift to the parametric modeling of NHPPs. Maximum likelihood can again be used for estimating the parameters in a parametric NHPP model. The likelihood function for estimating the vector of unknown parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_q)'$ from a single realization of event times $t_1, t_2, \ldots, t_n$ drawn from an NHPP with intensity $\lambda(t)$ on $(0, S]$ is

$$L(\boldsymbol{\theta}) = \left[\prod_{i=1}^{n} \lambda(t_i)\right] \exp\left[-\int_0^S \lambda(t)\,\mathrm{d}t\right].$$

Maximum likelihood estimators can be determined by maximizing $L(\boldsymbol{\theta})$ or its logarithm with respect to the q unknown parameters. Asymptotically exact confidence regions for the unknown parameters can be found via the likelihood ratio statistic.

Because of the additive property of the intensity function for multiple realizations, the likelihood function for the case of k realizations is

$$L(\boldsymbol{\theta}) = \left[\prod_{i=1}^{n} k\lambda(t_i)\right] \exp\left[-\int_0^S k\lambda(t)\,\mathrm{d}t\right].$$

Example. There are many potential parametric models for nonhomogeneous Poisson processes. Consider a *power law process*, where the intensity function is

$$\lambda(t) = b^a a t^{a-1}, \quad t > 0,$$

for shape parameter $a > 0$ and scale parameter $b > 0$. The analysis for other intensity functions is similar. This intensity function can assume monotone increasing and monotone decreasing shapes. Thus, the likelihood function for k realizations is

$$L(b, a) = k^n b^{na} a^n \mathrm{e}^{-k(bS)^a} \prod_{i=1}^{n} t_i^{a-1}.$$

The log-likelihood function is

$$\log L(b, a) = n\log(ka) + na\log b - k(bS)^a + (a-1)\sum_{i=1}^{n}\log t_i.$$

Differentiating with respect to a and b and equating to zero yields

$$\frac{\partial \log L(b, a)}{\partial a} = n\log b + \frac{n}{a} + \sum_{i=1}^{n}\log t_i - k(bS)^a\log(bS) = 0$$

and

$$\frac{\partial \log L(b, a)}{\partial b} = \frac{an}{b} - kS^a a b^{a-1} = 0.$$

These equations can be solved in closed form. The analytic expressions for b and a are:

$$\hat{a} = \frac{n}{n\log S - \sum_{i=1}^{n}\log t_i},$$
$$\hat{b} = \frac{1}{S}\left(\frac{n}{k}\right)^{1/a}.$$

It is oftentimes the case that a standard parametric model such as a power law process is unable to adequately describe the intensity function. When this is the case, the EPTMP (exponential-polynomial-trigonometric function with multiple periodicities) model, originally given by Lee et al. (1991) and generalized by Kuhl et al. (1997) with intensity function

$$\lambda(t) = \exp\left[\sum_{i=0}^{m} \alpha_i t^i + \sum_{j=1}^{p} \gamma_j \sin(\omega_j t + \phi_j)\right], \quad t > 0,$$

can also model a nonmonotonic intensity function. The exp function forces the intensity function to be positive, the first summation models a polynomial trend and the second summation models sinusoidal periodicities in the intensity. The cyclic portion of the model has been used in discrete-event simulation applications to model storm times in the Arctic Sea (Lee et al., 1991) and arrivals of livers for transplantation by donors (Pritsker et al., 1995). Goodness-of-fit tests associated with the fitted models are given in Rigdon and Basu (2000).

1.5 Process generation

The algorithms presented in this section generate a sequence of event times (in our setting they are typically the arrival times in a discrete-event simulation model) on the time interval $(0, S]$, where S is a real, fixed constant. If the next-event approach is taken for placing events onto the calendar in a discrete-event simulation model, then these algorithms should be modified so that they take the current event time as an argument and return the next event time. All processes are assumed to begin at time 0. The event times that are generated by the counting process are denoted by $T_1, T_2, \ldots,$ and random numbers (i.e., $\mathrm{U}(0, 1)$ random variables) are denoted by U or $U_1, U_2, \ldots.$ If just $T_0 = 0$ is returned, then no arrivals were observed on $(0, S]$. Indentation is used in the algorithms to indicate nesting.

Poisson processes

Since the times between arrivals in a Poisson process are i.i.d. exponential(λ) random variables, the following algorithm generates the arrival times of a Poisson process on $(0, S]$.

```
$T_0 \leftarrow 0$
$i \leftarrow 0$
while $T_i \leqslant S$
  $i \leftarrow i + 1$
  generate $U_i \sim \mathrm{U}(0, 1)$
  $T_i \leftarrow T_{i-1} - \log(1 - U_i)/\lambda$
return $T_1, T_2, \ldots, T_{i-1}$
```

Renewal processes

Arrivals in a renewal process are generated in a similar fashion to a Poisson process. Let $F_X(x)$ denote the cumulative distribution function of the interarrival times $X_1, X_2, \ldots$ in a renewal process. The following algorithm generates the arrival times on $(0, S]$.

```
$T_0 \leftarrow 0$
$i \leftarrow 0$
while $T_i \leqslant S$
    $i \leftarrow i + 1$
    generate $U_i \sim \mathrm{U}(0, 1)$
    $T_i \leftarrow T_{i-1} + F_X^{-1}(U_i)$
return $T_1, T_2, \ldots, T_{i-1}$
```

Alternating renewal processes

Arrivals in an alternating renewal process are generated in a similar fashion to a renewal process, but the interarrival time must alternate between $F_X(x)$, the cumulative distribution function of the times to failure $X_1, X_2, \ldots$ and $F_R(r)$, the cumulative distribution function of the times to repair $R_1, R_2, \ldots$. The following algorithm generates the arrival times on $(0, S]$.

```
$T_0 \leftarrow 0$
$i \leftarrow 0$
$j \leftarrow 0$
while $T_i \leqslant S$
    $i \leftarrow i + 1$
    generate $U_i \sim \mathrm{U}(0, 1)$
    if $j = 0$
        $T_i \leftarrow T_{i-1} + F_X^{-1}(U_i)$
        $j \leftarrow j + 1$
    else
        $T_i \leftarrow T_{i-1} + F_R^{-1}(U_i)$
        $j \leftarrow j - 1$
return $T_1, T_2, \ldots, T_{i-1}$
```

Nonhomogeneous Poisson process

Arrivals can be generated for use in discrete-event simulation as $\Lambda^{-1}(E_1)$, $\Lambda^{-1}(E_2), \ldots$, where $E_1, E_2, \ldots$ are the event times in a unit Poisson process (Çinlar, 1975). This technique is often referred to as "inversion" and is implemented below.

```
$T_0 \leftarrow 0$
$E_0 \leftarrow 0$
$i \leftarrow 0$
while $T_i \leqslant S$
    $i \leftarrow i + 1$
    generate $U_i \sim \mathrm{U}(0, 1)$
    $E_i \leftarrow E_{i-1} - \log(1 - U_i)$
    $T_i \leftarrow \Lambda^{-1}(E_i)$
return $T_1, T_2, \ldots, T_{i-1}$
```

The inversion algorithm is ideal when $\Lambda(t)$ can be inverted analytically, although it also applies when $\Lambda(t)$ needs to be inverted numerically. There may be occasions when the numerical inversion of $\Lambda(t)$ is so onerous that the *thinning* algorithm devised by Lewis and Shedler (1979) and given below may be preferable. This algorithm assumes that the modeler has determined a *majorizing* value λ^* that satisfies $\lambda^* \geqslant \lambda(t)$ for all $t > 0$.

```
$T_0 \leftarrow 0$
$i \leftarrow 0$
**while** $T_i \leqslant S$
    $t \leftarrow T_i$
    **repeat**
        **generate** $U \sim \mathrm{U}(0, 1)$
        $t \leftarrow t - \log(1 - U)/\lambda^*$
        **generate** $U \sim \mathrm{U}(0, \lambda^*)$
    **until** $U \leqslant \lambda(t)$
    $i \leftarrow i + 1$
    $T_i \leftarrow t$
**return** $T_1, T_2, \ldots, T_{i-1}$
```

The majorizing value λ^* can be generalized to a majorizing function $\lambda^*(t)$ to decrease the CPU time by minimizing the probability of "rejection" in the **repeat–until** loop.

All of the arrival process models given in this section are elementary. More complex models are considered in Nelson et al. (1995).

2 Generating random lifetimes

This section concerns algorithms for generating continuous, positive random variables, referred to here as "lifetimes". Although the main two applications areas are reliability (e.g., a machine or product lifetime, see, for example, Meeker and Escobar, 1998) and survival analysis (e.g., patient survival time after an organ transplant, see, for example, Lawless, 2003), their use is widespread in other disciplines (e.g., sociological applications as in Allison, 1984). The discussion here is limited to generating *continuous*, as opposed to discrete or mixed distributions, due to their pervasiveness in the reliability and survival analysis literature.

2.1 Hazard-based methods

There are three hazard-based methods for generating lifetimes which parallel the associated density-based methods that were introduced in Chapter 4: (1) the inverse cumulative hazard function technique, an inversion technique that parallels the inverse-c.d.f. technique, (2) competing risks, a linear combination technique that parallels composition, and (3) thinning, a majorization technique that parallels acceptance/rejection.

Definitions

Consider a positive, continuous random variable T, referred to here as a “lifetime”. We briefly introduce four functions, each of which completely describes the distribution of T: the survivor function, the probability density function, the hazard function, and the cumulative hazard function.

The survivor function, also known as the reliability function and complementary c.d.f., is defined by

$$S(t) = \mathrm{P}(T \geqslant t), \quad t > 0,$$

a nonincreasing function of t satisfying $S(0) = 1$ and $\lim_{t\to\infty} S(t) = 0$. The survivor function is important in the study of systems of components since it is the appropriate argument in the structure function to determine system reliability (Meeker and Escobar, 1998). Notice that $S(t)$ is the fraction of the population that survives to time t, as well as the probability that a single item survives to time t. For continuous random variables, $S(t) = 1 - F(t)$, where $F(t) = \mathrm{P}(T \leqslant t)$ is the c.d.f.

When the survivor function is differentiable,

$$f(t) = -S'(t), \quad t \geqslant 0,$$

is the associated p.d.f. For any interval (a, b), where $a < b$,

$$\mathrm{P}(a \leqslant T \leqslant b) = \int_a^b f(t)\,\mathrm{d}t.$$

The hazard function, also known as the rate function, failure rate and force of mortality, can be defined by

$$h(t) = \frac{f(t)}{S(t)}, \quad t \geqslant 0.$$

The hazard function is popular in reliability because it has the intuitive interpretation as the amount of *risk* associated with an item that has survived to time t. The hazard function is mathematically equivalent to the intensity function for a nonhomogeneous Poisson process, and the failure time corresponds to the first event time in the process. Competing risks models are easily formulated in terms of $h(t)$, as shown subsequently.

The cumulative hazard function can be defined by

$$H(t) = \int_0^t h(\tau)\,\mathrm{d}\tau, \quad t \geqslant 0.$$

Any one of the four functions that describes the distribution of T can be used to determine the others, e.g., $H(t) = -\log S(t)$.

We now introduce the three hazard-based lifetime variate generation algorithms: the inverse cumulative hazard function technique, competing risks, and thinning. Random numbers are denoted by U and the associated random lifetimes are denoted by T.

Inverse cumulative hazard function technique

If T is a random lifetime with cumulative hazard function H, then $H(T)$ is an exponential random variable with a mean of one. This result, which is an extension of the probability integral transformation, is the basis for the inverse cumulative hazard function technique. Therefore,

```
**generate** $U \sim \mathrm{U}(0, 1)$
$T \leftarrow H^{-1}(-\log(1 - U))$
**return** $T$
```

generates a single random lifetime T. This algorithm is easiest to implement when H can be inverted in closed form. This algorithm is monotone and synchronized. Although the sense of the monotonicity is reversed, $1 - U$ can be replaced with U in order to save a subtraction.

Competing risks

Competing risks (David and Moeschberger, 1978) is a linear combination technique that is analogous to the density-based composition method. The composition method is viable when the p.d.f. can be written as the convex combination of k density functions

$$f(t) = \sum_{j=1}^{k} p_j f_j(t), \quad t \geqslant 0,$$

where $\sum_{j=1}^{k} p_j = 1$. The competing risks technique applies when the hazard function can be written as the sum of hazard functions, each corresponding to a "cause" of failure

$$h(t) = \sum_{j=1}^{k} h_j(t), \quad t \geqslant 0,$$

where $h_j(t)$ is the hazard function associated with cause j of failure acting in a population. The minimum of the lifetimes from each of these risks corresponds to the system lifetime. Competing risks is most commonly used to analyze a series system of k components, but it can also be used in actuarial applications with k causes of failure. The competing risks model is also used for modeling competing failure modes for components that have multiple failure modes. The algorithm to generate a lifetime T is

```
**for** $j$ **from** $1$ **to** $k$
    **generate** $T_j \sim h_j(t)$
$T \leftarrow \min\{T_1, T_2, \ldots, T_k\}$
**return** $T$
```

The $T_1, T_2, \ldots, T_k$ values can be generated by any of the standard random variate generation algorithms.

Thinning

The third class of techniques for generating random lifetimes is majorization techniques. The density-based acceptance/rejection technique uses a *majorizing function* $f^*(t)$ that satisfies $f^*(t) \geqslant f(t)$ for all $t \geqslant 0$. Likewise *thinning*, which was originally suggested by Lewis and Shedler (1979) for generating the event times in a nonhomogeneous Poisson process, can be adapted to produce a single lifetime by returning only the first event time generated. A majorizing hazard function $h^*(t)$ must be found that satisfies $h^*(t) \geqslant h(t)$ for all $t \geqslant 0$. The algorithm is

```
$T \leftarrow 0$
repeat
  generate $Y$ from $h^*(t)$ given $Y > T$
  $T \leftarrow T + Y$
  generate $S \sim \mathrm{U}(0, h^*(T))$
until $S \leqslant h(T)$
return $T$
```

Generating Y in the loop can be performed by inversion or any other method. The name *thinning* comes from the fact that T can make several steps, each of length Y, that are thinned out before the loop condition is satisfied.

2.2 Survival models involving covariates

The accelerated life and proportional hazards lifetime models can account for the effects of covariates on a random lifetime (Cox and Oakes, 1984). Variate generation for these models is a straightforward extension of the basic methods for generating random lifetimes. Variate generation algorithms for Monte Carlo simulation of nonhomogeneous Poisson processes are a simple extension of the inverse cumulative hazard function technique.

The effect of covariates (explanatory variables) on survival often complicates the analysis of a set of lifetime data. In a medical setting, these covariates are usually patient characteristics such as age, gender, or blood pressure. In reliability, these covariates are exogenous variables such as the turning speed of a machine tool or the stress applied to a component that affect the lifetime of an item. We use the generic term *item* here to represent a manufactured product or organism whose survival time is of interest. Two common models to incorporate the effect of the covariates on lifetimes are the *accelerated life* and *Cox proportional hazards* models.

The $q \times 1$ vector $\mathbf{z}$ contains covariates associated with a particular item. The covariates are linked to the lifetime by the function $\psi(\mathbf{z})$, which satisfies $\psi(\mathbf{0}) = 1$ and $\psi(\mathbf{z}) \geqslant 0$ for all $\mathbf{z}$. A popular choice is the log linear form $\psi(\mathbf{z}) = \exp(\boldsymbol{\beta}'\mathbf{z})$, where $\boldsymbol{\beta}$ is a $q \times 1$ vector of regression coefficients.

Accelerated life model

The cumulative hazard function for T in the accelerated life model is

$$H(t) = H_0\big(t\psi(\mathbf{z})\big),$$

where H_0 is a baseline cumulative hazard function. When $\mathbf{z} = 0, H(t) = H_0(t)$. In this model, the covariates accelerate ($\psi(\mathbf{z}) > 1$) or decelerate ($\psi(\mathbf{z}) < 1$) the rate that the item moves through time.

Proportional hazards model

The cumulative hazard function for T in the proportional hazards model is

$$H(t) = \psi(\mathbf{z})H_0(t).$$

In this model, the covariates increase ($\psi(\mathbf{z}) > 1$) or decrease ($\psi(\mathbf{z}) < 1$) the hazard function associated with the lifetime of the item by the factor $\psi(\mathbf{z})$ for all values of t. This model is known in medicine as the "Cox model" and is a standard model for evaluating the effect of covariates on survival. We do not explicitly consider the estimation of the regression coefficients $\boldsymbol{\beta}$ here since the focus is on random lifetime generation. Cox and Oakes (1984) and others give the details associated with estimation of $\boldsymbol{\beta}$ and most modern statistical packages estimate these coefficients using built-in numerical methods.

Lifetime generation

All of the algorithms for variate generation for these models are based on the fact that $H(T)$ is exponentially distributed with a mean of one. Therefore, equating the cumulative hazard function to $-\log(1 - U)$, where $U \sim \mathrm{U}(0, 1)$, and solving for t yields the appropriate generation technique (Leemis, 1987).

In the accelerated life model, since time is being expanded or contracted by a factor $\psi(\mathbf{z})$, variates are generated by

$$T \leftarrow \frac{H_0^{-1}(-\log(1 - U))}{\psi(\mathbf{z})}.$$

In the proportional hazards model, equating $-\log(1 - U)$ to $H(t)$ yields the variate generation formula

$$T \leftarrow H_0^{-1}\left(\frac{-\log(1 - U)}{\psi(\mathbf{z})}\right).$$

In addition to generating individual lifetimes, these variate generation techniques can be applied to point processes. A renewal process, for example, with time between events having a cumulative hazard function $H(t)$, can be simulated by using the appropriate generation formula for the two cases shown above. These variate generation formulas must be modified, however, to generate variates from an NHPP.

In an NHPP, the hazard function, $h(t)$, is analogous to the intensity function $\lambda(t)$, which governs the rate at which events occur. To determine the appropriate method for generating variates from an NHPP model which involves covariates, assume that the last event in a point process has occurred at time a. The cumulative hazard function for the time of the next event, conditioned on

Table 1.
Lifetime generation in regression survival models

	Renewal	NHPP
Accelerated life	$T \leftarrow a + H_0^{-1}(-\log(U))/\psi(\mathbf{z})$	$T \leftarrow H_0^{-1}(H_0(a\psi(\mathbf{z})) - \log(U))/\psi(\mathbf{z})$
Proportional hazards	$T \leftarrow a + H_0^{-1}(-\log(U)/\psi(\mathbf{z}))$	$T \leftarrow H_0^{-1}(H_0(a) - \log(U)/\psi(\mathbf{z}))$

survival to time a, is

$$H_{T|T>a}(t) = H(t) - H(a), \quad t \geqslant a.$$

In the accelerated life model, where $H(t) = H_0(t\psi(\mathbf{z}))$, the time of the next event is generated by

$$T \leftarrow \frac{H_0^{-1}(H_0(a\psi(\mathbf{z})) - \log(1-U))}{\psi(\mathbf{z})}.$$

Equating the conditional cumulative hazard function to $-\log(1-U)$, the time of the next event in the proportional hazards case is generated by

$$T \leftarrow H_0^{-1}\left(H_0(a) - \frac{\log(1-U)}{\psi(\mathbf{z})}\right).$$

Table 1 summarizes the variate generation algorithms for the accelerated life and proportional hazards models (the last event occurred at time a). The $1-U$ has been replaced with U in this table to save a subtraction, although the sense of the monotonicity is reversed.

The renewal and NHPP algorithms are equivalent when $a = 0$ (since a renewal process is equivalent to an NHPP restarted at zero after each event), the accelerated life and proportional hazards models are equivalent when $\psi(\mathbf{z}) = 1$, and all four cases are equivalent when $H_0(t) = \lambda t$ (the exponential baseline case) because of the memoryless property associated with the exponential distribution.

3 Generating random objects

The notion of a random "object" is quite general. We therefore limit our discussion to (1) combinatorial objects, (2) matrices, and (3) polynomials. The wider literature on generating random objects is rather diverse, ranging from a monograph on shuffling playing cards (Morris, 1998), to generating random colors, to generating random geometric objects, to generating random spawning trees (Fishman, 1996, p. 379), to generating random sequences (Knuth, 1998), to generating Markov chains (Gilks et al., 1996; Ross, 2003; Metropolis et al., 1953; Hastings, 1970).

3.1 Combinatorial objects

A combinatorial object is an object which can be placed into a one-to-one correspondence with a finite set of integers. These objects can be used for a variety of purposes (e.g., entries in a matrix, coefficients of a polynomial). The algorithms given in the subsections below are from Nijenhuis and Wilf (1978), Wilf (1989) and Devroye (1986). These authors also include algorithms for generating *all* combinatorial objects in lexicographic order, as opposed to generating *one* random combinatorial object, which is the emphasis here. We consider only random subsets, combinations, and permutations here. Algorithms for generating other combinatorial objects (e.g., a random composition of n into k parts or a random partition of an integer n) are also considered by these authors.

Generating a random subset of $\{1, 2, \dots, n\}$

Generating a random subset of the first n positive integers is straightforward since the inclusion or exclusion of each element in the subset is a Bernoulli(1/2) random variable. Thus if $a_i = 0$ if i is excluded from the subset and $a_i = 1$ if i is included in the subset, the following $\mathrm{O}(n)$ algorithm generates the $a_1, a_2, \dots, a_n$ values.

```
for $i$ from 1 to $n$
    generate $U \sim \mathrm{U}(0, 1)$
    $a_i \leftarrow \lfloor 2U \rfloor$
return $a_1, a_2, \dots, a_k$
```

In this fashion, each of the 2^n subsets is generated with probability $1/2^n$.

Generating a random combination of k integers from $\{1, 2, \dots, n\}$

The naive algorithm for generating a random combination of k integers drawn from $\{1, 2, \dots, n\}$ is to repeatedly generate random integers between 1 and n, discarding those that have been generated previously. This inefficient $\mathrm{O}(k^2)$ algorithm which generates the combination in an unsorted order can be improved on by the following $\mathrm{O}(k)$ algorithm which uses exactly k memory locations $a_1, a_2, \dots, a_k$ for storing the combination and produces sorted output.

```
$c_1 \leftarrow k$
$c_2 \leftarrow n$
$k_0 \leftarrow 0$
$i \leftarrow 1$
while $c_1 > 0$
    generate $U \sim \mathrm{U}(0, 1)$
    if $U \leqslant c_1/c_2$
        $c_1 \leftarrow c_1 - 1$
        $k_0 \leftarrow k_0 + 1$
        $a_{k_0} \leftarrow i$
    $c_2 \leftarrow c_2 - 1$
```

```
        $i \leftarrow i+1$
    return $a_1, a_2, \dots, a_k$
```

This algorithm progressively chooses each element with the appropriate probability (c_1/c_2) based on the number of elements remaining in the combination and the number remaining in the selection pool, terminating when the kth element is selected for inclusion in the combination. In this fashion, each of the $\binom{n}{k}$ combinations is generated with probability $1/\binom{n}{k}$.

Generating a random permutation of $\{1, 2, \dots, n\}$

There are exactly $n!$ permutations of the integers $\{1, 2, \dots, n\}$. An algorithm that generates each of these permutations with probability $1/n!$ is referred to as a *random shuffle*. If $n = 52$, for example, a random shuffle algorithm generates one of the permutations of an ordinary deck of playing cards. The intuitive way to generate a random shuffle of $\{1, 2, \dots, n\}$ is to (1) select an integer from $\{1, 2, \dots, n\}$, (2) select an integer from $\{1, 2, \dots, n\}$ less the integer selected in the previous step, etc. The following $O(n)$ algorithm does just that while leaving the elements of $\{1, 2, \dots, n\}$ in place. Prior to executing the algorithm, the first n integers should be stored (in any order) in $a_1, a_2, \dots, a_n$.

```
for $i$ from 1 to $n$
    generate $U \sim U(0, 1)$
    $j \leftarrow \lfloor (n-i+1)U \rfloor + i$
    $t \leftarrow a_j$
    $a_j \leftarrow a_i$
    $a_i \leftarrow t$
return $a_1, a_2, \dots, a_n$
```

The $j \leftarrow \lfloor (n-i+1)U \rfloor + i$ step is used to generate a random subscript that is equally likely among $i, i+1, \dots, n$. The three steps that follow are used to swap a_i and a_j.

This completes the discussion of algorithms for the generation of three fundamental combinatorial objects: subsets, combinations, and permutations. Other sampling schemes (e.g., sampling with replacement) or other combinatorial objects (e.g., a random partition of an integer) are considered by the authors cited at the beginning of this subsection.

3.2 Random matrices

A random matrix is an $n \times m$ array of entries that are random variables. Generating random matrices is a topic without a well-organized theory. Carmeli (1983), Deift (2000) and Mehta (2004), for example, emphasize applications of random matrices in quantum mechanics where, for example, the energy levels of a system are given by the eigenvalues of a Hermitian operator called the Hamiltonian. When the complex entries are random variables, the associated eigenvalues are also random. Rather than attempt to create a hierarchy for random matrices (e.g., the meaning of a random positive definite matrix), we focus on three examples that illustrate the wide variety of questions that can be posed concerning random matrices: (1) the generation of a particular type of matrix known as a "correlation matrix", (2) the distribution of the number of real eigenvalues associated with a matrix of independent standard normal random variables, and (3) the distribution of the determinant of an M-matrix with U(0, 1) entries.

Generating correlation matrices

A correlation matrix is a symmetric, positive semidefinite matrix with 1's on the diagonal. The (i, j)th element of such a matrix gives the correlation between the ith and jth elements of a random vector. Marsaglia and Olkin (1984) consider the generation of random correlation matrices under the following three scenarios: (1) random correlation matrices with a given mean, (2) random correlation matrices of the form TT', and (3) random correlation matrices with given eigenvalues. We consider the first case.

Let C be a given correlation matrix and X be a random symmetric matrix with zeros on the diagonal and random variables with means of zero for the off-diagonal elements. The matrix $C + X$ is a random correlation matrix with expected value C if and only if the eigenvalues of $C+X$ are nonnegative. Thus X is a perturbation of C that can be used to generate the random correlation matrix $C + X$ under the appropriate restrictions on X. This setting provides a variety of methods for generating random correlation matrices. Here is one such algorithm:

Set up: Compute λ, the smallest eigenvalue of the $n \times n$ matrix C.

Marginal: Generate the upper-triangular elements of X from a radially symmetric distribution in (or on) the unit $n(n-1)/2$-sphere. Letting $x_{ij} = x_{ji}$ for $i \neq j$, the matrix $C + \lambda X/\sqrt{2}$ is a random correlation matrix with expected value C.

Example. Consider the generation of a random correlation matrix when $n = 3$. One way to generate the random elements of the perturbation matrix X is to generate values on the surface of the unit sphere. Using the transformation from spherical coordinates to rectangular coordinates:

$$x_{12} = \rho \sin\phi \cos\theta, \qquad x_{13} = \rho \sin\phi \sin\theta, \qquad x_{23} = \rho \cos\phi,$$

where $\theta \sim \mathrm{U}(0, 2\pi)$, $\phi \sim \mathrm{U}(0, \pi)$ and $\rho = 1$. Using the third random number stream, a perturbation matrix X can be generated in S-Plus (S-Plus is a registered trademark of Insightful Corporation) with the statements:

```
set.seed(3)
x <- matrix(0, 3, 3)
theta <- runif(1, 0, 2 * pi)
phi <- runif(1, 0, pi)
x[1, 2] <- sin(phi) * cos(theta)
x[1, 3] <- sin(phi) * sin(theta)
x[2, 3] <- cos(phi)
```

```
    x[2, 1] <- x[1, 2]
    x[3, 1] <- x[1, 3]
    x[3, 2] <- x[2, 3]
```

yielding

$$X = \begin{bmatrix} 0.0000000 & -0.3956487 & -0.8461130 \\ -0.3956487 & 0.0000000 & 0.3571483 \\ -0.8461130 & 0.3571483 & 0.0000000 \end{bmatrix}.$$

The associated random correlation matrix is thus $C + \lambda X/\sqrt{2}$.

Ghosh and Henderson (2003) give another method for generating correlation matrices.

Number of real eigenvalues

Some random matrices have analytic results that may preclude the need for simulation. This paragraph outlines one such case. An $n \times n$ matrix with independent standard normal entries, for example, has an expected number of real eigenvalues E_n given by Edelman et al. (1994):

$$E_n = \begin{cases} 1 + \sqrt{2}\sum_{k=1}^{(n-1)/2} \frac{(4k-3)!!}{(4k-2)!!}, & n \text{ odd}, \\ \sqrt{2}\sum_{k=0}^{n/2-1} \frac{(4k-1)!!}{(4k)!!}, & n \text{ even}, \end{cases}$$

where the double factorial (also known as the semifactorial) notation is defined by

$$n!! = \begin{cases} 1 \times 3 \times 5 \times \cdots \times n, & n \text{ odd}, \\ 2 \times 4 \times 6 \times \cdots \times n, & n \text{ even}. \end{cases}$$

The authors also show that E_n asymptotically converges to $\sqrt{2n/\pi}$ as $n \to \infty$ and that the distribution of the value of a real normalized eigenvalue (the eigenvalue divided by $\sqrt{n}$) of such a matrix converges in distribution to a $\mathrm{U}(-1, 1)$ distribution. Other distributions for the entries of the matrix or correlated entries would likely require Monte Carlo simulation to determine the expected number of real eigenvalues or the distribution of a real normalized eigenvalue.

Example. The analytic result indicates that the expected number of real eigenvalues of a 3×3 matrix of independent standard normal random variables is $E_3 = 1 + \sqrt{2}/2 \cong 1.71$. The following Monte Carlo simulation S-Plus code verifies the analytic result (it yields an estimated expected number of real eigenvalues as 1.738, 1.698 and 1.676 for random number streams 3, 4 and 5, which corresponds to an approximate 95% confidence interval of $1.63 < E_3 < 1.78$ based on the assumption of normal sampling) and can easily be modified for other independent or dependent matrix entries where no analytic result exists.

```
set.seed(3)
total.real <- 0
for (i in 1:1000) {
  x <- matrix(rnorm(9), 3, 3)
  total.real <- total.real
                + sum(Im(eigen(x)values) == 0)
}
total.real / 1000
```

Determinant of an M-matrix

Finally, we consider the following question: if the elements of a 3 × 3 real matrix are independent random numbers with positive diagonal elements and negative off-diagonal elements, what is the probability that the matrix has a positive determinant? That is, find the probability that

$$\begin{vmatrix} +u_{11} & -u_{12} & -u_{13} \\ -u_{21} & +u_{22} & -u_{23} \\ -u_{31} & -u_{32} & +u_{33} \end{vmatrix} > 0,$$

where the u_{ij}'s are independent random numbers. This question is rather vexing analytically due to the appearance of some of the random numbers multiple times in the determinant. This question is of interest to matrix theorists as it is an example of a probability that cannot be calculated easily. A positive determinant in this case is equivalent to the matrix being of a special type called an *M-matrix* (Horn and Johnson, 1990). A positive determinant in a matrix with this particular sign pattern (positive diagonal elements and negative off-diagonal elements) corresponds to having all three 2 × 2 principal minors (determinants of sub-matrices determined by deleting the same numbered row and column) being positive.

The implementation of the Monte Carlo simulation to estimate the probability of a positive determinant is straightforward. Random matrices are generated in a loop, counting the number of these matrices that have a positive determinant and returning the ratio of the count of the matrices that have a positive determinant to the number of replications. In order to estimate the probability with some precision, it is reasonable to make one long run. For the linear congruential generator $g(x) = ax \bmod m$ with $a = 48271$, $m = 2^{31} - 1$, and a seed of $x_0 = 123456789$ (see Chapter 3), and 200,000,000 replications, the program returns an estimated probability of a positive determinant as 0.050203. Two other simulation runs with different random number generators indicate that only three digits are reasonable to report: the estimated probability of a positive determinant is 0.0502.

3.3 Random polynomials

Like random matrices, the diversity of questions that can be asked concerning random polynomials is also very large. Edelman and Kostlan (1995), for example, use elementary geometric arguments to show that the expected number of real zeros E_n of a degree n polynomial with independent standard normal coefficients is given by the expression

$$E_n = \frac{4}{\pi} \int_0^1 \sqrt{\frac{1}{(1-t^2)^2} - \frac{(n+1)^2 t^{2n}}{(1-t^{2n+2})^2}}\, \mathrm{d}t.$$

The authors show that for a seemingly minor variation in this problem (the independent, normally distributed coefficients have means 0 and variance $\binom{n}{i}$ for the ith coefficient), then the random polynomial has $E_n = \sqrt{n}$ expected real zeros. They also extend these results for arbitrary and multivariate normal distributions.

Example. The Monte Carlo simulation S-Plus code below verifies that $E_2 = \sqrt{2} \cong 1.41421$ for quadratic equations, and can be modified to accommodate more general random coefficients of the polynomial. The program below returns 1.41394, 1.40886 and 1.41636 as estimates for the expected number of real roots for random number streams 3, 4 and 5, resulting in an approximate 95% confidence interval of $1.40 < E_2 < 1.42$ based on the assumption of normal sampling, which verifies the theoretical result

```
set.seed(3)
num.roots <- 0
for (i in 1:100000) {
  aa <- rnorm(1, 0, 1)
  bb <- rnorm(1, 0, sqrt(2))
  cc <- rnorm(1, 0, 1)
  if (bb ^ 2 - 4 * aa * cc > 0)
    num.roots <- num.roots + 2
}
num.roots / 100000
```

Acknowledgements

The author gratefully acknowledges the helpful comments from the editors and a referee.

References

Allison, P.D. (1984). *Event History Analysis: Regression for Longitudinal Event Data*. Sage Publications, Beverly Hills, CA.

Arkin, B.L., Leemis, L.M. (2000). Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process from overlapping realizations. *Management Science* 46 (7), 989–998.

Banks, J., Carson II, J.S., Nelson, B.L., Nicol, D.M. (2005). *Discrete-Event System Simulation*, 4th edition. Prentice-Hall, Englewood Cliffs, NJ.

Carmeli, M. (1983). *Statistical Theory and Random Matrices*. Marcel Dekker, New York.

Casella, G., Berger, R.L. (2002). *Statistical Inference*, 2nd edition. Duxbury, Pacific Grove, CA.

Çinlar, E. (1975). *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.

Cox, D.R., Isham, V. (1980). *Point Processes*. Chapman and Hall, New York.

Cox, D.R., Oakes, D. (1984). *Analysis of Survival Data*. Chapman and Hall, New York.

David, H.A., Moeschberger, M.L. (1978). *The Theory of Competing Risks*. Macmillan, New York.

Deift, P. (2000). *Orthogonal Polynomials and Random Matrices: A Riemann–Hilbert Approach*. American Mathematical Society, Providence, RI.

Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag, New York.

Edelman, A., Kostlan, E. (1995). How many zeros of a random polynomial are real? *Bulletin of the American Mathematical Society* 32 (1), 1–37.

Edelman, A., Kostlan, E., Shub, M. (1994). How many eigenvalues of a random matrix are real? *Journal of the American Mathematical Society* 7, 247–267.

Fishman, G.S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, New York.

Ghosh, S., Henderson, S.G. (2003). Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transaction on Modeling and Computer Simulation* 13, 276–294.

Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, New York.

Gross, D., Harris, C.M. (1985). *Fundamentals of Queueing Theory*, 2nd edition. Wiley, New York.

Hastings, W.K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 97–109.

Henderson, S.G. (2003). Estimation of nonhomogeneous Poisson processes from aggregated data. *Operations Research Letters* 31 (5), 375–382.

Horn, R.A., Johnson, C.R. (1990). *Topics in Matrix Analysis*. Cambridge University Press, Cambridge.

Knuth, D.E. (1998). *The Art of Computer Programming, volume 2: Seminumerical Algorithms*, 3rd edition. Addison–Wesley, Reading, MA.

Kuhl, M.E., Wilson, J.R., Johnson, M.A. (1997). Estimating and simulating Poisson processes having trends and multiple periodicities. *IIE Transactions* 29, 201–211.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Lawless, J.F. (2003). *Statistical Models and Methods for Lifetime Data*, 2nd edition. Wiley, Hoboken, NJ.

Lee, S., Wilson, J.R., Crawford, M.M. (1991). Modeling and simulation of a nonhomogeneous Poisson process having cyclic behavior. *Communications in Statistics – Simulation and Computation* 20 (2/3), 777–809.

Leemis, L.M. (1987). Variate generation for the accelerated life and proportional hazards models. *Operations Research* 35 (6), 892–894.

Leemis, L.M. (1991). Nonparametric estimation of the intensity function for a nonhomogeneous Poisson process. *Management Science* 37 (7), 886–900.

Leemis, L.M. (2004). Nonparametric estimation and variate generation for a nonhomogeneous Poisson process from event count data. *IIE Transactions* 36 (12), 1155–1160.

Lewis, P.A.W., Shedler, G.S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly* 26 (3), 403–413.

Marsaglia, G., Olkin, I. (1984). Generating correlation matrices. *SIAM Journal of Scientific and Statistical Computing* 5 (2), 470–475.

Meeker, W.Q., Escobar, L.A. (1998). *Statistical Methods for Reliability Data*. Wiley, New York.

Mehta, M.L. (2004). *Random Matrices*, 3rd edition. Elsevier, London.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics* 21, 1087–1091.

Morris, S.B. (1998). *Magic Tricks, Card Shuffling and Dynamic Computer Memories*. The Mathematical Association of America, Washington, DC.

Nelson, B.L. (2002). *Stochastic Modeling: Analysis and Simulation*. Dover Publications, Mineola, NY.

Nelson, B.L., Ware, P., Cario, M.C., Harris, C.A., Jamison, S.A., Miller, J.O., Steinbugl, J., Yang, J. (1995). Input modeling when simple models fail. In: Alexopoulos, C., Kang, K., Lilegdon, W.R., Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 93–100.

Nelson, W.B. (2003). *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*. ASA/SIAM, Philadelphia.

Nijenhuis, A., Wilf, H.S. (1978). *Combinatorial Algorithms for Computers and Calculators*, 2nd edition. Academic Press, New York.

Pritsker, A.A.B., Martin, D.L., Reust, J.S., Wagner, M.A., Wilson, J.R., Kuhl, M.E., Allen, M.D., Daily, O.P., Harper, A.M., Edwards, E.B., Bennett, L.E., Roberts, J.P., Burdick, J.F. (1995). Organ transplantation policy evaluation. In: Alexopoulos, C., Kang, K., Lilegdon, W.R., Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1314–1323.

Resnick, S.I. (1992). *Adventures in Stochastic Processes*. Birkhäuser, Boston, MA.

Rigdon, S.E., Basu, A.P. (2000). *Statistical Methods for the Reliability of Repairable Systems*. Wiley, New York.

Ross, S.M. (2003). *Introduction to Probability Models*, 8th edition. Academic Press, New York.

Schoenberg, F.P. (2003). Multidimensional residual analysis of point process models for earthquake occurrences. *Journal of the American Statistical Association* 98 (464), 789–795.

White, K.P. (1999). Simulating a nonstationary Poisson process using bivariate thinning: The case of "typical weekday" arrivals at a consumer electronics store. In: Farrington, P., Nembhard, P.A., Sturrock, H.B., Evans, G.W. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 458–461.

Whitt, W. (2002). *Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues*. Springer-Verlag, New York.

Wilf, H.S. (1989). *Combinatorial Algorithms: An Update*. SIAM, Philadelphia, PA.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13007-8

Chapter 7

Implementing Representations of Uncertainty

W. David Kelton

Department of Quantitative Analysis and Operations Management,
University of Cincinnati, USA
E-mail: david.kelton@uc.edu

Abstract

Chapters 3–6 discussed generating the basic random numbers that drive a stochastic simulation, as well as algorithms to convert them into realizations of random input structures like random variables, random vectors, and stochastic processes. Still, there could be different ways to implement such methods and algorithms in a given simulation model, and the choices made can affect the nature and quality of the output. Of particular interest is precisely how the basic random numbers being generated might be used to generate the random input structures for the model. This chapter discusses these and related issues and suggests how implementation might best proceed, not only in particular simulation models but also in simulation software.

1 Introduction

The preceding chapters in Part II of this volume (Chapters 3–6) discussed details of generating the basic random numbers at the root of stochastic simulations, and the ensuing methods for generating the various kinds of random structures (variates, multivariate input processes, arrival processes, random lifetimes, and other objects) that are used directly to drive the simulation.

The purpose of this (very) short chapter is merely to raise some issues, and suggest some potential resolutions to them, for precisely how such generation might best be implemented in the context of a large and complex simulation, especially in terms of the low-level assignment of specific sequences of random numbers to generating the various random structures listed above. You might want to read this if you are a simulation analyst interested in effecting some of these ideas (software permitting), or if you are a simulation-software developer considering implementing some of this in your products, either as defaults or as user-controlled options, to help your customers do a better job with their simulation studies.

The implementation of random-number generation for various random structures matters for reasons of both *validity* of the output (i.e., accuracy), as well as for its *precision*:

- *Validity*. If, as is quite possible (if not almost certain), sequences of random numbers are re-used in an uncontrolled or inappropriate way (most likely unknowingly), the simulation's very validity can be threatened. The reason is that many variate- and process-generation schemes rely on independent draws from the underlying random-number generator (as discussed in Chapters 4–6), and if care is not taken to eliminate the possibility of uncontrolled overlapping or re-using sequences of random numbers, these variate- and process-generation methods can become invalid – and hence the simulation's results could also be invalid.
- *Precision*. You could easily lose an opportunity to improve your simulation's efficiency (either computational or statistical, which can be viewed as the same thing) if you pass up what might be some relatively easy (maybe even automatic, as described in Section 5) steps involving intelligent re-use of random numbers. Now the preceding bullet point railed against re-use of random numbers, claiming that doing so could invalidate your simulation, and now it's being suggested that doing so can improve it (while not harming its validity). The difference is that now what's being considered is controlled and intelligent re-use, as opposed to uncontrolled or even unconscious re-use. A more detailed treatment of the probability and statistics behind this possibility, with an example, is in Section 4.

Section 2 discusses how the underlying random-number generator might be set up in terms of accessing and using it. Section 3 considers how such a random-number-generator structure might be exploited to organize and control how input variates, vectors, processes, and other objects are generated in the interest of improving the simulation's efficiency and thus value. In Section 4 an application of these ideas, to inducing desirable correlation to reduce simulation-output variance, is described along with an example. Finally, Section 5 offers some (personal) opinions about how these ideas might be implemented in high-level simulation-modeling software, which would help everyone simulate better.

Some of this material (and additional related issues, such as robust and automated integration into software of both input modeling and output analysis) appeared in Kelton (2001).

2 Random-number generation

Use of a modern, statistically valid and long-period random-number generator is absolutely critical to any simulation study (see Chapter 3 for a full treatment of this topic). Somewhat alarmingly, we now have a severe mismatch between computer-hardware capabilities and the characteristics of some older random-number generators still in common use.

First and foremost, random-number generators must be of high statistical quality, and produce a stream of numbers that behave as though they were independent and uniformly distributed on the unit interval, even when scrutinized closely for these properties by powerful structural and statistical tests. Given the speed of computers several decades ago, generators were developed at that time that were adequate to "fool" these tests up to the discriminatory power of the day. One aspect of this is the *cycle length* (or *period*) p of a generator, which is the number of random numbers before the generator repeats itself (as all algorithmic generators eventually will). Starting from some fixed deterministic initialization, the complete sequence of generated random numbers would be $u_1, u_2, \ldots, u_p$ and that is all, since $u_{p+1} = u_1, u_{p+2} = u_2, \ldots,$ and in general $u_{kp+i} = u_i$ for $k \in \{1, 2, \ldots\}$. In the 1950s, 1960s, and 1970s, generators with cycle length p around 2^{31} (on the order of 10^9) were developed; this cycle length resulted from the word length of fixed-point integers in computers at that time. These were adequate for many years, and were in wide use. Unfortunately, they are *still* in wide use even though computer speeds have increased to the point that any garden-variety PC can completely exhaust this cycle in just a few minutes! At that point the stream cycles and produces exactly the same "random" numbers in exactly the same order, with obvious potentially deadly implications for the integrity of simulations' results. This dirty little software scandal (including simulation software) is one that few users (or software developers) seem to be aware of, or even seem to care about. But it is easily avoidable. People now have developed and coded algorithms for extremely long-period generators, with cycle lengths p on the order of 10^{57} or more, which display superb statistical behavior; see Chapter 3 for a general discussion, or L'Ecuyer et al. (2002) for a specific example with downloadable code. Their cycle lengths will continue to be "long", even under Moore's "law", for a few centuries to come. And their speed is comparable to the poor little old generators from decades ago. It is hard to think of any excuse at all for not implementing these kinds of generators immediately, especially in commercial simulation software.

Another highly desirable aspect of random-number generators is the ability to specify separate *streams* of the generator, which are really just (long) contiguous subsegments of the entire cycle, and to make these readily available, perhaps via on-the-fly object-oriented instantiation as opposed to storing static seed vectors to initiate each stream. With s streams of length l each ($sl = p$), we could re-index the generated random numbers as u_{ji} being the ith random number in stream j, for $i \in \{1, 2, \ldots, l\}$ and $j \in \{1, 2, \ldots, s\}$. These streams can be further subdivided into *substreams*, and so on, for multidimensional indexing (u_{jki} is the ith random number from substream k within stream j) and assignment of separate and independent chunks of random numbers to separate activities in the simulation; the importance of this is really in variance reduction via correlation induction, as discussed in Section 4, as well as in Banks et al. (2005, Chapter 12) and Law and Kelton (2000, Chapter 11). The specific generator in L'Ecuyer et al. (2002) provides a very large number of very long streams and substreams (i.e., two-dimensional indexing).

As an example of the utility of such multidimensional stream and substream indexing, you could assign a stream (first index j) to generating, say, service times of a class of parts at a workcenter, and within that stream you would (automatically, one would hope) move to the next substream (second index k) for each new replication of the simulation run. The importance of moving to a new substream with each replication is that, in comparing alternate *scenarios* of the model (e.g., change some input parameters or model logic), different scenarios will in general consume different numbers of random numbers for this purpose in a replication, and moving to the next substream for the next replication will ensure that in replications k subsequent to the first, the same random numbers will still be used for the same purpose; see the discussion of random-number *synchronization* in Section 4, Banks et al. (2005, Chapter 12), or Law and Kelton (2000, Chapter 11). Another way to ensure synchronization throughout multi-replication comparative runs, without substreams, would be to advance all stream numbers at the beginning of each replication after the first by the number of streams being used; with the astronomical number of streams that should be available, there would be no worries about running out of streams. It is easy to imagine the need for stream indexing of dimension higher than two in order to ensure synchronization in more complex projects involving designed experiments or comparing and ranking multiple scenarios (see Chapter 17) – u_{jkdi} is the ith random number for the jth source of randomness during the kth replication of design point d in a designed simulation experiment.

Obviously, being able to do this reliably and generally requires extremely long-period underlying generators, which, as pointed out, now exist. And we also have easy and fast methods to create and reference such multidimensional streams (and the length of even the substreams is astronomically long so there are no worries about exhausting or overlapping them); see, for example L'Ecuyer et al. (2002) for a package based on a generator with cycle length 10^{57}, streams and substreams (i.e., two-dimensional indexing), as well as downloadable code for implementation.

3 Random-structure generation

With multidimensional stream indexing based on a high-quality, long-period underlying random-number generator, it then becomes possible (ideally, automatic) to assign separate streams of random numbers to separate points in the model where uncertainty in the input is required. As mentioned, the reason for doing so is to ensure, so far as possible given the structure of the model(s), that in simulation studies comparing alternate scenarios of a general base model, the same random numbers are used for the same purposes across the scenarios, to maintain the best synchronization and thus improve the performance of correlation-induced variance-reduction techniques (see Section 4).

Though precisely how this is done will always have some dependence on the model, there are generally two approaches, by *faucets* and by *body art*:

- *Faucets*. Identify the activities in the model where uncertainty is an input. Examples include interarrival times between successive customers, their service times at different stations, their priorities, and points at which probabilistic branching occurs. Assign a separate stream in the random-number generator to each. This is then like having a separate faucet at each such point, pouring out random numbers from a separate reservoir. True, the reservoirs are finite, but with the right underlying random-number generator broken appropriately into streams and substreams, etc., you do not need to worry about draining any reservoirs during your lifetime.
- *Body art*. Identify all the uncertain things that might happen to an entity (like a customer in a queueing model) during its life in the model, and write them on the entity when it is born; these are sometimes called *attributes* of the entity. Examples would include service times, pre-ordained probabilistic branching decisions, and the number of clones made if there is downstream copying of the entity. Use nonindelible ink to allow for possible reassignment of these attributes during the entity's life.

Neither faucets nor body art provides the universal or uniformly best way to go, since, as mentioned, particular model logic might render one or the other difficult or impossible (or meaningless).

It seems that body art would, in general, provide the best and strongest synchronization, since it is more in keeping with the principle that, for all scenarios, we want the "same" customers showing up at the same times (by the "same" customers we mean that they are identical in terms of exogenous traits like service requirements, arrival times, branching decisions, etc.). However, it also makes the entities bloated (thus consuming more memory) in terms of their having to drag around all these attributes when most are not needed until later (maybe some will never be needed).

Faucets are easy to implement (maybe they should be automatic in the software, which would automatically increment the stream/faucet index every time the modeler puts a source of randomness into the model), and in general provide a level of synchronization that is certainly better than nothing. However, given the vagaries of what might happen in a simulation (e.g., branching decisions), they would not appear necessarily to provide the "same-customer" level of synchronization that might be achieved by body art, but this is model-dependent (see the example in Section 4).

Another aspect of how simulations are actually coded (or how the simulation software being used was coded) that can affect the level of synchronization is the method by which random variates (and vectors and processes) are generated, if there is a choice; this was discussed in Chapters 4–6 and many specific methods and references are in Banks et al. (2005, Chapter 8) and Law and Kelton (2000, Chapter 8). The opinion here is that inversion of the cumulative distribution function (or of a process analogue of it, such as the cumulative rate function for generating realizations of a nonstationary Poisson process, as in Çinlar, 1975) is preferable since:

- the number of random numbers consumed per variate (or vector) is fixed, thus making it easier to ensure synchronization,
- the sign and strength of the correlation between the basic random numbers and the objects realized is protected, and
- it generally uses the fewest number of random numbers (though this is the least important reason since in most complex dynamic systems simulation random-number generation accounts for only a small proportion of the overall execution time).

Now inversion is not universally applicable via simple closed-form formulas, but in some such cases a numerical method to effect the inversion is available and might be considered in order to realize the benefits above. True, this carries a computation-time penalty, but that might be justified by efficiency gains in the statistical sense (and thus ultimately in the computational sense as well if a specific level of precision in the output is sought). This is a consideration that might be brought to bear on that part of simulation-software design concerned with generation of random structures for input to a user's model.

4 Application to variance reduction

As mentioned above, one of the main applications and benefits of careful implementation of random-structure generation is in *variance reduction*. By taking a little care in allocation of random-number streams in the simulation model, it is often possible to achieve better output precision without additional computational effort.

Though not the only variance-reduction technique, one of the most accessible and widely used is *common random numbers* (CRN), also called *correlated sampling* or *matched pairs*. It applies not when simulating a single system configuration, but rather when comparing two (or more) alternative configurations, or *scenarios*, and uses the same random numbers to drive all scenarios, though care must be taken to do so wisely in order to get the most benefit.

One motivation for CRN is common sense – to compare alternative scenarios it seems best to subject them to the same "external" conditions. This way, any observed output differences could be more reliably attributed to differences in the scenarios rather than to the different external conditions that happened to occur in the runs, since such external differences are minimized.

This common sense is backed up by basic probability. If Y_{A} and Y_{B} are output responses from scenarios A and B, we might seek to estimate the expected difference $\mathrm{E}(Y_{\mathrm{A}} - Y_{\mathrm{B}})$ between the scenarios from simulation output data by $Y_{\mathrm{A}} - Y_{\mathrm{B}}$. One measure of the quality of this estimator, in terms of precision, is $\mathrm{Var}(Y_{\mathrm{A}} - Y_{\mathrm{B}}) = \mathrm{Var}(Y_{\mathrm{A}}) + \mathrm{Var}(Y_{\mathrm{B}}) - 2\,\mathrm{Cov}(Y_{\mathrm{A}}, Y_{\mathrm{B}})$. If the covariance is zero, as would happen if the runs were independent, we get a higher variance than if we could make this covariance positive, as we would anticipate if the external conditions were controlled to be mostly the same across the simulations of A and B (barring pathological model behavior, we would expect that, say, heavy customer demands would cause A and B to respond in the same direction). Of course, this thinking goes through with multiple replications on a replication-by-replication basis, and the variance reduction propagates through to the sample means. One way to try to induce such positive covariance in comparative simulations is CRN.

For instance, a facility receives blank parts and processes them. About a fourth of the blank parts need repair before processing. There are currently two separate external sources; from each source, parts arrive one at a time with interarrival times that are exponentially distributed with mean one minute. There is a single repair person, with repair times' being exponential with mean 1.6 minutes. There are two processing operators, both "fed" by a single queue, with processing times' being exponential with mean 0.8 minute. All queues are first-come, first-served, and all probabilistic inputs (interarrival times, service times, and whether a part needs repair) are independent. The simulation starts empty and idle and runs for 24 hours. From 50 replications, a 95% confidence interval for the expected time in system of parts is 3.99 ± 0.19 minutes, and for the expected number of parts present is 8.00 ± 0.40.

Under consideration is taking on a third but smaller source of blank parts, with interarrival times' being exponential with mean five minutes, but otherwise the same in terms of service times and proportion that need repair. Adding this to the model and replicating 50 times, 95% confidence intervals on expected time in system and expected number of parts present are 6.79 ± 0.60 minutes and 15.01 ± 1.39 parts. It is hardly surprising that both went up, but it would have been hard to quantify by how much without a simulation.

How should we make statistically valid statements about the changes in the model responses? For example, are the changes statistically significant? One way is to pair up the results, replication by replication, subtract, and compute a confidence interval on the expected difference (assuming normal-theory statistics are valid, this is the *paired t* method discussed in any basic statistics book). Subtracting in the direction (original model)–(second model), we get 95% confidence intervals of −2.80 ± 0.59 minutes for the difference between expected times in system, and −7.00 ± 1.35 for the difference between expected numbers of parts present; since neither confidence interval contains zero, the differences are statistically significant.

Exactly what random numbers were used (or should have been used) in the comparison? We just ignored the whole issue above and used the default stream for everything in both scenarios, initialized at the same point. Thus, the results from the two scenarios arose from the same sequence of random numbers (the lengths of the sequences were very likely different, but there was substantial overlap) so are not independent, though fortunately the paired t method is still valid in this case. This is probably how the vast majority of simulation comparisons are done, so the same random numbers are in fact used, but not in a controlled way – call this the *default* approach. (We did, however, advance to the next substream for each new replication in both scenarios, as discussed in Section 2, so the random numbers started off the same way in each replication.)

To see if the default approach got us any variance reduction, we reran the comparison, again with 50 replications per scenario, using one stream for everything in A, and a different stream throughout B (and both different from the one used in the default comparison). Again, substreams were advanced for each replication. The 95% confidence intervals on the differences were -2.49 ± 0.47 minutes for time in system and -6.33 ± 1.11 for mean number of parts present, which appear to be of about the same precision as in the default approach. Indeed, a formal F test for equality of variances (on which the confidence-interval half-lengths are based) failed to find a statistically significant difference. So if default CRN achieved any variance reduction at all it was vanishingly small since there was no attempt to synchronize random-number use, even though essentially the same random numbers were used across A and B.

We next synchronized the comparison by faucets, assigning separate streams to each of the random inputs (one for each arrival source, one for the repair decision, one for repair times, and one for processing times), and using the same stream assignments for both A and B. Substreams, as discussed in Section 2, were used to maintain synchronization in replications subsequent to the first. From 50 replications, the confidence intervals on the differences were -2.32 ± 0.29 minutes for time in system and -5.99 ± 0.70 for mean number of parts present, both noticeably tighter than the default and independent approaches, indicating that faucets produced enough commonality in the external conditions to sharpen the comparison. Formally, the F test for equality of variances, in comparison with both the default and independent approaches was highly significant, confirming that synchronizing by faucets did reduce variance. Structurally, though, in this model faucets do not perfectly synchronize the repair decision on a part-by-part basis; this decision is made after parts arrive, and in scenario B additional parts are mixed into the incoming flow yet the sequence of repair decisions is the same. Thus, parts from the original two inputs are not all getting the same repair decisions they got in scenario A.

So finally we synchronized by body art, using separate streams to pre-assign as attributes to each part upon its arrival whether it needs repair, its repair time (which was ignored 3/4 of the time), and its processing time; the input processes used still separate streams, and the same streams were used for these assignments in A and B (all streams here were different from those used above, so all these experiments are independent). From 50 replications this yielded the tightest confidence intervals of all, -2.33 ± 0.25 minutes for time in system and -5.98 ± 0.59 for mean number of parts present. However, an F test for equality of variances did not signal a statistically significant difference in comparison with faucets synchronization; even though body art makes more intuitive sense for this model than do faucets (due to the repair decision, as described at the end of the preceding paragraph), the impact of body art vs. faucets is evidently minor.

There is no guarantee that CRN will always induce the desired positive covariance and attendant variance reduction, and "backfiring" counterexamples can be constructed (see Wright and Ramsay, 1979) where CRN induces negative covariance and thus increases the variance in the comparison. However, it is generally believed that such examples are rare and basically pathological, so that CRN, especially with care taken to synchronize random-number use as appropriate to the model and scenarios, is usually effective, though just how effective is generally model-dependent.

As mentioned, CRN is but one of several variance-reduction techniques that rely, in one way or another, on using and re-using random numbers in a controlled manner to induce correlation. Other methods such as *antithetic variates* and *control variates* are discussed in, for example, Bratley et al. (1987, Chapter 2) or Law and Kelton (2000, Chapter 11).

5 Conclusions and suggestions

This is an area where it is relatively easy to do a good job, so it is hard to think of justifications for not doing this well. If it were difficult, say, to use a good underlying random-number generator or to ensure reasonably effective synchronization for variance reduction, then it would be understandable (though still incorrect, or at least inefficient) if people avoided doing so.

So simulation analysts should take a little care up front in their projects (and maybe ask the right questions of their software vendors). A little effort here will repay substantial dividends later in terms of accuracy and precision of the output, and thus in terms of the quality of conclusions and decisions as well.

And simulation-software vendors have a responsibility to ensure that they are doing at least the basics right. An opinion about what constitutes "the basics" is (approximately in decreasing order of importance):

1. If you are still using an old random-number generator of period on the order of 2^{31}, replace it immediately with one of the more recent generators with (*much*) longer period and (much) better statistical properties. If you are worried about your users' getting upset about suddenly getting

different results from their old models, then they (and maybe you) are in denial about the fundamental fact of randomness in simulation output.

2. Provide very widely spaced streams in the random-number generator, such that the user can specify the stream to be used. By the way, the old de facto standard of spacing seeds 100,000 random numbers apart is no longer even close to adequate. Use the literature on random-number generators (see Chapter 3) to choose the seeds or seeding method (and the generator itself).
3. Within the streams, provide very widely spaced substreams (that need not be user-addressable) to which each new replication of the simulation is advanced automatically, as discussed in Section 2. The spacing between consecutive substreams (and thus streams) should be (and *can* be) so large that many years of computing would be required to exhaust them. And the number of streams and substreams should be (and *can* be) so large that no simulation project could ever use them up.
4. As a default (that could be overridden by a savvy user with a *very* good reason), move to the next random-number stream every time the user specifies generation of a random object in the model. For instance, every place in the model where generation from a standard univariate probability distribution is called for, there will be a unique stream assigned to it. This would automatically implement synchronization by faucets as described in Section 2. Do not worry about having enough streams, because the underlying random-number generator and the stream/substream structure ensure that no human can build a model that will use them all up.
5. Include support for modeling (estimating) and generating from nonstationary Poisson processes. Almost 25 years of personal experience suggests that these are present in many real systems (even in student projects of limited scope and complexity) and flattening them out, even to the correct aggregate mean rate, does serious damage to such models' validity – imagine an urban-freeway simulation with twice-daily rush hours flattened to a mean that includes 3AM. Recent work on this includes Kuhl et al. (2006) and Leemis (2004).

While the wish list above contains only items that can be implemented right now, there is at least one other modeling capability that would seem to be worth considering, though maybe having it fully automated is not, at present, realistic. Current practice for specifying input probability distributions (or, more generally, input random objects) goes something like this: collect data from the field, fit a distribution to those data using software that comes with the modeling package or separate third-party software, translate the results into the correct syntax for your modeling package, and then type or paste it into your model wherever it belongs. While the syntax-translation part of this sequence is commonly automated, the rest is not – but maybe it could be. The result would be not only streamlining of this process, but also automatic updating as things change and the field data begin to look different. The modeling software could establish dynamic links between the places in the model where these input-process distributions are needed and the fitting software, and indeed back to the files containing the raw field data themselves, thus making the whole fitting process transparent to the user (and rendering the fitting software invisible). The user would tell the simulation model where the data set is on some input feature, and it would automatically link to the fitting software, analyze the data, and give the results back to the simulation model. Clearly, there would have to be some kind of escape from this automation should problems arise during the fitting phase (e.g., none of the "standard" probability distributions provides an adequate fit to the data), though use of empirical distributions, as suggested by Bratley et al. (1987) could provide something of an automated escape in the case of simple univariate input distributions. This kind of dynamic integration is now common in office-suite software (e.g., updated spreadsheet data automatically update a chart in a word-processing document or presentation or web page), and it could be common in simulation software as well, but would require better integration of activities within simulation software than we now have.

In summary, there are a few nitty-gritty, low-level items almost down at the programming level that should be attended to in order to promote simulation-model building that is more robustly accurate, as well as possibly considerably more efficient. At present, the burden for most of this falls on the user (at least to check that the modeling software is doing things right), but much of this burden could be shifted to software designers to enhance their products' value, even if somewhat unseen to most users.

References

Banks, J., Carson II, J.S., Nelson, B.L., Nicol, D.M. (2005). *Discrete-Event System Simulation*, 4th edition. Prentice-Hall, Upper Saddle River, NJ.

Bratley, P., Fox, B.L., Schrage, L.E. (1987). *A Guide to Simulation*, 2nd edition. Springer-Verlag, New York.

Çinlar, E. (1975). *Introduction to Stochastic Processes*. Prentice-Hall, Englewood Cliffs, NJ.

Kelton, W.D. (2001). Some modest proposals for simulation software: Design and analysis of experiments. In: *Proceedings of the 34th Annual Simulation Symposium*. Advanced Simulation Technologies Conference Seattle, WA. IEEE Computer Society, New York, pp. 237–242, keynote address.

Kuhl, M., Sumant, S., Wilson, J. (2006). An automated multiresolution procedure for modeling complex arrival processes. *INFORMS Journal on Computing* 18, 3–18.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

L'Ecuyer, P., Simard, R., Chen, E., Kelton, W. (2002). An object-oriented random-number package with many long streams and substreams. *Operations Research* 50, 1073–1075.

Leemis, L. (2004). Nonparametric estimation and variate generation from event count data. *IIE Transactions* 36, 1155–1160.

Wright, R., Ramsay, T. (1979). On the effectiveness of common random numbers. *Management Science* 25, 649–656.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13008-X

Chapter 8

Statistical Estimation in Computer Simulation

Christos Alexopoulos

School of Industrial and Systems Engineering, Georgia Institute of Technology,
Atlanta, GA 30332-0205, USA
E-mail: christos@isye.gatech.edu

Abstract

A fundamental goal of a discrete-event simulation experiment is the computation of point and confidence interval estimators for various performance measures, such as means and quantiles. Data generated during a single simulation run are sample paths from serially correlated stochastic processes. This chapter presents various forms of point estimators for means and quantiles, and shows how the user can account for the dependencies in within-run data towards the computation of valid confidence intervals. Finally, this chapter reviews methods for estimating a density function using a random sample.

1 Introduction

The ultimate goal of discrete-event simulations is the computation of point estimators and confidence intervals (CIs) for a variety of system performance measures. Towards this goal simulations use the generated data during event times to compute finite-sample time averages or sample averages as approximations to unknown means. There are two types of simulations, *finite-horizon* ones and *steady-state* ones. The first type is concerned with performance measures evaluated over a specific time window, such as the mean cost incurred by an inventory policy during a month. On the other hand, a steady-state simulation studies the long-term behavior of the system of interest. A performance measure of a system is called a steady-state parameter if it is a characteristic of the equilibrium distribution of an output process. An example is the simulation of a continuously operating communication system where the objective is the computation of the mean delay of a data packet. Regardless of their type, simulations produce within-run data that are correlated, making the application of "standard" statistical techniques nontrivial.

In finite-horizon simulations systems start at a pre-determined state, and the primary issue is the number of independent replications (or runs) that need to be made to compute estimators with a pre-specified precision. This issue can be addressed by standard statistical techniques, as we will illustrate in Section 5. On the other hand, steady-state simulations often require substantial statistical expertise. The most notable issues related to steady-state simulations are:

- Choosing the initial state for the system; the effect of this choice lessens as the run length increases.
- Choosing a *warm-up* interval that elapses before data collection can begin. The objective is to reduce the impact of the initial conditions on the finite-sample averages.
- Choosing a run length (time interval length or number of observations to be collected) to yield estimators that meet certain precision criteria.
- Choosing between a long run and several short runs – not a simple issue by any means.

Virtually all commercial software packages contain efficient routines for the recursive computation of finite-sample averages. For instance, they compute estimates for mean queue lengths, mean entity delays in queues, and server utilizations. However, they typically provide limited guidance for performing "sound" statistical analysis. The main reason is the difficulty in automating these analyses and the (often justifiable) desire of analysts to look at the final results without "wasting time" with the intricacies of the underlying statistical methodologies. This collective attitude ends up misguiding analysts and managers with regard to the usefulness of the simulation's estimates. The aforementioned issues are discussed in a variety of texts, e.g., Fishman (2001) and Law and Kelton (2000). This chapter formalizes them within a single theme and serves as a guide for several forthcoming chapters that address them in more detail.

In addition to point and interval estimates, users are often interested in estimating the density functions (or cumulative distribution functions) of random variables generated by a computer simulation. Several simulation packages can generate histograms, but such plots are often "poor" estimates of the unknown density function because their shape depends heavily on the chosen origin and the bin width. Although the statistical literature contains many state-of-the-art density estimation techniques, such as those based on kernel functions, the simulation literature (in particular texts) barely mentions such techniques, and only within the context of independent input data. This chapter aims to close the gap between the statistical and simulation literatures by reviewing univariate kernel density estimators based on independent samples and sample paths of stationary dependent processes.

Section 2 lists notation and reviews basic concepts from probability theory and statistics. Section 3 uses a simple queueing system to illustrate the properties of estimators based on sample averages and time averages. Section 4 reviews issues related to dependencies in stochastic processes and sets the stage for the methods described in Chapter 15. Section 5 applies concepts from

Section 2 to single measurements from independent replications. Section 6 discusses modern kernel density estimators.

2 Background

This section reviews tools needed to establish asymptotic (as the sample size increases) properties of estimators and to obtain confidence intervals. Let $X, X_1, X_2, \ldots$ be random variables (RVs) from a common probability space $(\Omega, \mathcal{F}, P)$, where Ω is the sample space, $\mathcal{F}$ is a σ-field of events and P is the probability measure. Recall that an RV on $(\Omega, \mathcal{F}, P)$ is a real-valued function $X = X(\omega)$ such that for each $x \in \mathbb{R}$, the set $\{\omega\colon X(\omega) \leqslant x\}$ is an event. The cumulative distribution function (CDF) of X is $F_X(x) = P(X \leqslant x)$, $x \in \mathbb{R}$. L^p is the space of RVs X with $\mathrm{E}(|X|^p) < \infty$.

We say that

$$X_n \to X \begin{cases} \text{almost surely (w.p.1)} & \text{if } \Pr\bigl(X_n(\omega) \to X(\omega) \\ & \qquad \text{as } n \to \infty\bigr) = 1, \\ \text{in probability} & \text{if } \lim_{n\to\infty} \Pr\bigl(|X_n - X| \leqslant \varepsilon\bigr) = 1 \\ & \quad \forall \varepsilon > 0, \\ \text{in distribution} & \text{if } \lim_{n\to\infty} F_{X_n}(x) = F_X(x) \\ & \quad \text{at all continuity points } x \text{ of } F_X, \\ \text{in } L^p,\ p \geqslant 1, & \text{if } \lim_{n\to\infty} \mathrm{E}\bigl[|X_n - X|^p\bigr] = 0. \end{cases}$$

The respective notations for these modes are $\xrightarrow{\text{a.s.}}$, $\xrightarrow{\text{p}}$, $\xrightarrow{\text{d}}$ and $\xrightarrow{L^p}$. Convergence in L^1 (L^2) is also called convergence in mean (quadratic mean). Among the first three modes, almost sure convergence is the strongest while convergence in distribution is the weakest (and easiest to establish). For additional details and related results, see Chapter 5 of Karr (1993).

Now suppose that the RVs $X_1, X_2, \ldots$ are from some distribution with an unknown parameter θ and the objective is to estimate a quantity δ that is a function of θ, say $\delta := g(\theta)$. For fixed n, let $\delta_n = \delta_n(X_1, \ldots, X_n)$ be an estimator of δ. The bias of δ_n is defined as $\mathrm{Bias}(\delta_n) := \mathrm{E}(\delta_n) - \delta$, and the mean squared error of δ_n is $\mathrm{MSE}(\delta_n) := \mathrm{E}[(\delta_n - \delta)^2] = \mathrm{Bias}^2(\delta_n) + \mathrm{Var}(\delta_n)$. If $\mathrm{E}(\delta_n) = \delta$, then δ_n is said to be *unbiased*. Furthermore, δ_n is said to be a *consistent* (respectively, *strongly consistent*) estimator of δ if $\delta_n \xrightarrow{\text{p}} \delta$ (respectively, $\delta_n \xrightarrow{\text{a.s.}} \delta$). If δ_n is unbiased for each n and $\mathrm{Var}(\delta_n) \to 0$ as $n \to \infty$, then δ_n is also consistent. This is a direct implication of Chebyshev's inequality (Karr, 1993, p. 122): $\Pr(|\delta_n - \delta| \geqslant \varepsilon) \leqslant \mathrm{E}[(\delta_n - \delta)^2]/\varepsilon^2 = \mathrm{Var}(\delta_n)/\varepsilon^2$.

The remainder of this section illustrates the aforementioned concepts with a few classical results. Suppose that $X_1, X_2, \ldots$ are independent and identically distributed (IID) RVs with finite mean μ. The sample mean $\overline{X}_n := \frac{1}{n}\sum_{i=1}^{n} X_i$ is an unbiased estimator of μ because $\mathrm{E}(\overline{X}_n) = \mu$. $\overline{X}_n$ is also a strongly consistent estimator of μ by the *strong law of large numbers* (Karr, 1993, pp. 188–189),

$$\overline{X}_n \xrightarrow{\text{a.s.}} \mu \quad \text{as } n \to \infty.$$

If $\sigma_X^2 := \operatorname{Var}(X_1) \in (0, \infty)$, the *central limit theorem* (CLT) (see Karr, 1993, p. 174) states that

$$\frac{\overline{X}_n - \mu}{\sigma_X/\sqrt{n}} \xrightarrow{\text{d}} \text{N}(0, 1) \quad \text{as } n \to \infty,$$

where N(0, 1) is a standard normal RV. The CLT remains valid when the typically unknown variance σ_X^2 is replaced by its unbiased and consistent estimator $S_n^2(X) := \frac{1}{n-1}\sum_{i=1}^n (X_i - \overline{X}_n)^2$, the sample variance of the X_i. Therefore, for sufficiently large n,

$$p_{n,1-\alpha} := \Pr\left(\frac{|\overline{X}_n - \mu|}{S_n(X)/\sqrt{n}} \leqslant z_{1-\alpha/2}\right) \approx 1 - \alpha, \tag{1}$$

where z_β denotes the β-quantile of the standard normal distribution. Solving the inequality in the middle of (1) for μ, one has the well-known approximate (two-sided) $1 - \alpha$ CI for μ,

$$\overline{X}_n \pm z_{1-\alpha/2}\frac{S_n(X)}{\sqrt{n}}. \tag{2}$$

We call $p_{n,1-\alpha}$ the "coverage probability" of the CI (2). One interprets this CI as follows: Suppose that a large number of independent trials are performed; in each trial, n observations are collected and a CI for μ is computed using (2). As the number of trials grows, the proportion of CIs that contain μ approaches $1 - \alpha$.

The number of observations n required for $p_{n,1-\alpha} \approx 1 - \alpha$ depends on the symmetry of the distribution of X_i. The more skewed (asymmetric) the density/probability function of X_i, the larger n required. To reduce potential undercoverage problems ($p_{n,1-\alpha} < 1 - \alpha$) for small n, one may replace the normal quantile $z_{1-\alpha/2}$ by the larger quantile $t_{n-1,1-\alpha/2}$ of Student's t_{n-1} distribution with $n - 1$ degrees of freedom. This choice for degrees of freedom is due to the fact that for IID normally distributed X_i,

$$\frac{\overline{X}_n - \mu}{S_n(X)/\sqrt{n}} \sim t_{n-1},$$

where the notation $X \sim Y$ is used to indicate that the RVs X and Y have the same distribution.

3 Sample averages and time averages

Simulation output data are realizations (or *sample paths*) of *stochastic processes*. A stochastic process is a probabilistic model of a system that evolves randomly. Let $X := \{X_t, t \in T\}$ be a stochastic process with state space S and time set T. Each X_t is a random element in S: a measurable mapping from an underlying probability space $(\Omega, \mathcal{F}, P)$ to the measurable space $(S, \mathcal{E})$ (all functions herein will be measurable). Typically, the state space S is a countable set, the real line $\mathbb{R}$, or a subset of a Euclidean space or a metric space; and $T = \mathbb{R}$ or $\mathbb{R}_+$ for continuous-time processes and $T = \mathbb{Z}$ or $\mathbb{Z}_+$ for discrete-time processes. We use the following simple system to illustrate the main concepts associated with data collection and analysis.

The notation $GI/G/c$ denotes a queueing system with c parallel servers and a single queue with unlimited buffer. The times between successive job arrivals are IID RVs, say $\{A_i, i \geqslant 1\}$ with finite mean $\mathrm{E}(A) := 1/\lambda$ and variance. The service times are also IID RVs, say $\{S_i, i \geqslant 1\}$ with finite mean $\mathrm{E}(S) := 1/\omega$ ($\lambda < c\omega$ for stability) and variance, and are independent of the arrival process. Let X_t denote the number of units in the system at time t, let B_t denote the number of busy servers at time t, and let W_i denote the time the ith job spends in the system. The system is formally described by the stochastic process $Y_t = (X_t, R_t^1, \ldots, R_t^c)$, where R_t^j is the remaining service time for the unit served at server j at time t. Suppose that the rule for assigning customers to servers is such that the process $\{Y_t, t \geqslant 0\}$ is *regenerative* over a sequence of epochs $\{\tau_n\}$ (see Chapter 16). A typical rule is to send an arrival to an arbitrary idle server or to the lowest numbered idle server. If the system empties out with a reasonable frequency, it is natural to let τ_n be the nth time an arrival finds the system empty. Below we consider two data collection mechanisms often encountered in practice.

Scheme I. Suppose data collection starts at some time $s \geqslant 0$ and ends at time $s + \tau$ ($\tau > 0$). This approach is convenient for continuous-time sample paths. The *time-average*

$$\overline{X}(s, s+\tau) = \frac{1}{\tau}\int_s^{s+\tau} X_t \,\mathrm{d}t \tag{3}$$

estimates the mean number of units in the system during the time interval $[s, s+\tau]$, while the time-average

$$\overline{B}(s, s+\tau) = \frac{1}{\tau}\int_s^{s+\tau} B_t \,\mathrm{d}t \tag{4}$$

estimates the (mean) number of busy servers during $[s, s+\tau]$.

The processes $\{X_t\}$ and $\{B_t\}$ have limiting distributions with finite means, denoted by L and $\nu := \lambda/\omega$ (see Serfozo, 1999). We study the properties of the estimators (3) and (4) by first fixing τ. One can show that as $s \to \infty$,

$$\overline{X}(s, s+\tau) - L \xrightarrow{\mathrm{d}} \xi(\tau) \quad \text{and} \quad \overline{B}(s, s+\tau) - \nu \xrightarrow{\mathrm{d}} \zeta(\tau),$$

where $\xi(\tau)$ and $\zeta(\tau)$ are RVs with zero mean and distribution that is independent of the system state at time s. Furthermore, as $\tau \to \infty$, $\xi(\tau)$ and $\zeta(\tau)$

converge in distribution (and hence in probability) to zero so that

$$\overline{X}(s, s+\tau) \xrightarrow{\mathrm{d}} L \quad \text{and} \quad \overline{B}(s, s+\tau) \xrightarrow{\mathrm{d}} \nu \quad \text{as } s, \tau \to \infty. \tag{5}$$

Now, under some additional mild assumptions (e.g., $\mathrm{E}(A^4) < \infty$ and $\mathrm{E}(S^4) < \infty$), one can show that for fixed s there exist finite constants $\sigma^2(X) > 0$ and $\sigma^2(B) > 0$ so that, as $\tau \to \infty$,

$$\sqrt{\tau}\big[\overline{X}(s, s+\tau) - L\big] \xrightarrow{\mathrm{d}} \mathrm{N}\big(0, \sigma^2(X)\big) \quad \text{and}$$
$$\sqrt{\tau}\big[\overline{B}(s, s+\tau) - \nu\big] \xrightarrow{\mathrm{d}} \mathrm{N}\big(0, \sigma^2(B)\big).$$

The last two properties allow the user to compute CIs for the respective steady-state means L and ν.

The estimation of measures other than means is also possible. For instance, the time average

$$\frac{1}{\tau}\int_s^{s+\tau} \mathbf{1}\{B_t \geqslant r\}\,\mathrm{d}t$$

is an estimator for the probability that at least r servers are busy during the time window $[s, s+\tau]$.

Scheme II. Data collection starts when unit m departs and ends when unit $m+n$ ($n \geqslant 1$) departs. Denote the respective departure times by T_m and T_{m+n}. Then the mean time a job spends in the system during the interval $[T_m, T_{m+n}]$ can be estimated by the sample average

$$\overline{W}(T_m, T_{m+n}) = \frac{1}{n}\sum_{j=m+1}^{m+n} W_j. \tag{6}$$

This estimator has properties analogous to the properties of the estimator $\overline{X}(s, s+\tau)$. Specifically, as m is held constant and $n \to \infty$, $\overline{W}(T_m, T_{m+n}) \xrightarrow{\text{a.s.}} W$, where $W = L/\lambda$ by Little's law; see Stidham (1974) and Serfozo (1999, Chapter 5). As with Scheme I, usually there exists a finite constant $\sigma^2(W) > 0$ so that, as $n \to \infty$,

$$\sqrt{n}\big[\overline{W}(T_m, T_{m+n}) - W\big] \xrightarrow{\mathrm{d}} \mathrm{N}\big(0, \sigma^2(W)\big).$$

Remark 1. Sample averages can be collected under Scheme I, but have random denominators. For instance, let N_t be the numbers of departures by time t. The sample average

$$\overline{W}(s, s+\tau) = \frac{1}{N_{s+\tau} - N_s}\sum_{j=N_s+1}^{N_{s+\tau}} W_j \tag{7}$$

estimates the mean time a job that completed service in $[s, s+\tau]$ spent in the system. Since the denominator of Equation (7) is an RV, the derivation of the limiting properties of the estimator $\overline{W}(s, s+\tau)$ requires extra care, but they are similar to the limiting properties of its counterpart in Equation (6).

Finally, parallel properties hold for the estimator of the mean number of jobs in the system

$$\overline{X}(T_m, T_{m+n}) = \frac{1}{T_{m+n} - T_m} \int_{T_m}^{T_{m+n}} X_t \, \mathrm{d}t$$

computed under Scheme II.

4 Stationary processes

The scope of many simulation studies is the estimation of limiting measures associated with stochastic processes. This section starts with some common structural assumptions that allow us to study the impact of the dependence within a stochastic process and proceeds with limit theorems. This section contains a synopsis of material from Alexopoulos et al. (2006) and Durrett (2005).

One way to describe a stochastic process is to specify the joint distribution of $X_{t_1}, X_{t_2}, \ldots, X_{t_n}$ for each set of times $t_1 < t_2 < \cdots < t_n$ and each n. This approach is typically too complicated to be attempted in practice. A simpler approach relies on the specification of the first and second moment functions of the process. These functions are the *mean* function $\mu_t := \mathrm{E}[X_t]$ and the *autocovariance* function

$$C_{t_1,t_2} := \mathrm{Cov}[X_{t_1}, X_{t_2}], \quad t_1 \leqslant t_2.$$

Notice that $C_{t_1,t_2} = C_{t_2,t_1}$. The *variance* function is then given by $\sigma_t^2 := \mathrm{Var}(X_t) = C_{t,t}$ and the *autocorrelation* function is defined as

$$\rho_{t_1,t_2} := \frac{C_{t_1,t_2}}{\sigma_{t_1}\sigma_{t_2}}, \quad t_1 \leqslant t_2.$$

(Strict) Stationarity. The stochastic process X is *stationary* if, for any $t_1, \ldots, t_k$, $t \in T$,

$$(X_{t_1+t}, \ldots, X_{t_k+t}) \overset{\mathrm{d}}{=} (X_{t_1}, \ldots, X_{t_k}),$$

where "$\overset{\mathrm{d}}{=}$" denotes equality in distribution. This says that the distribution of the process with the time origin shifted to t is equal to that of X. In other words, $\theta^t X \overset{\mathrm{d}}{=} X$ for any $t \in T$, where $\{\theta^t, t \in T\}$ is the family of time-shift transformations defined by $\theta^t x = (x_{s+t}, s \in T)$. An immediate result is that the joint distribution of $X_{t_1}, X_{t_2}, \ldots, X_{t_k}$ depends only on the distances between $t_1, t_2, \ldots, t_k$.

Weak stationarity. The process X is said to be *weakly stationary* if its mean and variance functions are constant and its autocovariance function satisfies

$$\mathrm{Cov}[X_s, X_{s+t}] = C_t, \quad s \geqslant 0, t \geqslant 0,$$

that is, it depends only on the *lag* t. If the process X is Gaussian whence all finite-dimensional distributions are multivariate normal, then weak stationarity implies stationarity.

Ergodicity. A stationary process X is *ergodic* if $P\{X \in A\} = 0$ or 1, for each $A \in \mathcal{E}$ that is *time-shift invariant*: $\{X \in A\} = \{\theta^t X \in A\}$, $t \in T$. This property is important for steady-state simulations because it allows the estimation of limiting performance measures based on a single realization (sample path) of the process X. Examples are stationary Markov chains and stationary functionals of a stationary, ergodic process.

Example 1 (The $M/M/1$ queueing system). This is a special case of the $GI/G/c$ system with one server and exponentially distributed interarrival and service times. The ratio $\nu = \lambda/\omega$ is the traffic intensity or (long-run) server utilization. Suppose that the service discipline is first-come, first-served. Let D_i be the delay time in queue of the ith customer and assume that the system starts empty. Since $D_1 = 0$, we have $\mathrm{E}(D_1) = 0$. However, the first ($i = 1$) of the following recursive equations (Lindley, 1952)

$$D_{i+1} = \max\{D_i + S_i - A_{i+1}, 0\}, \quad i \geqslant 1, \tag{8}$$

implies $\Pr(D_2 > 0) = \Pr(S_1 > A_2) = \lambda/(\lambda + \omega) > 0$; hence $\mathrm{E}(D_2) > 0$. Therefore the delay process $\{D_i, i \geqslant 1\}$ is not stationary. Using queueing theory (Gross and Harris, 1998, Chapter 2), one has

$$\lim_{i\to\infty} \Pr(D_i \leqslant x) = 1 - \nu + \nu\bigl(1 - \mathrm{e}^{-(\omega-\lambda)x}\bigr), \quad x \geqslant 0, \tag{9}$$

$$\mu = \lim_{i\to\infty} \mathrm{E}(D_i) = \frac{\nu}{(1-\nu)\omega} \quad \text{and} \quad \lim_{i\to\infty} \mathrm{Var}(D_i) = \frac{\nu(2-\nu)}{\omega^2(1-\nu)^2}.$$

Equation (9) suggests that the delay process becomes asymptotically stationary. Indeed, if D_1 has the distribution on the right-hand side of (9), Equation (8) implies (after some work) that all D_i have the same distribution and the delay process is stationary.

The autocorrelation function of the delay process $\{D_i\}$ is given by (Blomqvist, 1967)

$$\rho_j = \frac{(1-\nu)^3(1+\nu)}{(2-\nu)\nu^3} \times \sum_{k=j+3}^{\infty} \left[\frac{\nu}{(\nu+2)^2}\right]^k \frac{(2k-3)!}{k!(k-2)!}(k-j-1)(k-j-2)$$

for $j = 0, 1, \ldots$. This function is monotone decreasing with a very long tail that increases as the server utilization ν increases (for instance, $\rho_{200} \approx 0.30$ when $\nu = 0.9$). This makes the $M/M/1$ system a good test bed for evaluating simulation methodologies.

Example 2 (Moving average process). A well-studied stationary sequence is the moving average process of order q (often abbreviated to MA(q))

$$X_i = \beta_0 Z_i + \beta_1 Z_{i-1} + \cdots + \beta_q Z_{i-q}, \quad i \geqslant 0,$$

where the coefficients β_i are constants and $\{Z_i, i \in \mathbb{Z}\}$ is an IID random sequence with mean zero and finite variance σ_Z^2. MA processes have applications in several areas, particularly econometrics (Chatfield, 1989).

Clearly,

$$\mathrm{E}(X_i) = 0, \qquad \mathrm{Var}(X_i) = \sigma_Z^2 \sum_{i=0}^{q} \beta_i^2$$

while some algebra yields the autocovariace function

$$C_j = \begin{cases} \sigma_Z^2 \sum_{i=0}^{q-j} \beta_i \beta_{i+j} & \text{for } j = 0, 1, \ldots, q, \\ 0 & \text{for } j > q, \end{cases}$$

which “cuts off” at lag q.

4.1 *Impact of dependence*

Although this topic will be the subject of Chapter 15, we will give a brief introduction of the main issues. Suppose one observes the portion $X_1, \ldots, X_n$ of a discrete-time stationary process for the purpose of estimating the mean $\mu := \mathrm{E}(X_1)$. Clearly, $\overline{X}_n$ is an unbiased estimator of μ while some algebra yields

$$\mathrm{Var}(\overline{X}_n) = \frac{\sigma_X^2}{n}\left[1 + 2\sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)\rho_j\right] := \frac{\sigma_X^2}{n}(1 + \gamma_n). \tag{10}$$

In order for $\overline{X}_n$ to be a consistent estimator of μ, we require that $\lim_{n\to\infty} \mathrm{Var}(\overline{X}_n) = 0$. The last condition holds if $\lim_{n\to\infty} n\,\mathrm{Var}(\overline{X}_n) < \infty$ or, equivalently,

$$\lim_{n\to\infty} \gamma_n < \infty. \tag{11}$$

The condition $\lim_{j\to\infty} C_j = 0$ is necessary for (11) but not sufficient. A necessary and sufficient condition is

$$\sum_{j=-\infty}^{\infty} |C_j| < \infty. \tag{12}$$

In simple terms, the covariance between X_i and X_{i+j} must dissipate sufficiently fast for the summation in (12) to remain bounded. If condition (12) holds, we call

$$\sigma^2 := \sum_{j=-\infty}^{\infty} C_j \tag{13}$$

the *variance parameter* of the process X.

Under the assumption that $\overline{X}_n$ is approximately normally distributed (which is reasonable for sufficiently large n), the construction of a CI for μ requires the derivation of an estimator for $\mathrm{Var}(\overline{X}_n)$. Based on the practice for IID data, one might be inclined to estimate $\mathrm{Var}(\overline{X}_n)$ by $S_n^2(X)/n$. Is this wise? After some algebra one can establish the following formula:

$$\begin{aligned}\mathrm{E}\left[\frac{S_n^2(X)}{n}\right] &= \frac{\sigma_X^2}{n}\left[1 - \frac{2}{n-1}\sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)\rho_j\right] \\ &= \frac{n/(1+\gamma_n) - 1}{n-1}\,\mathrm{Var}(\overline{X}_n).\end{aligned} \tag{14}$$

For processes that are positively correlated (i.e., $\rho_i > 0\ \forall i$), Equation (14) implies that $\mathrm{E}[S_n^2(X)/n] \ll \mathrm{Var}(\overline{X}_n)$. Hence the CI in (2) can exhibit severe undercoverage.

4.2 Ergodic and central limit theorems for stationary sequences

The strong law of large numbers and the CLT for sums of IID random variables have analogues for partial sums of stationary sequences. Our discussion of these will involve the notion of ergodicity. The following ergodic theorem is in Durrett (2005).

Theorem 1 (Ergodic theorem: von Neumann, Birkhoff). *Suppose that* $\{X_i\}$ *is a stationary, ergodic sequence of random variables with* $\mathrm{E}(|X_1|) < \infty$. *Then*

$$\overline{X}_n \xrightarrow{a.s.} \mu \quad \textit{as } n \to \infty. \tag{15}$$

Moreover, if $\mathrm{E}(|X_1|^p) < \infty$ *for* $p \geqslant 1$, *then*

$$\overline{X}_n \xrightarrow{L^p} \mu \quad \textit{as } n \to \infty. \tag{16}$$

These w.p.1 and L^p limit properties also hold for stationary processes that are not ergodic; the $\mathrm{E}(X_1)$ is just replaced by the conditional expectation of X_1 with respect to the σ-field of invariant events. The ergodic theorem also holds for averages of continuous-time processes, in which case

$$\frac{1}{t}\int_0^t X_s\,\mathrm{d}s \xrightarrow{\text{a.s.}} \mathrm{E}(X_0) \quad \text{as } t \to \infty.$$

There are several functional central limit theorems for stationary processes under various assumptions. Two prominent ones are as follows (Durrett, 2005). Suppose that $\{X_i, i \in \mathbb{Z}\}$ is a stationary, ergodic sequence of random variables with mean μ and finite variance. The assumption that the process has an infinite past is natural and is justified since a one-sided process with time set $\mathbb{Z}_+$ can be extended by (by Kolmogorov's extension theorem) to a two-sided process.

Define the partial sums $S_n = \sum_{i=1}^{n}(X_i - \mu)$, and consider the related stochastic process

$$n^{-1/2}S_{(nt)} = n^{-1/2}\sum_{i=1}^{\lfloor nt \rfloor}(X_i - \mu), \quad t \in \mathbb{R}_+.$$

Here $\{n^{-1/2}S_{(n\cdot)}, n \geqslant 1\}$ is a sequence of random elements in the Skorohod space $D(\mathbb{R}_+)$ of real-valued functions on $\mathbb{R}_+$ that are right-continuous with left-hand limits. We use "$\Longrightarrow$" to denote weak convergence of random elements in this space.

Theorem 2. *In addition to the assumptions above, suppose that*

$$\sum_{n=1}^{\infty}\left[\mathrm{E}(\xi_n^2)\right]^{1/2} < \infty, \tag{17}$$

where $\xi_n = \mathrm{E}[X_0 - \mu | X_k - \mu\colon\ k \leqslant -n]$. *Then the series* (13) *converges absolutely and*

$$n^{1/2}S_{(n\cdot)} \Longrightarrow \sigma\mathcal{W},$$

where $\{\mathcal{W}(t), t \geqslant 0\}$ *denotes a standard Brownian motion process.*

The following is another functional central limit theorem with a stronger assumption. This uses the notion that the sequence X is *ϕ-mixing* if there are $\varphi_j \downarrow 0$ such that, for each $j \geqslant 1$,

$$\left|\Pr(A \cap B) - \Pr(A)\Pr(B)\right| \leqslant \phi_j \Pr(A), \quad A \in \mathcal{F}_{-\infty}^{i}, B \in \mathcal{F}_{i+j}^{\infty},$$

where $\mathcal{F}_i^j$, $i \leqslant j$, denotes the σ-field generated by $X_i, X_{i+1}, \ldots, X_j$.

Theorem 3. *The assertion of Theorem 2 is true if assumption (17) is replaced by the assumption that X is ϕ-mixing and $\sum_{n=1}^{\infty}\phi_n^{1/2} < \infty$.*

Assuming $\sigma > 0$, Theorems 2 and 3 imply

$$\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \xrightarrow{\mathrm{d}} \mathrm{N}(0, 1) \quad \text{as } n \to \infty. \tag{18}$$

Equation (18) can be used to obtain an asymptotically (as $n \to \infty$) CI for μ. The estimation of σ^2 is the main topic of Chapter 15.

Before we proceed, we present an even stronger property to be revisited in Section 6.7. The process X is *strongly* α-mixing if there are $\alpha_j \downarrow 0$ such that, for each $j \geqslant 1$,

$$\left|\Pr(A \cap B) - \Pr(A)\Pr(B)\right| \leqslant \alpha_j, \quad A \in \mathcal{F}^i_{-\infty}, B \in \mathcal{F}^\infty_{i+j}.$$

Since $\alpha_j \leqslant \phi_j$, a ϕ-mixing process is also strongly mixing. Informally, mixing means that events in the distant future are approximately independent of those in the past. For details see Billingsley (1968) and Rosenblatt (1956).

Remark 2. Contrary to popular belief, many stochastic processes encountered in simulation output analysis are *not* ϕ-mixing. Examples are autoregressive processes, regenerative processes (see Chapter 16) with regenerations not occurring uniformly fast over the state space, and virtually all open queueing networks. However, positive recurrent regenerative processes are strongly mixing. For details, see Glynn and Iglehart (1985).

Example 3 (Gaussian first-order autoregressive process). Another well-known stationary process is the autoregressive process of order one, denoted by AR(1), and often called the Markov process in the time series literature,

$$X_i = \mu + \rho(X_{i-1} - \mu) + Z_i, \quad i \geqslant 1,$$

where $|\rho| < 1$, $X_0 \sim \mathrm{N}(\mu, 1)$, and the Z_i are IID $\mathrm{N}(0, 1-\rho^2)$.

The autocorrelation function of this process $\rho_j = \rho^j$, $j \geqslant 0$, is monotone decreasing if $\rho > 0$ with a tail that becomes longer as ρ increases, and exhibits damped harmonic behavior around the zero axis if $\rho < 0$. Applying Equation (10) one has

$$n \operatorname{Var}(\overline{X}_n) = 1 + 2\sum_{j=1}^{n-1}\left(1 - \frac{j}{n}\right)\rho^j \longrightarrow \frac{1+\rho}{1-\rho} \quad \text{as } n \to \infty.$$

Hence $\overline{X}_n$ is a consistent estimator of the mean $\mu = \mathrm{E}(X_i)$ and $\sigma^2 = (1+\rho)/(1-\rho)$. This process does not possess the ϕ-mixing property, but the CLT (18) holds by Theorem 2.

5 Analyzing data from independent replications

Suppose that a simulation run yields n output observations, say $X_1, \ldots, X_n$. Let $Y = f(X_1, \ldots, X_n)$ be a measurable function, and assume that the objective of the study is the estimation of $\mu := \mathrm{E}(Y)$. For instance, if X_i is the time spent in system by customer i, then $f(X_1, \ldots, X_n)$ could be the average time in system experienced by the first n customers or the maximum time in system. By definition, Y is an unbiased estimator for μ.

To bypass the impact of the autocorrelations, one can run k independent replications of the system simulation. Each replication starts in the same state and uses a portion of the random number stream that is different from the portions used to run the other replications. Assume that replication i produces the output data $X_{i,1}, X_{i,2}, \ldots, X_{i,n}$ and $Y_i = f(X_{i,1}, X_{i,2}, \ldots, X_{i,n})$. Since the RVs Y_i are IID, their sample mean $\overline{Y}_k$ is also an unbiased estimator of μ, and their sample variance $S_k^2(Y)$ is an unbiased estimator of $\mathrm{Var}(Y)$. If, in addition, k is sufficiently large, an approximate $1-\alpha$ CI for μ is

$$\overline{Y}_k \pm t_{k-1,1-\alpha/2}\frac{S_k(Y)}{\sqrt{k}}. \tag{19}$$

Denote the half-width of the interval (19) by $\delta(k,\alpha) = t_{k-1,1-\alpha/2}S_k(Y)/\sqrt{k}$.

5.1 Sequential estimation

Suppose that one wishes to estimate μ within a tolerance $\pm d$, where d is user-specified. More formally, one would like to make k runs so that

$$\Pr(\overline{Y}_k - d \leqslant \mu \leqslant \overline{Y}_k + d) \geqslant 1-\alpha, \tag{20}$$

where $\alpha \in (0,1)$. The sequential procedure of Chow and Robbins (1965) (see also Nadas, 1969) is to run one replication at a time and stop at run k^* such that

$$k^* = \min\left[k\colon\ k \geqslant 2,\ \delta(k,\alpha) \leqslant \sqrt{\frac{k}{k-1}d^2 - \frac{t^2_{k-1,\alpha/2}}{k(k-1)}}\right]. \tag{21}$$

The stopping rule (21) is based on the limiting result

$$\lim_{d\to 0}\Pr(\overline{Y}_{k^*} - d \leqslant \mu \leqslant \overline{Y}_{k^*} + d) = 1-\alpha. \tag{22}$$

Equation (22) indicates that as the tolerance d decreases, the probability that the interval $\overline{Y}_{k^*} \pm d$ contains μ converges to $1-\alpha$. Notice that as k increases, the right-hand side of the last inequality in (21) approaches d.

Now suppose that $Y_1, Y_2, \ldots$ are normally distributed. Starr (1966) showed that the choice

$$k^* = \min\bigl[k\colon\ k \geqslant 3,\ k \text{ odd},\ \delta(k,\alpha) \leqslant d\bigr]$$

yields

$$\Pr(\overline{Y}_{k^*} - d \leqslant \mu \leqslant \overline{Y}_{k^*} + d) \geqslant \begin{cases} 0.928 & \text{if } \alpha = 0.05, \\ 0.985 & \text{if } \alpha = 0.01. \end{cases}$$

The last inequalities indicate little loss in the confidence level for arbitrary d. Based on Starr's result and Equation (22), we recommend the simpler and more intuitive stopping rule that starts with at least 5 runs,

$$k^* = \min\bigl[k\colon\ k \geqslant 5, \delta(k,\alpha) \leqslant d\bigr].$$

Often, the t quantile is replaced by the normal quantile $z_{1-\alpha/2}$.

An alternative *two-stage* approach for computing a CI for μ with half-width at most d works as follows: The first stage uses k_0 replications to compute a variance estimate $S^2_{k_0}(Y)$ and a CI with half-width $\delta(k_0,\alpha)$. Assume that the estimate $S^2_{k_0}(Y)$ does not change significantly as k_0 increases. If $\delta(k_0,\alpha) \leqslant d$, the procedure terminates. Otherwise, an estimate of the total number of replications required to obtain a half-width of at most d is computed from

$$\begin{aligned}\hat{k} &= \min\left[k\colon\ k \geqslant k_0,\ \frac{t_{k_0-1,1-\alpha/2}S_{k_0}(Y)}{\sqrt{k}} \leqslant d\right] \\ &= \max\left\{k_0, \left\lceil \frac{t^2_{k_0-1,1-\alpha/2}S_{k_0}(Y)^2}{d^2} \right\rceil\right\},\end{aligned}$$

where $\lceil\cdot\rceil$ is the ceiling function. The efficacy of this method depends on the closeness of $S^2_{k_0}(Y)$ to the unknown $\operatorname{Var}(Y)$. If $S^2_{k_0}(Y)$ underestimates $\operatorname{Var}(Y)$, then $\hat{k}$ will be smaller than actually needed. Conversely, if $S^2_{k_0}(Y)$ overestimates $\operatorname{Var}(Y)$, then unnecessary replications will have to be made.

Alexopoulos and Seila (1998) studied the performance of the sequential and two-stage methods by a set of experiments concerning the throughput of an $M/M/1$ system. Based on their experiments, the sequential procedure with an initial sample of at least 5 replications appears to outperform the two-stage procedure as: (a) the resulting CI half-width is always less than or equal to the target value; (b) the variation in the final sample sizes and CI half-widths is substantially smaller; and (c) the CIs had estimated coverage close to the nominal probability.

An alternative problem is the computation of an estimate for μ with relative error $|\overline{Y}_k - \mu|/|\mu| \leqslant c$, where c is a positive constant. Formally, one requests

$$\Pr\left(\frac{|\overline{Y}_k - \mu|}{|\mu|} \leqslant c\right) \geqslant 1-\alpha.$$

Let $c' = c/(1+c)$ and assume that we perform one replication at a time until the relative half-width $\delta(k,\alpha)/|\overline{Y}_k| \leqslant c'$. Since $\delta(k,\alpha)/|\overline{Y}_k|$ estimates the actual relative error, we have

$$\begin{aligned}1-\alpha &\approx \Pr\left(\frac{|\overline{Y}_k - \mu|}{|\overline{Y}_k|} \leqslant \frac{\delta(k,\alpha)}{|\overline{Y}_k|}\right) \\ &\leqslant \Pr\bigl(|\overline{Y}_k - \mu| \leqslant c'|\overline{Y}_k|\bigr) \\ &= \Pr\bigl(|\overline{Y}_k - \mu| \leqslant c'|\overline{Y}_k - \mu + \mu|\bigr) \\ &\leqslant \Pr\bigl(|\overline{Y}_k - \mu| \leqslant c'|\overline{Y}_k - \mu| + c'|\mu|\bigr)\end{aligned}$$

$$
\begin{aligned}
&= \Pr\bigl[(1-c')|\overline{Y}_k - \mu| \leqslant c'|\mu|\bigr] \\
&= \Pr\biggl(\frac{|\overline{Y}_k - \mu|}{|\mu|} \leqslant \frac{c'}{1-c'}\biggr) \\
&= \Pr\biggl(\frac{|\overline{Y}_k - \mu|}{|\mu|} \leqslant c\biggr).
\end{aligned}
$$

Based on these observations, one can use the following stopping rule:

$$
k^* = \min\biggl[k\colon\ k \geqslant k_0,\ \frac{\delta(k,\alpha)}{|\overline{Y}_k|} \leqslant c'\biggr]. \tag{23}
$$

Law et al. (1981) showed that when c is close to 0, the coverage of the CI $\overline{Y}_k \pm \delta(k,\alpha)$ can be arbitrarily close to $1-\alpha$. They recommend that rule (23) be used with $c \leqslant 0.15$ and $k_0 \geqslant 10$.

5.2 *Quantile estimation*

The method of replications can also be used to implement nonparametric methods for estimating performance measures other than means. For example, let Y be the total cost incurred by an inventory system during a month. Suppose we wish to estimate the p-quantile $y_p = F_Y^{-1}(p) := \inf\{y \in \mathbb{R}\colon F_Y(y) \geqslant p\}$, $0 < p < 1$, of Y. If the cumulative distribution function (CDF) $F_Y(\cdot)$ is continuous and strictly monotone, then y_p is the unique solution to the equation $F_Y(y) = p$.

Let $Y_1, \ldots, Y_k$ be a random sample from a continuous CDF F_Y obtained by performing k independent replications, and let $Y_{(1)} < Y_{(2)} < \cdots < Y_{(k)}$ be the order statistics corresponding to the Y_i's (without loss of generality we assume that the Y_i are distinct). Then a point estimator for y_p is

$$
\hat{y}_p = \begin{cases} Y_{(kp)}, & \text{if } kp \text{ is integer,} \\ Y_{(\lfloor kp+1 \rfloor)}, & \text{otherwise,} \end{cases}
$$

where $\lfloor\cdot\rfloor$ is the floor function.

To compute a $1-\alpha$ CI for y_p, one identifies indices $i < j$ such that $\Pr(Y_{(i)} < y_p < Y_{(j)}) \geqslant 1-\alpha$. Then $(Y_{(i)}, Y_{(j)})$ is the required interval. The event $Y_{(i)} < y_p < Y_{(j)}$ has the binomial probability

$$
\begin{aligned}
\Pr(Y_{(i)} < y_p < Y_{(j)}) &= \sum_{\ell=i}^{j-1} \binom{k}{\ell} p^\ell (1-p)^{k-\ell} \\
&\approx \Phi\biggl(\frac{j-1-kp}{\sqrt{kp(1-p)}}\biggr) - \Phi\biggl(\frac{i-1-kp}{\sqrt{kp(1-p)}}\biggr),
\end{aligned}
$$

where the normal approximation is recommended for $kp \geqslant 5$ (Hogg and Craig, 1995). Since several index pairs can satisfy the last inequality, one would choose a symmetric range of indices. In this case, the indices would be

$$i = \lfloor kp + 0.5 - z_{1-\alpha/2}\sqrt{kp(1-p)} \rfloor \quad \text{and}$$
$$j = \lceil kp - 0.5 + z_{1-\alpha/2}\sqrt{kp(1-p)} \rceil.$$

(The last expressions involve continuity corrections.) It should be noted that quantile estimation is usually more difficult than estimation of the mean because point estimates for quantiles are biased and substantially larger sample sizes are required to obtain reasonably tight CIs. These problems are much more severe for more extreme quantiles, i.e., for p close to 0 or 1. Exceptions are heavy-tailed distributions (Gross et al., 2002). An introduction to nonparametric interval estimation methods is given in Hogg and Craig (1995).

5.3 *Using independent replications to estimate steady-state measures*

The method of independent replications can also be used for estimating the steady-state mean μ of an output process. Unfortunately, most simulations start at an initial state that is either fixed (e.g., the empty and idle state in a queueing system) or is randomly chosen from a distribution other than the steady-state one. As a result, the process X goes through an initial phase before reaching steady state. How can one deal with the initial transient phase? The simulation folklore suggests that, to bypass trouble arising from this portion of a run, the experimenter should simply allow the simulation to "warm up" before retaining data. Then the question is, how much truncation (deletion) on every replication is enough to produce unbiased, low-variance point estimators and valid CIs?

Suppose that the output data form a discrete-time process. Several heuristic methods try to identify an appropriate index ℓ and then truncate the first ℓ observations. For instance, Welch (1983) gives a graphical truncation method that has met some success; see Alexopoulos and Seila (1998) or Law and Kelton (2000) for a description and illustrative examples. A host of other heuristic methods are somewhat naive and ill-suited for use in practical problems, and so will not be discussed here. References for various statistics-based tests are listed in Alexopoulos and Seila (1998) and Law and Kelton (2000); they too are not completely trustworthy in some situations, and so we will dispense with any additional details for now. For these and other reasons, Alexopoulos and Seila (1998) state that it is difficult and problematic to select a proper truncation index, especially for congested queueing systems with slowly decreasing autocorrelation functions.

Since the initial conditions induce a systematic error that has a more-pronounced impact when multiple independent replications start at the same state, it is of interest to see what happens if we indeed incorporate a truncation strategy into the CI for the steady-state mean μ. Assuming that one still uses k independent replications with n observations per replication (after truncation) to compute a CI for μ, the replicate means are $Y_i(\ell) = n^{-1}\sum_{j=\ell+1}^{\ell+n} X_{i,j}$,

$i = 1, 2, \ldots, k$, the point estimator for μ is

$$\hat{\mu}_{\mathrm{R}} = \overline{Y}_k(\ell) = \frac{1}{k} \sum_{i=1}^{k} Y_i(\ell),$$

and the estimator for the variance parameter σ^2 is

$$\widehat{\sigma^2_{\mathrm{R}}} = n S_k^2\bigl(Y(\ell)\bigr) = \frac{n}{k-1} \sum_{i=1}^{k} \bigl[Y_i(\ell) - \overline{Y}_k(\ell)\bigr]^2.$$

Glynn and Heidelberger (1991) study truncation schemes for processes whose bias satisfies the condition

$$\sum_{j=1}^{\infty} \bigl|\mathrm{E}(X_j) - \mu\bigr| < \infty. \tag{24}$$

They propose rules for choosing the truncation index ℓ so that the replicate truncated mean $\overline{Y}_k(\ell)$ satisfies a central limit theorem. Two cases that satisfy condition (24) are $\mathrm{E}(X_j) - \mu = \mathrm{O}(1/j^r)$, for some constant $r > 1$, and $\mathrm{E}(X_j) - \mu = \mathrm{O}(\beta^j)$, for some constant $\beta \in [0, 1)$. The "big-oh" notation $\mathrm{O}(h(u))$ denotes a function $g(u)$ for which there exist constants c and u_0 such that $|g(u)| \leqslant c|h(u)|$ for all $u \geqslant u_0$.

Fishman (2001) considers the common practice of fixing the replication length $\ell + n$ (after a truncation index has been identified) and increasing the number of replications. Theorem 4 describes conditions for assuring proper coverage for the resulting CI. Assumptions (F.1)–(F.3), which feed into Theorem 4 as sufficient conditions, are satisfied by a variety of processes encountered in simulation output analysis, e.g., the AR(1) process, irreducible aperiodic finite-state-space Markov chains, and irreducible finite-state-space Markov processes. S_0 denotes the (possibly random) initial state of the underlying system.

(F.1) There exists a constant $\beta \in [0, 1)$ such that for any initial state s_0,

$$\mu_j(s_0) := \mathrm{E}(X_j | S_0 = s_0) = \mu + \mathrm{O}\bigl(\beta^j\bigr) \quad \text{as } j \to \infty.$$

This implies that $\mathrm{E}(Y_1(\ell) | S_0 = s_0) = \mu + \mathrm{O}(\beta^\ell / n)$ as $\ell \to \infty$ and $n \to \infty$.

(F.2) For each s_0 and n, there is a constant $\sigma_n^2 \in (0, \infty)$ such that

$$\begin{aligned} \sigma_n^2(\ell; s_0) &:= n \operatorname{Var}\bigl(Y_1(\ell) | S_0 = s_0\bigr) \\ &= \sigma_n^2 + \mathrm{O}\left(\frac{\beta^\ell}{n}\right) \quad \text{as } \ell \to \infty \text{ and } n \to \infty. \end{aligned} \tag{25}$$

Equation (25) implies $\sigma_n^2(\ell; s_0) \to \sigma_n^2$ as $\ell \to \infty$, independently of s_0.

(F.3) There is a constant $\sigma^2 \in (0, \infty)$ such that

$$\sigma_n^2 = \sigma^2 + \mathrm{O}\left(\frac{1}{n}\right) \quad \text{as } n \to \infty.$$

Hence for all s_0 and ℓ, $\sigma_n^2(\ell; s_0) \to \sigma^2$, as $n \to \infty$.

Theorem 4 (Fishman, 2001, Section 6.4). *If assumptions* (F.1)–(F.3) *hold*, *n is fixed*, *and* $\ell/\ln(k) \to \infty$ *as* $k \to \infty$ *and* $\ell \to \infty$, *then*

$$\frac{\overline{Y}_k(\ell) - \mu}{S_k(Y(\ell))/\sqrt{k}} \xrightarrow{d} \mathrm{N}(0, 1).$$

We see that the CI for μ will be of the form

$$\overline{Y}_k(\ell) \pm t_{1-\alpha/2,k-1} \frac{S_k(Y(\ell))}{\sqrt{k}}.$$

Further, if one fixes the number of observations n retained during a replication and attempts to compute a narrower CI by making more runs k, the theorem says that we can achieve the nominal coverage $1 - \alpha$ by forcing the truncation index ℓ to grow faster than the logarithm of k. This requirement is hard to implement in practice as the user has to consider three parameters (n, k and ℓ). This "systematic error" due to the initial transient is not as pertinent to the methods presented in Chapter 15 (e.g., the batch means method and the standardized time series method) and Chapter 16 (the regenerative method).

We finish this discussion by pointing out that Fishman (2001) and Glynn and Heidelberger (1991) give several additional sufficient conditions among n, k and ℓ that yield a central limit theorem for the grand replicate mean $\overline{Y}_k(\ell)$. For additional results and experiments related to this issue, see Alexopoulos and Goldsman (2004).

6 Density estimation

The probability density function is a fundamental concept in probability and statistics. Suppose that we possess a finite univariate sample, say $X_1, \ldots, X_n$ from an unknown density function f. *Density estimation* refers to the construction of an estimate of f using the observed data. In simulation experiments this need arises in input data analysis, in analysis of data from independent replications, and in the estimation of the marginal density function of a stationary process. We start with estimators for IID data; the case that has received most attention.

If one assumes that the data come from a parametric distribution, then the parameters can be estimated by "standard" methods, such as the method of maximum likelihood, and the goodness-of-fit can be assessed by a battery of tests (e.g., the omnibus chi-square test, the Kolmogorov–Smirnov test and the Anderson–Darling test). Unfortunately, this approach is quite restrictive because the commonly-used parametric families have unimodal density functions. Below we give a brief introduction to *nonparametric* methods for estimating the unknown density. In this case, we will allow the data themselves to guide the estimation of f. For a thorough treatment of this subject, we refer the eager reader to Härdle (1991), Silverman (1986), and many references therein. Multivariate density estimation is treated thoroughly by Scott (1992).

6.1 Histograms

The oldest and most commonly-used density estimator is the histogram. Given an origin x_0 and a bin width h, the bins are the intervals $[x_0 + \ell h, x_0 + (\ell + 1)h)$, for $\ell = 0, \pm 1, \pm 2, \ldots$. Notice that the leftmost interval can be open and the rightmost interval can be closed. The histogram is then defined by

$$\hat{f}_{\mathrm{H}}(x) = \frac{1}{nh}(\text{number of } X_i \text{ in the same bin as } x), \quad x \in \mathbb{R}.$$

The choice of the bin width h controls the smoothness of the histogram. Unfortunately, the choice of the origin and the bin width can have a severe impact on the histogram. Clearly, more robust methods are required. We start with the naive estimator, and proceed with various state-of-the-art estimators.

6.2 The naive estimator

By definition, the density function of an absolutely continuous random variable X obeys

$$f(x) = \lim_{h \to 0} \frac{1}{2h} \Pr(x - h < X < x + h), \quad x \in \mathbb{R}.$$

For given h, we can approximate the $\Pr(x - h < X < x + h)$ by the fraction of the sample that falls in the interval $(x - h, x + h)$. Then the *naive* estimator of f is

$$\hat{f}_{\mathrm{N}}(x) = \frac{1}{2nh}\bigl[\text{number of } X_i \text{ in } (x - h, x + h)\bigr], \quad x \in \mathbb{R}. \tag{26}$$

Notice that the naive estimator is neither continuous nor differentiable – it has jumps at the points $X_i \pm h$ and has zero derivative everywhere else.

To motivate the estimators in the following subsections, we write (26) as

$$\hat{f}_{\mathrm{N}}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} w\left(\frac{x - X_i}{h}\right),$$

where the weight function,

$$w(x) = \begin{cases} \frac{1}{2}, & \text{if } |x| < 1, \\ 0, & \text{otherwise}, \end{cases} \tag{27}$$

is the uniform density in the interval $(-1, 1)$.

6.3 The kernel estimator

The naive estimator in Equations (26) and (27) can be generalized by replacing the uniform weight by a *kernel* function $K(\cdot)$ obeying

$$\int_{-\infty}^{\infty} K(x)\,\mathrm{d}x = 1, \qquad \sup_x \bigl|K(x)\bigr| < \infty, \quad \text{and} \quad \lim_{|x|\to\infty} \bigl|xK(x)\bigr| = 0. \tag{28}$$

Most often, the kernel K is an even and symmetric density, e.g., the normal density. The general form of the kernel estimators is

$$\hat{f}_K(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right), \quad x \in \mathbb{R}, \tag{29}$$

and h is frequently called the *smoothing parameter* or the *bandwidth*. The kernel estimator inherits the properties of the kernel related to continuity and differentiability: if the kernel K is nonnegative (i.e., a density function), then $\hat{f}_K$ is also a probability density function as it integrates to one.

Clearly, the kernel estimator is the sum of “bumps” $K\{(x - X_i)/h\}/(nh)$ placed at the observations X_i. Although the function K determines the shape of the bumps, the bandwidth h affects the smoothness of the estimator. If h is chosen too small, the estimate becomes overly spurious. A few of the most common kernels are listed in Table 1 ($\mathbf{1}(\cdot)$ is the indicator function).

6.4 Bandwidth selection

Although the bandwidth h plays an important role, there are no firm rules for selecting it. A common approach relies on the minimization of the integrated MSE of the kernel estimator $\hat{f}_K$, namely,

$$\mathrm{IMSE}(h) = \int_{-\infty}^{\infty} \mathrm{E}\bigl[\bigl(\hat{f}_K(x; h) - f(x)\bigr)^2\bigr]\,\mathrm{d}x.$$

Assuming that the kernel is a symmetric density and that the density f has continuous derivatives of all orders needed (Silverman, 1986, Section 3.3), the

Table 1.
Common kernels

Name	$K(x)$
Triangular	$(1 - \lvert x\rvert)\mathbf{1}(\lvert x\rvert < 1)$
Epanechnikov	$\frac{3}{4}(1 - x^2)\mathbf{1}(\lvert x\rvert < 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)$
Quartic	$\frac{15}{16}(1 - x^2)^2\mathbf{1}(\lvert x\rvert < 1)$
Triweight	$\frac{35}{32}(1 - x^2)^3\mathbf{1}(\lvert x\rvert < 1)$
Cosine	$\frac{\pi}{4}\cos(\pi x/2)\mathbf{1}(\lvert x\rvert < 1)$

IMSE is asymptotically (as $n \to \infty$) approximated by the asymptotic mean integrated squared error (AMISE)

$$\mathrm{AMISE}(h) = \frac{R(K)}{nh} + \frac{h^4 R(f'')\kappa^4}{4}, \tag{30}$$

where $R(f'') = \int_{-\infty}^{\infty} [f''(x)^2]\,\mathrm{d}x$ is the *roughness* of f'' and $\kappa^2 = \int_{-\infty}^{\infty} x^2 \times K(x)\,\mathrm{d}x$. (If the kernel is a symmetric density, then κ^2 is the variance of K.) Expression (30) quantifies the effects of the bandwidth h: the first term (from the integrated variance) is large when h is too small, whereas the second term (from the integrated squared bias) is large when h is too large.

The minimizer of AMISE(h), namely

$$h_{\mathrm{AMISE}} = \left[\frac{R(K)}{nR(f'')\kappa^4}\right]^{1/5}, \tag{31}$$

depends on the unknown quantity $R(f'')$ but offers a couple of interesting suggestions. First, the optimal bandwidth will approach zero as the sample size increases, but at a very slow rate. Second, small bandwidths will be required for rapidly varying densities f. Silverman (1986), Section 6.1.5, and Jones et al. (1996) review a variety of bandwidth selection methods, in particular, the method of Sheather and Jones (1991). An older simple rule is given in Section 3.4.1 of Silverman (1986):

$$h = 1.06 \min\{S_n(X), R/1.3\} n^{-1/5}, \tag{32}$$

where R is their inter-quartile range (difference between the 75 and 25 percentiles).

Remark 3. Epanechnikov's kernel from Table 1 is optimal as it minimizes AMISE(h_{AMISE}); see Bartlett (1963) and Epanechnikov (1969).

Remark 4. On top of conditions (28), assume that $\int_{-\infty}^{\infty} uK(u)\,\mathrm{d}u = 0$. Then the mean of $\hat{f}_K$ is

$$\begin{aligned}
\int_{-\infty}^{\infty} x\hat{f}_K(x)\,\mathrm{d}x &= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}\frac{1}{h}xK\left(\frac{x-X_i}{h}\right)\mathrm{d}x \\
&\quad \text{(using the transformation } u = (x-X_i)/h\text{)} \\
&= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty}(X_i + uh)K(u)\,\mathrm{d}u \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i\int_{-\infty}^{\infty}K(u)\,\mathrm{d}u + \frac{1}{n}\sum_{i=1}^{n}h\int_{-\infty}^{\infty}uK(u)\,\mathrm{d}u \\
&= \frac{1}{n}\sum_{i=1}^{n}X_i,
\end{aligned}$$

the sample mean of the data. Notice that the last expression is *not* the mean $\mathrm{E}[\hat{f}_K(x)]$.

Working similarly, one can show that the second moment of $\hat{f}_K(x)$ can be written as

$$\int_{-\infty}^{\infty} x^2 \hat{f}_K(x)\,\mathrm{d}x = \frac{1}{n}\sum_{i=1}^{n} X_i^2 + h^2\kappa^2.$$

Hence the variance of the density $\hat{f}_K$ is

$$\frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X}_n)^2 + h^2\kappa^2,$$

which is the sample second central moment of the data inflated by the term $h^2\kappa^2$.

Example 4. Table 2 contains 100 interarrival times from two customer classes into a system. Unfortunately, the data collector failed to record the proper class type; as a result, we cannot "separate" the data into the respective sets and proceed with each set.

Figure 1 displays density estimates based on the Gaussian kernel and bandwidths h equal to 6.258 and 2, respectively, in conjunction with a histogram based on a bin width of 10. The first value of h is based on Equation (32). The estimates were computed using MATLAB. Although both kernel density curves capture the bimodality of the data, the estimate based on $h = 2$ is very "jagged".

6.5 The variable kernel estimator

A potential problem arises when the kernel estimator is applied to data from long-tailed distributions: since the bandwidth h is fixed across the range of the

Table 2.
Interarrival times from 2 customer classes

11.97	2.04	12.02	2.81	11.45	8.19	1.05	7.35	13.42	2.28
24.22	14.44	17.42	21.47	38.53	41.06	29.89	20.40	44.11	38.31
13.48	1.49	1.67	10.44	29.79	9.21	1.88	2.94	6.07	3.49
25.43	21.05	30.31	37.41	78.71	19.95	42.67	31.26	30.55	19.09
14.49	2.10	6.36	15.36	8.17	6.75	4.37	6.98	3.83	17.30
36.61	27.80	31.72	69.06	20.34	36.21	44.65	52.72	12.83	21.00
5.12	17.86	17.76	13.16	4.31	5.92	36.88	3.53	10.53	5.05
22.22	40.69	37.38	32.02	32.71	17.28	16.13	20.45	38.72	24.35
2.04	8.63	4.10	8.26	4.86	7.68	11.93	14.31	1.02	10.11
19.57	14.07	12.95	21.64	19.52	18.02	29.37	15.33	21.12	23.35

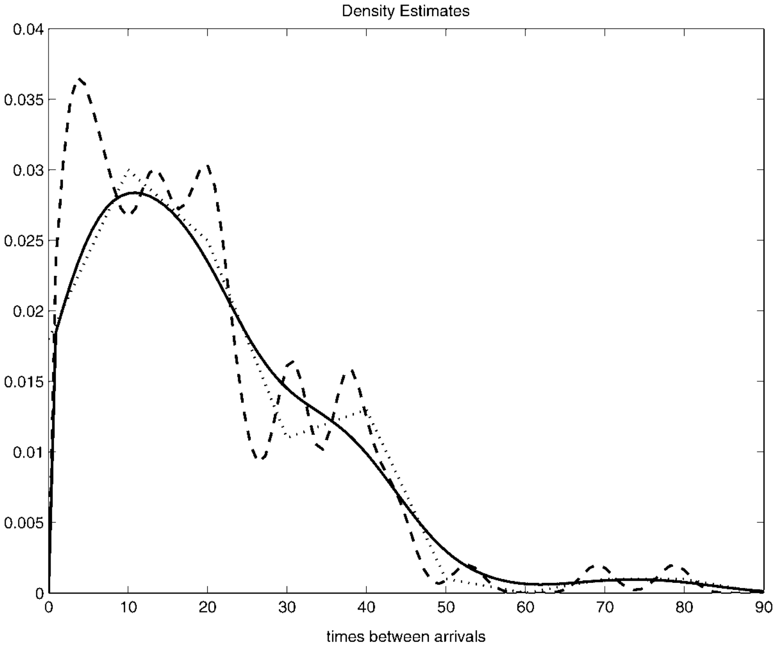


Fig. 1. Density estimates for the data set in Table 2 based on Gaussian kernels with bandwidths $h = 6.258$ (obtained from (32)) and $h = 2$. (···· histogram; — Gaussian kernel ($h = 6.258$); - - Gaussian kernel ($h = 2$).)

data, there is a tendency for spurious "noise" at the tails of the estimate due to the local sparsity of the data. This problem can be addressed as follows.

Let k be a positive integer, and define $d_{i,k}$ to be the distance from X_i to the kth closest observation among the set of remaining data. The *variable kernel* estimator with bandwidth h is defined by

$$\hat{f}_{\mathrm{VK}}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{hd_{i,k}} K\left(\frac{x - X_i}{hd_{i,k}}\right), \quad x \in \mathbb{R}. \tag{33}$$

Since the window width of the kernel placed at the point X_i is proportional to $d_{i,k}$, the observations in sparse regions will be associated with flatter kernels. As expected, for fixed k, the overall degree of smoothness in the estimate $\hat{f}_{\mathrm{VK}}$ will depend on the choice of h. A typical choice of k is $k \approx \sqrt{n}$.

Again, if K is a density, then $\hat{f}_{\mathrm{VK}}$ will also be a density that inherits the continuity and differentiability properties of K.

6.6 Bounded domains

Often, the natural support of a density is an interval bounded on one or both sides. An example, is the density of the interarrival times in Example 4.

In this case, the estimated density should also be zero outside the support of the unknown density. Silverman (1986) discussed this issue and a variety of ways to address it. We discuss a singe approach for the case where X is a nonnegative variable. The method starts by augmenting the original set of data by adding their reflections with respect to zero; the new data set becomes $\{\pm X_1, \pm X_2, \ldots, \pm X_n\}$. If $\tilde{f}_K$ is a kernel estimate constructed based on the augmented set of data, then a natural estimate based on the original data set is the "folded" estimate

$$\hat{f}_K(x) = \begin{cases} 2\tilde{f}_K(x) & \text{for } x \geqslant 0, \\ 0 & \text{for } x < 0. \end{cases}$$

6.7 Density estimation for stationary processes

Kernel estimation for dependent stationary processes has received increasing interest during the last two decades. As we mentioned in the Introduction, the main objective is the estimation of the marginal density of a stationary process.

The literature considers two classes of estimators. The first class includes nonrecursive estimators of the form

$$f_n(x) = \frac{1}{n}\sum_{i=1}^{n} K_n(x - X_i),$$

where the $K_n(\cdot)$ are kernel functions and the bandwidth for each n is part of K_n. (In what follows we replace the subscript "K" by "n" to denote the sample size.) Asymptotic properties of such estimators (e.g., convergence in mean, quadratic means and normality) are studied in Rosenblatt (1970) for Markov processes, and in Masry (1983), Robinson (1983, 1986) for strongly mixing processes.

Below we focus on estimators of the form

$$\hat{f}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h_i} K\left(\frac{x - X_i}{h_i}\right), \quad x \in \mathbb{R}, \tag{34}$$

where $\{h_i\}$ is a bandwidth sequence. The estimators (34) are attractive because they can be computed recursively by

$$\hat{f}_n(x) = \frac{n-1}{n}\hat{f}_{n-1}(x) + \frac{1}{nh_n} K\left(\frac{x - X_n}{h_n}\right).$$

Further we review results from Masry (1986). The reader is also pointed to Masry (1989), Masry and Györfi (1987), Györfi and Masry (1990), and several other references therein, for additional properties and alternative estimators. Clearly, the asymptotic results for the stationary processes generalize results for IID processes.

We will assume that the bandwidth sequence satisfies $\lim_{n\to\infty} h_n = 0$ and $\lim_{n\to\infty} nh_n = \infty$. A popular choice is

$$h_n = hn^{-\lambda}, \quad 0 < \lambda < 1, \tag{35}$$

a generalization of (32).

Theorem 5 shows that the estimator (34) is asymptotically (as $n \to \infty$) unbiased at each continuity point of f and displays the finite-sample bias. The "little-oh" notation $\mathrm{o}(h(n))$ denotes a function $g(n)$ such that $g(n)/h(n) \to 0$ as $n \to \infty$.

Theorem 5 (Masry, 1986). *Suppose that the kernel K satisfies conditions (28), and let x be a continuity point of f. Then*

$$\mathrm{E}\bigl[\hat{f}_n(x)\bigr] \to f(x) \quad \textit{as } n \to \infty.$$

Suppose in addition that f is $r+1$ times continuously differentiable at the point x, $\sup_u |f^{(r+1)}(u)| < \infty$, and

$$\int_{-\infty}^{\infty} |u|^j \bigl|K(u)\bigr| \,\mathrm{d}u < \infty \quad \textit{for } j = 1, \ldots, r+1.$$

Also assume that there are finite constants β_ℓ such that, as $n \to \infty$,

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{h_i}{h_n}\right)^{\ell} \to \beta_\ell \quad \textit{for } \ell = 1, \ldots, r+1.$$

Then

$$\mathrm{E}\bigl[\hat{f}_n(x)\bigr] = f(x) + \bigl(1 + \mathrm{o}(1)\bigr)\sum_{\ell=1}^{r}\frac{c_\ell \beta_\ell (-h_n)^\ell}{\ell!} f^{(\ell)}(x) + \mathrm{O}\bigl(h_n^{(r+1)}\bigr), \tag{36}$$

where

$$c_\ell = \int_{-\infty}^{\infty} u^\ell K(u)\,\mathrm{d}u \quad \textit{for } \ell = 1, \ldots, r.$$

Remark 5. If K is an even function, then $c_\ell = 0$ when ℓ is odd. For $r \geqslant 2$ in Theorem 5, the dominant term in the right-hand side of Equation (36) is $c_2\beta_2 f^{(2)}(x)/2$ so that the bias of $\hat{f}_n(x)$ decreases at the rate h_n^2. Also the bandwidth sequence in Equation (35) yields

$$\beta_\ell = \frac{1}{1 - \lambda\ell} \quad \text{if } 0 < \lambda\ell < 1.$$

Theorem 6 establishes the rate of convergence of the variance of $\hat{f}_n(x)$ to zero and the rate at which the covariance between values of $\hat{f}_n$ at distinct points approaches zero.

Theorem 6 (Masry, 1986). *Suppose that*:

(a) *The joint probability density $f(x, y; j)$ of the RVs X_i and X_{i+j} exists and satisfies*

$$\left|f(x, y; j) - f(x)f(y)\right| \leqslant M \quad \textit{for all } (x, y) \textit{ and } j \geqslant 1,$$

where M is a finite constant.

(b) *$\{X_j\}$ is strongly α-mixing with*

$$\sum_{j=1}^{\infty} \alpha_j^p < \infty \quad \textit{for some } p \in \left(0, \frac{1}{2}\right).$$

(c) *The kernel K satisfies conditions (28).*

(d) *The bandwidth sequence satisfies*

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{h_n}{h_i}\right)^r \to \theta_r < \infty \quad \textit{as } n \to \infty,$$

for $r \in [1, 2)$, where θ_r are constants.

Then at all continuity points of f,

$$\lim_{n\to\infty} nh_n \operatorname{Var}\left[\hat{f}_n(x)\right] = \theta_1 f(x) \int_{-\infty}^{\infty} K^2(u)\,\mathrm{d}u \tag{37}$$

and

$$\lim_{n\to\infty} nh_n \operatorname{Cov}\left[\hat{f}_n(x), \hat{f}_n(y)\right] = 0 \quad \textit{for } x \neq y.$$

Remark 6. The asymptotic variance expression (37) coincides with the expression obtained by Parzen (1962) for independent samples. Additional terms due to the serial dependencies are listed in the proof of Theorem 3 of Masry (1986).

Remark 7. If f is three times continuously differentiable ($r = 2$ in Theorem 5) and K is an even function (e.g., a density), we have

$$\operatorname{Bias}\left[\hat{f}_n(x)\right] \approx \frac{c_2\beta_2 f^{(2)}(x)}{2} h_n^2$$

and

$$\operatorname{Var}\left[\hat{f}_n(x)\right] \approx \frac{\theta_1 f(x) \int_{-\infty}^{\infty} K^2(u)\,\mathrm{d}u}{nh_n}.$$

One can show that the bandwidth that minimizes the mean squared error $\mathrm{MSE}[\hat{f}_n(x)]$ is $h_n = hn^{-1/5}$ and the convergence rate of $\mathrm{MSE}[\hat{f}_n(x)]$ to zero is $n^{-4/5}$. Further, $\beta_2 = 5/3$ and $\theta_1 = 5/6$.

We finish our discussion with the asymptotic normality of the recursive estimators. The following theorem is similar to Theorem 8 of Masry (1986).

Theorem 7. *Assume that the conditions of Theorem 6 hold. Also assume that there exist a constant* $\gamma \in (0,1)$ *and a sequence* $\{m_n\}$ *of integers with* $\lim_{n\to\infty} m_n = 0$ *such that* $\lim_{n\to\infty} nh_n^{3-2\gamma} = 0$, $m_n = \mathrm{o}(n^{1/2}h_n^{3/2-\gamma})$, *and*

$$\sqrt{\frac{n}{h_n}} \sum_{j=m_n}^{\infty} \alpha_j^{1-\gamma} \to 0 \quad \textit{as } n \to \infty.$$

Then at each continuity point of f *with* $f(x) > 0$,

$$\frac{\hat{f}_n(x) - \mathrm{E}[\hat{f}_n(x)]}{\sigma_f/\sqrt{nh_n}} \xrightarrow{d} \mathrm{N}(0,1) \quad \textit{as } n \to \infty, \tag{38}$$

where

$$\sigma_f^2 = \theta_1 f(x) \int_{-\infty}^{\infty} K^2(u)\,\mathrm{d}u.$$

Equation (38) is not convenient for computing confidence intervals for $f(x)$ because it does not involve the actual error $\hat{f}_n(x) - f(x)$. To overcome this problem we write

$$\frac{\hat{f}_n(x) - \mathrm{E}[\hat{f}_n(x)]}{\sigma_f/\sqrt{nh_n}} = \frac{\hat{f}_n(x) - f(x)}{\sigma_f/\sqrt{nh_n}} - \frac{\mathrm{E}[\hat{f}_n(x)] - f(x)}{\sigma_f/\sqrt{nh_n}}. \tag{39}$$

Under the conditions in Remark 7, we have $\mathrm{Bias}[\hat{f}_n(x)] = \mathrm{O}(h_n^2)$. Hence if $\{h_n\}$ is such that $\lim_{n\to\infty} nh_n^5 = 0$, the last term in Equation (39) converges to zero and Slutsky's theorem (Karr, 1993, Theorem 5.20), implies

$$\frac{\hat{f}_n(x) - f(x)}{\sigma_f/\sqrt{nh_n}} \xrightarrow{\mathrm{d}} \mathrm{N}(0,1) \quad \text{as } n \to \infty. \tag{40}$$

Equation (40) yields the following approximate $1-\alpha$ CI for $f(x)$:

$$\hat{f}_n(x) \pm z_{1-\alpha_2} \frac{\hat{\sigma}_f}{\sqrt{nh_n}},$$

where

$$\hat{\sigma}_f = \left[\theta_1 \hat{f}_n(x) \int_{-\infty}^{\infty} K^2(u)\,\mathrm{d}u\right]^{1/2}$$

is an estimator for σ_f.

Notice that the condition $nh_n^5 \to 0$ is not satisfied by Silverman's rule in Equation (32) or by the bandwidth sequence in Remark 7 that minimizes

MSE$[\hat{f}_n(x)]$. Equations (38) and (40) reveal that the rate of the pointwise convergence of the recursive kernel estimator $\hat{f}_n(x)$ to the normal distribution is $(nh_n)^{-1/2}$ or $n^{-(1-\lambda)/2}$ for bandwidth sequences of the form $h_n = hn^{-\lambda}$, with $1/5 < \lambda < 1$ used to obtain Equation (40). Such rates are significantly slower that the usual rate of $n^{-1/2}$ for point estimators for means, but typical for kernel estimators.

Remark 8. For Markov chains (not necessarily discrete-time ones) one can obtain the convergence rate of $n^{-1/2}$ by means of the "look-ahead" density estimation method of Henderson and Glynn (2001).

Example 5. Consider an $M/G/1$ queueing system where units arrive according to a Poisson process with a mean interarrival time $\mathrm{E}(A) = 10$ and experience service times from the distribution with density function

$$g(x) = 0.75\left(\frac{1}{4}x\mathrm{e}^{-x/2}\right) + 0.75\left(\frac{1}{19!}x^{19}\mathrm{e}^{-x}\right).$$

This model is applicable when the units are classified into two types, 1 and 2, with respective probabilities 0.75 and 0.25; type 1 units require service times from the gamma distribution with shape parameter 2 and scale parameter 2; and type 2 units require service times from the gamma distribution with shape parameter 20 and scale parameter 1. In other words, the service time distribution is a mixture of two gamma distributions with modes of 2 and 19. Since the mean service time is $\mathrm{E}(S) = 0.75(4) + 0.25(20) = 8$, the traffic intensity is $\nu = 8/10 < 1$ and the system is stable.

The objective of this experiment is the estimation of the density of the time a unit spends in the system (flow time) in steady state. The substantial distance between the modes of the gamma distributions associated with the service times causes the steady-state density of the flow times to be bimodal. We made a single run starting with the empty and idle state and collected 7000 observations. To reduce the effect of the transient phase, we discarded the first 5000 observations and computed the density estimates in Figure 2 using the remaining 2000 observations. The data and the estimates were generated using MATLAB. The kernel estimate is based on the Gaussian kernel with bandwidth $10n^{-1/5}$ whereas the histogram is based on a bin width of 5. Both estimates clearly indicate the bimodal nature of the steady-state density of the flow times.

7 Summary

This chapter served as a "precursor" to several forthcoming chapters. Its scope was the introduction of the reader to the statistical issues involved in simulation experiments. We started with a review of probabilistic and statistical

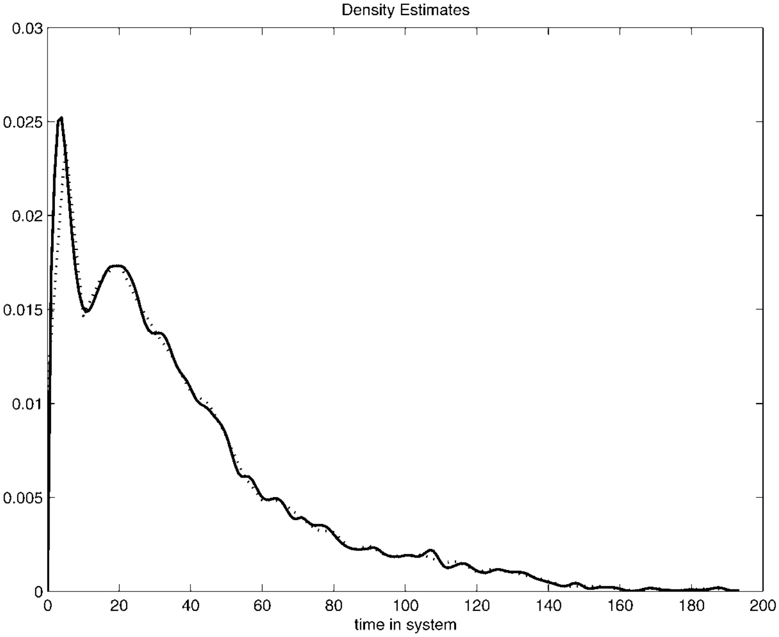


Fig. 2. Kernel estimate for the limiting distribution of the system flow time in an $M/G/1$ system. (··· histogram; — Gaussian kernel.)

concepts that suffice for analyzing data from finite-horizon simulations. Then we laid out the issues related to steady-state simulations. Since these issues are multi-faceted and challenging, we reviewed the basic concepts from stochastic processes and a few limit theorems that are required to carry out the necessary statistical analyses.

We proceeded with a review of a research area that has received little attention in the simulation domain, density estimation. Within this subject, we started with a review of techniques for IID data and concluded with a recent results related to data from stationary sequences. Since this subject will not receive further consideration in this volume, we encourage the eager reader to study the literature.

We finish with a couple of observations. First, kernel density estimators converge to the respective densities with rates that are significantly lower than the usual rate of $n^{-1/2}$ for point estimators for means (an exception is the estimator in Henderson and Glynn, 2001). Second, a problem that has received little or no attention in the literature is the generation of realizations from kernel density estimators (or the respective CDFs). This is an important issue as the efficient generation of random variates is a paramount issue in simulation experiments.

Acknowledgements

I would like to thank Jim Wilson for many fruitful discussions and Melike Meterelliyoz for her careful reading.

References

Alexopoulos, C., Goldsman, D. (2004). To batch or not to batch. *ACM Transactions on Modeling and Computer Simulation* 14 (1), 76–114.

Alexopoulos, C., Seila, A.F. (1998). Output data analysis. In: Banks, J. (Ed.), *Handbook of Simulation*. Wiley, New York, pp. 225–272.

Alexopoulos, C., Goldsman, D., Serfozo, R.F. (2006). Stationary processes: Statistical estimation. In: Balakrishnan, N., Read, C., Vidakovic, B. (Eds.), *Encyclopedia of Statistical Sciences*, 2nd edition. Hoboken, NJ, pp. 7991–8006.

Bartlett, M.S. (1963). Statistical analysis of density functions. *Sankhyā Series A* 25, 245–254.

Billingsley, P. (1968). *Convergence of Probability Measures*. Wiley, New York.

Blomqvist, N. (1967). The covariance function of the $M/G/1$ queueing system. *Skandinavisk Aktuarietidskrift* 50, 157–174.

Castellana, J.V., Leadbetter, M.R. (1986). On smoothed probability density estimation for stationary processes. *Stochastic Processes and Their Applications* 21, 179–193.

Chatfield, C. (1989). *The Analysis of Time Series: An Introduction*, 4th edition. Chapman and Hall, New York.

Chow, Y.S., Robbins, H. (1965). On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *Annals of Mathematical Statistics* 36, 457–462.

Durrett, R. (2005). *Probability: Theory and Examples*, 3rd edition. Brooks/Cole–Thomson Learning, Belmont, CA.

Epanechnikov, V.A. (1969). Nonparametric estimation of a multidimensional probability density. *Theory of Probability and Its Applications* 14, 153–158.

Fishman, G.S. (2001). *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag, New York.

Glynn, P.W., Heidelberger, P. (1991). Analysis of initial transient deletion for replicated steady-state simulations. *Operations Research Letters* 10, 437–443.

Glynn, P.W., Iglehart, D.L. (1985). Large-sample theory for standardized time series: An overview. In: Gantz, D., Blais, G., Solomon, S. (Eds.), *Proceedings of the 1985 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 129–134.

Gross, D., Harris, C.M. (1998). *Fundamentals of Queueing Theory*, 3rd edition. Wiley, New York.

Gross, D., Shortle, J.F., Fisher, M.J., Masi, D.M.B. (2002). Difficulties in simulating queues with Pareto services. In: Yucesan, E., Chen, C.H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 407–415.

Györfi, L., Masry, E. (1990). The L_1 and L_2 strong consistency of recursive density estimation from dependent samples. *IEEE Transactions on Information Theory* 36 (3), 531–539.

Härdle, W. (1991). *Smoothing Techniques with Implementation in S*. Springer-Verlag, New York.

Henderson, S.G., Glynn, P.W. (2001). Computing densities for Markov chains via simulation. *Mathematics of Operations Research* 26 (2), 375–400.

Hogg, R.V., Craig, A.T. (1995). *Introduction to Mathematical Statistics*, 5th edition. Prentice Hall, Upper Saddle River, NJ.

Jones, M.C., Marron, J.S., Sheather, S.J. (1996). A brief survey of bandwidth selection for density estimation. *Journal of the Americal Statistical Association* 91 (433), 402–407.

Karr, A.F. (1993). *Probability*. Springer-Verlag, New York.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Law, A.M., Kelton, W.D., Koenig, L.W. (1981). Relative width sequential confidence intervals for the mean. *Communications in Statistics B* 10, 29–39.

Lindley, D.V. (1952). The theory of queues with a single server. *Proceedings of the Cambridge Philosophical Society* 48, 277–289.

Masry, E. (1983). Probability density estimation from sampled data. *IEEE Transactions on Information Theory* 29, 696–709.

Masry, E. (1986). Recursive probability density estimation for weakly dependent stationary processes. *IEEE Transactions on Information Theory* 32 (2), 254–267.

Masry, E. (1989). Nonparametric estimation of conditional probability densities and expectations of stationary processes: Strong consistency and rates. *Stochastic Processes and Their Applications* 31, 109–127.

Masry, E., Györfi, L. (1987). Strong consistency and rates for recursive probability density estimators of stationary processes. *Journal of Multivariate Analysis* 22, 79–93.

Nadas, A. (1969). An extension of the theorem of Chow and Robbins on sequential confidence intervals for the mean. *Annals of Mathematical Statistics* 40, 667–671.

Parzen, E. (1962). On estimation of probability density function and mode. *Annals of Mathematical Statistics* 33, 1065–1076.

Robinson, P.M. (1983). Nonparametric estimators for time series. *Journal of Time Series Analysis* 4, 185–207.

Robinson, P.M. (1986). On the consistency and finite sample properties of nonparametric kernel time series regression, autoregression and density estimators. *Annals of the Institute of Mathematical Statistics* 38, 539–549.

Rosenblatt, M. (1956). A central limit theorem and strong mixing condition. *Proceedings of the National Academy of Sciences* 42, 43–47.

Rosenblatt, M. (1970). Density estimates and Markov sequences. In: Puri, M. (Ed.), *Nonparametric Techniques in Statistical Inference*. Cambridge University Press, London, pp. 199–210.

Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley-Interscience, New York.

Serfozo, R. (1999). *Introduction to Stochastic Networks*. Springer-Verlag, New York.

Sheather, S.J., Jones, M.C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society B* 53, 683–690.

Silverman, B.W. (1986). *Density Estimation*. Chapman and Hall, London.

Starr, N. (1966). The performance of a statistical procedure for the fixed-width interval estimation for the mean. *Annals of Mathematical Statistics* 37 (1), 36–50.

Stidham, S. (1974). A last word on $L = \lambda W$. *Operations Research* 22, 417–421.

Welch, P.D. (1983). The statistical analysis of simulation results. In: Lavenberg, S. (Ed.), *The Computer Performance Modeling Handbook*. Academic Press, New York, pp. 268–328.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13009-1

Chapter 9

Subjective Probability and Bayesian Methodology

Stephen E. Chick
INSEAD, Technology Management Area,
Boulevard de Constance, 77300 Fontainebleau, France
E-mail: stephen.chick@insead.edu

Abstract

Subjective probability and Bayesian methods provide a unified approach to handle not only randomness from stochastic sample-paths, but also uncertainty about input parameters and response metamodels. The chapter surveys some basic concepts, principles and techniques useful for a subjective Bayesian approach to uncertainty analysis, data collection plans to reduce input uncertainty, response surface modeling, and expected value-of-information approaches to experimental designs for selection procedures. Some differences from the classical technique are identified.

Introduction

If simulation is defined to be the analysis of stochastic processes through the generation of sample paths of the process, then Bayesian and subjective probability methods apply in several ways for the modeling, design and analysis of simulation experiments. By Bayesian methods, we refer here to parameter inference through repeated observations of data with Bayes' rule. Examples in simulation are input parameter inference using field data or the inference of metamodel parameters from simulation replications. The Bayesian approach entails postulating a 'prior probability' model that describes a modeler's initial uncertainty about parameters, a likelihood function that describes the distribution of data, given that a parameter holds a specific value, and Bayes' rule, which provides a coherent method of updating beliefs about uncertainty when data becomes available. By subjective probability, we refer to probability assessments for all unknown quantities, including parameters that can be inferred with Bayes' rule, as well as unknown quantities for which parameters cannot be inferred from repeated sampling of data (e.g., one-shot deals like the total potential market size for a particular new product from a simulated manufacturing facility). By frequentist, we mean methods based on sampling statistics from repeated observations, such as maximum likelihood (MLE) methods to fit input parameters, or ranking and selection procedures that provide worst-case probability of correct selection guarantees based on repeated applications of the procedure. The chapter describes applications of Bayesian and subjective probability methods in simulation, and identifies some ways that the Bayesian approach differs from the frequentist approach that underlies much of simulation theory.

In the simulation community, Glynn (1986) first suggested Bayesian applications of *uncertainty analysis* for statistical input parameter uncertainty. In that paper, the traditional role of estimating $\alpha = h(\mathrm{E}[Y])$ is extended to account for statistical input parameter uncertainty, so $\alpha(\theta) = h(\mathrm{E}[Y|\theta])$ depends upon unknown parameters with distribution $p(\theta)$ that can be updated with data from the modeled system. Three questions he poses are: (i) how to estimate the distribution of $\alpha(\Theta)$ induced by the random variable Θ, (ii) how to estimate the mean $\mathrm{E}[\alpha(\Theta)]$, and (iii) estimation of credible sets, e.g., finding a, b so the probability $\Pr(\alpha(\Theta) \in [a, b])$ equals a pre-specified value, like 0.95. Chick (1997) provided a review of the few works to that date that applied Bayesian ideas to simulation, then suggested a broader range of application areas than uncertainty analysis, including *ranking and selection*, *response surface modeling*, and *experimental design*.

The basic goal is to understand how uncertainty and decision variables affect system performance, so that better decisions can be made. The premise in this chapter is that representing *all* uncertainty with probability can aid decision-makers that face uncertainty. Stochastic uncertainty, the randomness in simulation models that occurs even if all parameters are known, is already widely modeled with probability. The subjective Bayesian approach also models input parameter and response surface uncertainty with probability distributions, a practice that has been less common in stochastic process simulation.

Probabilistic models for uncertainty are increasingly employed for at least three reasons. One, doing so allows the modeler to quantify how parameter uncertainty influences the performance of a simulated system. Parameters of models of real systems are rarely known with certainty. The Bayesian approach for uncertainty analysis overcomes some limitations of the classical approach for parameter and model selection (Chick, 2001; Barton and Schruben, 2001; Draper, 1995). Two, simulation experiments can be designed to run more efficiently (Chick and Inoue, 2001a; Santner et al., 2003). And three, Bayesian and subjective probability methods are not new but are increasingly implemented due to the development of improved computing power and Markov chain Monte Carlo (MCMC) methods (Gilks et al., 1996).

This chapter describes the subjective Bayesian formulation for simulation. Section 1 presents the basics of subjective probability and Bayesian statistics in the context of quantifying uncertainty about one statistical input parameter. Section 2 summarizes the main ideas and techniques for addressing three main challenges in implementing Bayesian inference: maximization, integration, and sampling variates from posterior distributions. Section 3 addresses input distribution selection when multiple candidate distributions exist. Section 4 presents a joint formulation for input and output modeling, and reviews applications for data collection to reduce input uncertainty in a way that reduces output uncertainty, and for response surface modeling and simulation experiments to reduce response surface uncertainty. Section 5 describes applications of Bayesian expected value of information methods for efficiently selecting the best of a finite set of simulated alternatives.

Simulation research with Bayesian methods has grown rapidly since the mid to late 1990s. A partial reference list is Chen and Schmeiser (1995), Chen (1996), Scott (1996), Nelson et al. (1997), Chen et al. (1999), Cheng (1999), Lee and Glynn (1999), Andradóttir and Bier (2000), Chick and Inoue (2001a, 2001b), Chick (2001), Cheng and Currie (2003), Steckley and Henderson (2003), Chick et al. (2003), Zouaoui and Wilson (2003, 2004), Ng and Chick (2004), as well as applications to insurance, finance, waterway safety, civil engineering and other areas described in the *Winter Simulation Conference Proceedings*. Work on deterministic simulation with potentially important implications for stochastic simulation includes O'Hagan et al. (1999), Kennedy and O'Hagan (2001), Craig et al. (2001), Santner et al. (2003). Excellent references for subjective probability and Bayesian statistics in general, not just in simulation, include Lindley (1972), Berger (1985), Bernardo and Smith (1994), with special mention for de Finetti (1990), Savage (1972) and de Groot (1970).

1 Main concepts

A stochastic simulation is modeled as a deterministic function of several inputs,

$$Y_r = g(\boldsymbol{\theta}_p, \boldsymbol{\theta}_e, \boldsymbol{\theta}_c; \mathbf{U}_r), \tag{1}$$

where Y_r is the output of the rth replication. The vector of statistical input parameters $\boldsymbol{\theta}_p = (\theta_1, \theta_2, \ldots, \theta_{n_p})$ describes n_p sources of randomness whose values can be inferred from field data. For example, θ_1 may be a two-dimensional parameter for log-normally distributed service times, and θ_2 may be defect probabilities inferable from factory data. Environmental parameters $\boldsymbol{\theta}_e$ are beyond the control of a decision maker, and no data is available for inference. Examples are the total potential market size for a new product, the general economic climate for a high-level model, or the actual climate for a production process influenced by temperature or humidity. The vector $\boldsymbol{\theta}_c$ represents all control parameters (decision variables) under direct control of the decision maker, such as production capacity, supply chain operating procedures, scheduling policies, and the number of servers at each node in a service system.

Random output Y_r for replication r can be generated even for the same inputs $(\boldsymbol{\theta}_{cr}, \boldsymbol{\theta}_{pr}, \boldsymbol{\theta}_{er})$ by sampling different portions of a random number

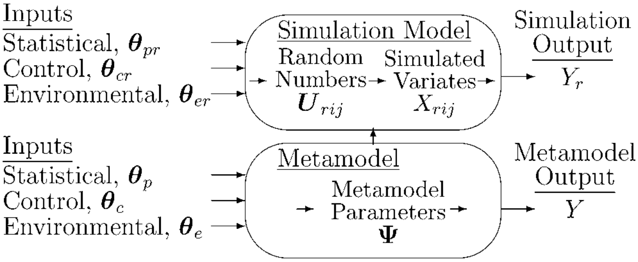

Inputs
Statistical, $\boldsymbol{\theta}_{pr}$
Control, $\boldsymbol{\theta}_{cr}$
Environmental, $\boldsymbol{\theta}_{er}$
Simulation Model
Random Numbers $\boldsymbol{U}_{rij}$
Simulated Variates X_{rij}
Simulation Output
Y_r
Inputs
Statistical, $\boldsymbol{\theta}_p$
Control, $\boldsymbol{\theta}_c$
Environmental, $\boldsymbol{\theta}_e$
Metamodel
Metamodel Parameters
$\boldsymbol{\Psi}$
Metamodel Output
Y

Fig. 1. Simulation takes multiple types of inputs and metamodels predict outputs for unsimulated input values.

stream $\mathbf{u}_r$, to obtain random variates x_{rij} (the jth simulated variate using input parameter θ_i during replication r), as in the top row of Figure 1. We use upper case for random variables, lower case for realizations, and bold-face to emphasize that a quantity is a vector. We may suppress the r, as in X_{ij}, to describe data collected from the actual system being simulated. That data would be used to infer the parameters of the statistical distributions to describe the system.

One reason a simulation experiment may be run is to estimate the function $g(\cdot)$ because its exact form is not known. *Metamodels* can be used to predict the output of a simulation model (or the simulated system) when a full simulation takes a long time to run. Section 4 describes some Bayesian methods to describe uncertainty about the parameters $\boldsymbol{\psi}$ of a metamodel.

A subjective probabilist represents all uncertain quantities with probability distributions. Uncertainty about statistical input parameters and environmental parameters are described as random quantities by the subjective Bayesian framework; we use a prior distribution $\pi(\boldsymbol{\theta}_p, \boldsymbol{\theta}_e)$ for the random quantity $(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e)$. The specification of prior distributions and Bayesian inference with data are discussed in Section 1.1. Loss functions and the expected value of information follow in Section 1.2, with uncertainty analysis in Section 1.3.

1.1 Bayesian modeling

An important simulation design issue is the selection of appropriate input distributions to characterize the stochastic behavior of the modeled system (see Chapter 6). Failure to select appropriate input distributions can result in misleading simulation output, and therefore poor system design decisions. This section reviews basic ideas and important theorems for inferring statistical parameters $\boldsymbol{\theta}_p$ from data with the Bayesian formalism. To simplify the discussion, we focus on selecting a single statistical parameter θ for a given, fixed candidate model for input into a computer simulation. A candidate model could be, for example, a Bernoulli distribution. We therefore drop the extra subscripts from Equation (1) in this subsection. The subscripts are needed in later sections.

Section 3 explores multiple candidate models for a given source of randomness.

For a Bayesian, the idea of exchangeability is preferred to the idea of independent and identically distributed (i.i.d.) random variables. Exchangeability is weaker than the i.i.d. assumption and plays a role in specifying probability models. Let $\mathbf{X}_N = (X_1, X_2, \dots, X_N)$ be a generic vector of random variables on an outcome space Ω. A probability p on Ω is *exchangeable* if it is invariant with respect to permutations of the coordinates (e.g., $p(\mathbf{x}_n) = p(x_1, x_2, \dots, x_n) = p(x_{h_1}, x_{h_2}, \dots, x_{h_n})$ for permutations h on $\{1, 2, \dots, n\}$ for arbitrary $n \leqslant N$).

Simulation is often concerned with conceptually infinite ($N \to \infty$) exchangeable sequences, e.g., no conceptual bound on the number of data observations or simulation replications. A key theorem (de Finetti, 1990; Bernardo and Smith, 1994) for infinite exchangeable sequences of Bernoulli random variables says that outcomes are conditionally independent, given the limiting fraction of heads, $\boldsymbol{\Theta} = \lim_{N\to\infty} \sum_{i=1}^{N} X_i/N$, with some mixture distribution $\pi(\theta)$,

$$\lim_{N\to\infty} p(\mathbf{x}_n) = \int \left\{ \prod_{i=1}^{n} f(x_i|\theta) \right\} \mathrm{d}\pi(\theta), \tag{2}$$

where $p(x_i|\theta) = f(x_i|\theta) = \theta^{x_i}(1-\theta)^{1-x_i}$ is viewed as a conditional probability when considered as a function of x_i and as a likelihood when written as a function of θ. A mixture written in the form of Equation (2) for an arbitrary parameter θ, distribution $\pi(\theta)$ and likelihood model f is called a *de Finetti-type representation*. The notation anticipates the convention of writing a *prior distribution* as $\pi(\cdot)$, representing the a priori belief that the parameter takes on a given value. Equation (2) is the basis for inference of statistical input parameter θ from data $\mathbf{x}_n = (x_1, \dots, x_n)$ via *Bayes' rule*,

$$p(\theta|\mathbf{x}_n) = \frac{\pi(\theta)p(\mathbf{x}_n|\theta)}{p(\mathbf{x}_n)} = \frac{\pi(\theta)\prod_{i=1}^{n} f(x_i|\theta)}{\int p(\mathbf{x}_n|\theta)\,\mathrm{d}\pi(\theta)}. \tag{3}$$

The first equality of Equation (3) is Bayes' rule and applies in general. The second equality follows from conditional independence. The *posterior probability* $p(\theta|\mathbf{x}_n)$ of θ given $\mathbf{x}_n$, summarizes uncertainty about θ via the likelihood model, the prior distribution and the data $\mathbf{x}_n$.

Bayesian methods require probability distributions to quantify initial uncertainty before data is observed. The *selection of a prior distribution* is controversial. Bruno de Finetti (1990) argues that a prior distribution is a subjective expression of uncertainty, and that You (yes, You) may justifiably specify a different distribution than I, since we may have different beliefs about the likelihood of a given event. Savage (1972) suggests a process for eliciting a prior distribution from a modeler through the evaluation of 'fair bets' (as opposed to limiting frequencies). Kahneman et al. (1982) illustrate potential pitfalls with eliciting probability judgments and present techniques to counter them. While this may seem 'too subjective' and open to biases (Edwards, 1984), the ability to include prior information provides important flexibility and can be considered an advantage of the approach. Frequentist methods apply only with data, and problems remain (e.g., see Section 3).

To avoid the impression of subjectivity, several 'automated' mechanisms have nonetheless been proposed to support the selection of a prior distribution. When a lot of data is available, the likelihood function is the dominant term in Bayes' rule, rather than the prior distribution, so these methods may be helpful. The first approach is to obtain a prior distribution for a parameter through an *indifference judgment*. For example, for the unknown probability θ of a Bernoulli outcome, this would give a uniform[0, 1] distribution, the prior probability model used by Laplace (1812) to assess his prior probability that the sun would come up tomorrow. That approach is coordinate dependent (e.g. indifference over θ or $\log\theta$).

Jeffreys (1946) suggested $\pi(\theta) \propto |H(\theta)|^{1/2}\,\mathrm{d}\theta$, where H is the expected information in one observation,

$$H(\theta) = \mathrm{E}\left[-\frac{\partial^2 \log p(X|\theta)}{\partial\theta^2}\bigg|_\theta\right], \tag{4}$$

because it has the attractive property of being invariant with respect to coordinate changes in θ. It is 'uniform' with respect to the natural metric induced by the likelihood function (Kass, 1989). *Jeffreys' prior* for Bernoulli sampling is a beta(1/2, 1/2) distribution. For some models, Jeffreys' prior is improper (does not integrate to one), but may be useful if the data results in a proper posterior after Bayes' rule is formally applied.

A third approach that is mathematically convenient is to assume a *conjugate prior* distribution, meaning that the posterior distribution has the same functional form as the prior distribution. For Bernoulli(θ) sampling, the beta(α, β) distribution with probability density function (p.d.f.) $f(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$ is a conjugate prior. If data $\mathbf{x}_n$ is observed, with $s_n = \sum_{i=1}^n x_i$, then the posterior p.d.f. is $f(\theta|\mathbf{x}_n) \propto \theta^{\alpha+s_n-1}(1-\theta)^{\beta+n-s_n-1}$, a beta($\alpha + s_n$, $\beta + n - s_n$) distribution. Conjugate prior distributions exist for all members of the regular exponential family (Bernardo and Smith, 1994), which includes the exponential, normal, gamma, log-normal, Wishart, Bernoulli, geometric and Poisson distributions, and linear regression models with normally distributed error, among others. The uniform[0, 1] distribution is in the conjugate family for Bernoulli sampling – it is a beta(1, 1) distribution. Prior distributions selected in this way are often selected to be as *noninformative* as possible, meaning that probability is spread 'evenly' over the space of parameters. Although 'evenly' is subjectively defined, heuristics are available for members of the regular exponential family, whose likelihood function can be written $p(x|\theta) = a(x)h_0(\theta)\exp[\sum_{j=1}^d c_j\phi_j(\theta)h_j(x)]$ for some $a(\cdot)$, $h_0(\cdot)$, c_j, $\phi_j(\cdot)$, $h_j(\cdot)$. The conjugate prior is $p(\theta) = [K(\mathbf{t})]^{-1}[h_0(\theta)]^{n_0}\exp[\sum_{j=1}^d c_j\phi_j(\theta)t_j]$, where $\mathbf{t} = (t_1, t_2, \ldots, t_d)$ is a hyperparameter. The posterior distribution given

n conditionally independent data points then has parameters $n_0 + n$ and the sum of $\mathbf{t}$ and the sufficient statistics (Bernardo and Smith, 1994). The parameter n_0 is therefore interpreted by some to be the 'strength' of the prior, measured in terms of the number of samples. In that case, evenly spreading probability can be taken to mean selecting n_0 close to 0, while insuring that the prior is still proper.

Jaynes (1983) suggests a fourth approach that is common in image and signal processing: maximum entropy methods define 'diffuse' priors with respect to a background measure, subject to moment constraints on the parameters. Berger (1994) and Kass and Wasserman (1996) comment further on default prior distributions and sensitivity analysis with respect to them.

Probability modeling is inherently subjective – even so-called 'objective' methods require the subjective specification of a likelihood model. One standard Bayesian practice is to use a slightly informative conjugate distribution for the unknown mean, by choosing it to be proper but diffuse (Gilks et al., 1996). For example, the conjugate prior for an unknown mean of a normal distribution is also a normal distribution. A diffuse prior would be Normal(0, σ^2_{big}) for some large σ^2_{big}. Conjugate prior distributions are mathematically convenient, but care is still required with their use, as with any statistical analysis, Bayesian or otherwise.

Classical asymptotic theorems (laws of large numbers, LLN; central limit theorems, CLT; e.g., Billingsley, 1986) have Bayesian interpretations when considered to be conditional on the mean and standard deviation of an infinite exchangeable sequence. A Bayesian extension of the LLN allows for a sample average to converge to an 'unknown' mean (random variable) rather than to a 'true' mean.

Theorem 1 (Bayesian LLN). *Let X_i be an exchangeable sequence of random variables, and let $\overline{X}_n$ and $\overline{Y}_m$ be the averages of n and m of the X_i, respectively. If* Var$[X_1] < \infty$*, then the* probability *that*

$$|\overline{X}_n - \overline{Y}_m| > \varepsilon$$

may be made arbitrarily small by taking n and m sufficiently large (de Finetti, 1990, p. 216).

Although the mode of a posterior distribution may not be the true mean, an asymptotic normality property holds for posterior distributions of parameters.

Theorem 2 (Posterior normality). *For each n, let $p_n(\cdot)$ be the posterior p.d.f. of the d-dimensional parameter θ_n given $\mathbf{x}_n = (x_1, \ldots, x_n)$, let $\tilde{\theta}_n$ be its mode, and define the $d \times d$ Bayesian observed information matrix Σ_n^{-1} by*

$$\Sigma_n^{-1} = -L_n''(\tilde{\theta}_n), \quad \textit{where } L_n''(\vartheta) = \left.\frac{\partial^2 \log p_n(\theta|\mathbf{x}_n)}{\partial\theta^2}\right|_{\theta=\vartheta}. \tag{5}$$

Then $\phi_n = \Sigma_n^{-1/2}(\theta_n - \tilde{\theta}_n)$ *converges in distribution to a standard (multivariate) normal random variable, if* 3 *technical conditions hold*: (c1) Steepness: $\lim_{n\to\infty} \bar{\sigma}_n^2 \to 0$, *where* $\bar{\sigma}_n^2$ *is the largest eigenvalue of* Σ_n, (c2) Smoothness: *Let* $B_\delta(\tilde{\theta}_n) = \{\vartheta\colon |\vartheta - \tilde{\theta}_n| < \delta\}$. *For any* $\varepsilon > 0$, *there exists* N *and* $\delta > 0$ *such that, for any* $n > N$ *and* $\vartheta \in B_\delta(\tilde{\theta}_n)$, *the derivatives in Equation* (5) *exist and satisfy* $I - A(\varepsilon) \leqslant L_n''(\vartheta)\{L_n''(\tilde{\theta}_n)\}^{-1} \leqslant I + A(\varepsilon)$, *where* I *is a* $d \times d$ *identity matrix and* $A(\varepsilon)$ *is a* $d \times d$ *symmetric positive semidefinite matrix whose largest eigenvalue tends to* 0 *as* $\varepsilon \to 0$, (c3) Concentration: *For any* δ, $\int_{B_\delta(\tilde{\theta}_n)} p_n(\theta)\,\mathrm{d}\theta \to 1$ *as* $n \to \infty$. (Bernardo and Smith, 1994, Proposition 5.14.)

Theorem 2 asserts that uncertainty about the value of the unknown parameter value can be approximated asymptotically with a normal distribution. The Bayesian observed information Σ_n^{-1} is a measure of precision of the posterior distribution of θ, and behaves asymptotically like the frequentist observed information (which ignores the prior distribution) under rather general conditions, but the interpretation differs somewhat. The classical analog of Theorem 2 asserts that the MLE is asymptotically normally distributed about a 'true' parameter θ_0 (Law and Kelton, 2000), rather than describing uncertainty about θ. The mode $\tilde{\theta}_n$ is often called a MAP (maximum a posteriori probability) estimator. Conditions (c1) and (c2) basically insure that the posterior mode is asymptotically shaped like a normal distribution, and (c3) insures that probability outside a neighborhood of $\tilde{\theta}_n$ is negligible. Bernardo and Smith (1994) also discuss alternate conditions.

The above results apply to conceptually infinite exchangeable sequences that can be used to infer statistical parameters, $\boldsymbol{\theta}_p$. Environmental parameters $\boldsymbol{\theta}_e$ do not have such sequences to help inference, but the subjective probability methods of de Finetti (1990), Savage (1972), Kahneman et al. (1982) still apply for assessing prior distributions. Exchangeability is relevant if a finite exchangeable sequence exists to help inference for $\boldsymbol{\theta}_e$.

Here are facts that link finite and infinite exchangeable sequences that are not used further in this paper, but are useful for further subjective probability work. One, exchangeability is weaker than even conditional independence for finite sequences. For example, let $\Omega = \{0,1\}^N$ model $N = 3$ Bernoulli outcomes, and let $\theta_N = \sum_{i=1} X_i/N$. The subjective probability assessment $p((1,0,0)) = p((0,1,0)) = p((0,0,1)) = 1/3$ is an exchangeable Bernoulli model, but not independent, because $X_1 + X_2 + X_3 = 1$. Similarly, X_1 and X_2 are not conditionally independent, given $\theta_N = 1/3$. (This model is an atypical subjective assessment for coin flips, but matches well the 'hide a coin under a shell' game.) Two, suppose that each of the finite set of alternatives $\theta_N \in \{0/N, 1/N, \ldots, (N-1)/N, 1\}$ is judged equally likely for each N, then $\lim_{N\to\infty} p(\theta_N)$ converges in distribution to Laplace's (1812) continuous uniform[0, 1] prior distribution for $\theta = \lim_{N\to\infty} \theta_N$. Three, de Finetti (1990) derives Equation (2) as a consequence of having a conceptually infinite exchangeable sequence of Bernoulli outcomes, as opposed to directly assuming conditional independence. Four, judgments stronger than exchangeability, such as invariance to sums or to an ℓ_p-norm, may be required to justify de Finetti-type representations for other random variables (Barlow and Mendel, 1992; Chick and Mendel, 1998).

1.2 Loss and value of information

The fact that input uncertainty is described by probability distributions allows the modeler to (1) assess the *expected value of information* (EVI) of additional data collection and (2) to perform an uncertainty analysis. The EVI is useful in experimental design. It measures the value of resolving uncertainty with respect to a *loss function* $\mathcal{L}(d, \omega)$ that describes the loss when a *decision* d is chosen and the state of nature is ω. Experiments can bring information about ω, so the expected improvement in the loss given by the experiment is a Bayesian experimental design criterion.

The value of information idea directly leads to the selection procedures in Section 5. A simplified version of that problem adapted from de Groot (1970, Sections 11.8–11.9) illustrates the key concepts. Suppose we must decide whether or not the unknown mean W of a normal distribution is smaller (decision $d = 1$) or larger ($d = 2$) than w_0. Assume the variance σ^2 is known. Conditionally independent samples $\mathbf{X}_n = (X_1, X_2, \ldots, X_n)$, with $p(X_i) \sim \mathrm{Normal}(w, \sigma^2)$ given $W = w$, can be used to infer the value of the mean. The decision maker designs a sampling experiment (chooses n) to balance the cost of sampling, cn, and the expected penalty if the wrong answer is chosen. Here the penalty for incorrect selection is the opportunity cost $\mathcal{L}(d, w)$, the difference between the actual value of w and w_0 when the wrong answer is selected, and 0 if the right answer is selected. Hence,

$$\mathcal{L}(1, w) = \begin{cases} 0 & \text{if } w \leqslant w_0, \\ w - w_0 & \text{if } w > w_0, \end{cases} \quad \text{and}$$

$$\mathcal{L}(2, w) = \begin{cases} w_0 - w & \text{if } w \leqslant w_0, \\ 0 & \text{if } w > w_0. \end{cases}$$

Since the mean is not known exactly, there is a potential penalty for incorrectly specifying whether W is smaller or larger than w_0. We model uncertainty about W with a $\mathrm{Normal}(\mu, 1/\tau)$ prior distribution, which is conjugate for normal sampling with an unknown mean and known variance (de Groot, 1970). Here τ is the *precision* in our uncertainty about W. Observing $\mathbf{X}_n = \mathbf{x}_n$ would reduce the uncertainty and result in the posterior distribution

$$p(w|\mathbf{x}_n) \sim \mathrm{Normal}\bigl(z, \tau_n^{-1}\bigr),$$

where

$$z = \text{posterior mean of } W = \mathrm{E}[W|\mathbf{x}_n] = \frac{\tau\mu + n\bar{x}_n/\sigma^2}{\tau + n/\sigma^2}$$

and

$$\tau_n = \text{posterior precision of } W = \tau + n/\sigma^2.$$

The variance τ_n^{-1} equals the posterior variance approximation Σ_n in Equation (5) because Σ_n is based on a normal distribution approximation.

The posterior mean z influences the decision, but it depends upon n, which must be selected before $\mathbf{X}_n$ is observed. We therefore need the *predictive distribution* $p(z)$ of the posterior mean $Z = \mathrm{E}[W|\mathbf{X}_n] = (\tau\mu + n\overline{X}_n/\sigma^2)/\tau_n$ to see how n samples influence the decision d. The conditional distribution of $\overline{X}_n$ given w is Normal$(w, \sigma^2/n)$. Mixing over the prior distribution of W implies that the predictive distribution for Z is Normal(μ, τ_z^{-1}), where

$$\tau_z = \tau\frac{\tau + n/\sigma^2}{n/\sigma^2}. \tag{6}$$

The variance τ_z^{-1} of Z is 0 as $n \to 0$ (no new information). If $n \to \infty$ (perfect information about w), then $\mathrm{Var}[Z] \to \tau^{-1}$, the prior variance of W.

The experimental design that minimizes risk (the cost of sampling plus expected losses due to a potentially incorrect decision) is the n that minimizes a nested expectation, the inner expectation corresponding to the expected loss after $\mathbf{X}_n = \mathbf{x}_n$ is observed, the outer expectation averaging over $\mathbf{X}_n$.

$$\rho(n) = cn + \mathrm{E}\big[\mathrm{E}\big[\mathcal{L}\big(d(\mathbf{X}_n), W\big)|\mathbf{X}_n\big]\big]. \tag{7}$$

A general technique for determining $\mathrm{E}[\mathcal{L}(d(\mathbf{X}_n), W)|\mathbf{X}_n]$ is to obtain an auxiliary loss function $\mathcal{L}^*$ that has the same optimal decision, but simplifies the loss function by making the loss of one of the decisions equal to 0. Adding a function of w does not change the optimal decision (de Groot, 1970). Set $\mathcal{L}^*(d, w) = \mathcal{L}(d, w) - \mathcal{L}^*(1, w)$, which is 0 if $d = 1$ and is $w_0 - w$ if $d = 2$. Then

$$\mathrm{E}\big[\mathcal{L}^*\big(d(\mathbf{X}_n), W\big)|\mathbf{X}_n\big] = \begin{cases} 0 & \text{if } d(\mathbf{X}_n) = 1, \\ w_0 - Z & \text{if } d(\mathbf{X}_n) = 2. \end{cases} \tag{8}$$

The decision that minimizes the loss in Equation (8) is to assert $d(\mathbf{X}_n) = 2$ ('bigger') if the posterior mean exceeds the threshold, $Z > w_0$, and to assert $d(\mathbf{X}_n) = 1$ ('smaller') if $Z \leqslant w_0$.

The expectation over the outcomes of this experiment can be determined with well-known tables because the decision depends upon $\mathbf{X}_n$ only through Z, and Z has a normal distribution. Define $\phi(\cdot)$ and $\Phi(\cdot)$ to be the p.d.f. and c.d.f. of a standard normal random variable, respectively. The expected loss can be determined from the standard normal loss function $\Psi[s] = \int_s^\infty (t-s)\phi(t)\,\mathrm{d}t = \phi(s) - s(1-\Phi(s))$ for expected lost sales in the newsvendor problem if demand is normally distributed (e.g., Nahmias, 2000, p. 262, standardized loss).

$$\begin{aligned}\mathrm{E}\big[\mathrm{E}\big[\mathcal{L}^*\big(d(\mathbf{X}_n), W\big)|\mathbf{X}_n\big]\big] &= -\int_{w_0}^{\infty} (z - w_0)p(z|\mathbf{X}_n)\,\mathrm{d}z \\ &= -\tau_z^{-1/2}\Psi\big[\tau_z^{1/2}(w_0 - \mu)\big].\end{aligned}$$

The expected loss of the original loss function is recovered by adding back $\mathrm{E}[\mathcal{L}(1, W)]$, using the prior distribution of W for the expectation.

$$\begin{aligned}&\mathrm{E}\bigl[\mathrm{E}\bigl[\mathcal{L}\bigl(d(\mathbf{X}_n), W\bigr)|\mathbf{X}_n\bigr]\bigr]\\&\quad = \tau^{-1/2}\Psi\bigl[\tau^{1/2}(w_0 - \mu)\bigr] - \tau_z^{-1/2}\Psi\bigl[\tau_z^{1/2}(w_0 - \mu)\bigr].\end{aligned} \tag{9}$$

The EVI for m samples is the difference between Equation (9) when $n \to 0$ and when $n = m$ (τ_z depends on n). If $w_0 > \mu$, the EVI simplifies to $\tau_z^{-1/2}\Psi[\tau_z^{1/2}(w_0 - \mu)]$. Combine Equation (9) with Equations (6) and (7), note that $\mathrm{d}\Psi/\mathrm{d}s = \Phi(s) - 1$ and $\mathrm{d}\tau_z/\mathrm{d}n = -\tau^2\sigma^2/n^2$, and take the derivative with respect to n (relaxing the integer assumption) to obtain an optimality condition for the sample size.

$$\frac{\partial\rho}{\partial n} = \frac{1}{2}\tau_z^{-3/2}\phi\bigl[\tau_z^{1/2}(w_0 - \mu)\bigr]\frac{-\tau^2\sigma^2}{n^2} + c = 0.$$

For diminishing costs $c \to 0$, the sample size is large. Since $\tau_z \to \tau$ as $n \to \infty$, the optimal sample size n is approximately

$$n^* = \left(\frac{\tau^{1/2}\sigma^2\phi[\tau^{1/2}(w_0 - \mu)]}{2c}\right)^{1/2}. \tag{10}$$

This argument illustrates the basic ideas of loss functions, and the use of predictive distributions for future samples to infer the EVI of sampling. *The technique of adding functions of the unknowns can be useful to simplify the derivation of the optimal solution. Asymptotic approximations are a further tool to identify criteria-based sampling plans.* Extensions of this basic argument justify the value of information based selection procedures of Section 5.1 and Chick and Inoue (2001a, 2001b, 2002).

An alternate mechanism to approximate the effect of information on parameter uncertainty is based on a thought experiment for the posterior probabilities of parameters. For members of the regular exponential family, the asymptotic variance approximation Σ_n in Equation (5) simplifies to the form $H^{-1}(\theta)/(n_0 + n)$, where H is the expected information from one observation (Equation (4)), when a canonical conjugate prior distribution is used (Bernardo and Smith, 1994). To approximate the effect of collecting m additional samples on the parameter uncertainty, one could presume that the posterior distribution changes from $\mathrm{Normal}(\tilde{\theta}_n, \Sigma_n)$ to

$$\mathrm{Normal}\left(\tilde{\theta}_n, \Sigma_n\frac{n_0 + n}{n_0 + n + m}\right). \tag{11}$$

This transformation reflects an appropriate scaling of the posterior precision, and the idea is used in a frequentist context for estimating how many replications are required to achieve a confidence interval of a given size (Law and Kelton, 2000). Chen (1996) uses this type of approximation for the Bayesian posterior distribution of the unknown means of several simulated systems in order to motivate a class of ranking and selection procedures called the OCBA (optimal computing budget allocation). Ng and Chick (2001, 2006) use the idea to plan experiments to reduce input uncertainty in a way that reduces output uncertainty.

1.3 Uncertainty analysis

The fact that uncertainty about inputs is described with probability distributions allows a modeler to engage in *uncertainty analysis*, in addition to sensitivity analysis. A sensitivity analysis tests how the mean simulation output depends upon one or more input parameters as that parameter is varied (estimating $\mathrm{E}[g(\theta)|\mathcal{E}]$ as a function of θ, given all information $\mathcal{E}$). Uncertainty analysis entails propagating input parameter uncertainty about Θ through to uncertainty about outputs Y. Even if a simulation has no random number stream, a distribution on unknown inputs means that the output is random.

An unbiased estimator of the mean output $\mathrm{E}[Y|\mathcal{E}]$ with both stochastic (from $\mathbf{u}$) and systemic (or parameter) uncertainty accounted for is obtained from the Bayesian model average (BMA) in Figure 2, which averages over random inputs sampling according to the distribution $p(\theta|\mathcal{E})$ (Draper, 1995; Chick, 2001). Zouaoui and Wilson (2003) explore the relative magnitude of stochastic and systemic uncertainty with variations on the BMA, and discuss how to update the estimate should new data become available (so the algorithm need not be rerun from scratch). Importance sampling (cf. Chapter 11) techniques can re-weight estimates accordingly (with likelihood ratio determined as the ratio of the 'new' posterior divided by the 'old' distribution). Andradóttir and Glynn (2004) examine the estimation of $\mathrm{E}[Y|\mathcal{E}]$ when there may be bias in the estimates of Y given θ, when quasi-random sequences are used in place of the pseudo-random sequences assumed by Figure 2, or when numerical techniques like Simpson's rule are employed to select values of θ. Another goal is to estimate the distribution of the conditional expectation $\mathrm{E}[Y|\Theta, \mathcal{E}]$. When Y is a deterministic function of Θ, then naive Monte Carlo simulation can be used with traditional kernel estimation techniques to assess the distribution of $Y(\Theta)$. When the simulation is stochastic (depends on $\mathbf{u}$), then $\mathrm{E}[Y|\theta, \mathcal{E}]$ is imperfectly estimated for any given θ. Given several technical conditions (e.g., univariate continuous-valued θ, monotonic mean re-

```
for $r = 1, \ldots, R$ replications
    sample parameter $\theta_r$ from $p(\theta|\mathcal{E})$
    for $i = 1, 2, \ldots, n$
        generate simulation output $y_{ri}$ given input $\theta_r$
    end loop
end loop
Estimate $\mathrm{E}[Y|\mathcal{E}]$ with $\bar{y} = \sum_{r=1}^{R} \frac{1}{R} \sum_{i=1}^{n} y_{ri}/n$.
```

Fig. 2. Bayesian model average (BMA).

sponse), Steckley and Henderson (2003) derive asymptotically optimal ways of selecting by cleverly selecting r and n in Figure 2 to produce a kernel density estimator based on the output. Their work builds upon Lee and Glynn (1999), which estimated the distribution function of $\mathrm{E}[Y|\Theta, \mathcal{E}]$ for discrete θ.

2 Computational issues

Three basic computational issues for implementing a Bayesian analysis are maximization (e.g., find the MLE $\hat{\theta}$, or MAP $\tilde{\theta}$ estimators for a posterior distribution); integration, either to find a marginal distribution (e.g., find $p(\theta_1|\mathbf{x}_n)$ from $p(\theta_1, \theta_2|\mathbf{x}_n)$) or constant of proportionality for a posterior distribution (e.g., find $c^{-1} = \int f(\mathbf{x}_n|\theta)\,\mathrm{d}\pi(\theta)$); and simulation (e.g., sample from $p(\theta|\mathbf{x}_n)$ in order to estimate $\mathrm{E}[g(\theta)|\mathbf{x}_n]$). Techniques to address these issues are described in a variety of sources (e.g., Naylor and Smith, 1982; Evans and Swartz, 1995; Tanner, 1996; Gilks et al., 1996; The Mathworks, 2002).

For maximization, a number of methods are available including gradient-based methods (e.g., Newton–Raphson), gradient-free methods (e.g., Nelder–Mead), and simulation-based methods. The expectation–maximization (EM) algorithm is a technique for finding the MAP or MLE when there is missing data or nuisance parameters are to be integrated out (e.g., the MAP of $p(\theta_1|\mathbf{x}_n)$ when it is 'messy' but $p(\theta_1, \theta_2|\mathbf{x}_n)$ is easier to manipulate).

For integration, five general techniques apply (Evans and Swartz, 1995) when analytical results (e.g., conjugate priors) are not available: quadrature, asymptotic methods, Markov chain methods, importance sampling, adaptive importance sampling. Quadrature is useful when the number of dimensions is not too large. The Laplace method is an interesting asymptotic approximation for integrals $\int g(\theta)f(\theta|\mathbf{x}_n)\,\mathrm{d}\pi(\theta)$. The Laplace method applies even if $f(\theta|\mathbf{x}_n)$ is only a likelihood when the constant of proportionality for the posterior is unknown, and can work well for integrating out nuisance parameters if regularity conditions hold. The method is based on asymptotic normality approximations like those used for Equation (5), and therefore require a large n. Another effective technique for approximating the density $p(\theta_1|\mathbf{x}_n)$ (not just the MLE or MAP) when it is 'messy' but $p(\theta_1|\theta_2, \mathbf{x}_n)$ and $p(\theta_2|\theta_1, \mathbf{x}_n)$ are easy to manipulate is data augmentation, often called the IP algorithm (for imputation, posterior algorithm). The IP algorithm is a nice alternative to other kernel estimation methods, and is closely related to the Markov chain Monte Carlo (MCMC) methods mentioned further. *Importance sampling (IS) remains one of the more powerful methods for efficient integration.* See Chapter 11.

For simulation of variates, classical methods for generating independent variates from posterior distributions may apply (see Chapter 4). *Posterior distributions are often known only up to a constant of proportionality* (*the numerator of Bayes' rule is easy to write, but the denominator may be hard to compute*). *It is therefore important to have a method to simulate variates for arbitrary functions proportional to posterior distributions. MCMC is the most important of those* *methods at present*. MCMC constructs a Markov chain whose stationary distribution is the desired posterior distribution (Chapter 4, this volume; Gilks et al., 1996). The ARMS (adaptive rejection Metropolis sampler) combines adaptive rejection sampling, which is useful for log-concave posterior distributions, together with an MCMC-type Metropolis step to handle nonlog-concave distributions (Gilks et al., 1995). States of a chain constructed with MCMC techniques can be sampled for input into the BMA of Figure 2. Samples and estimators based on MCMC need evaluation to assure reasonable convergence for estimators and faithfulness to the posterior distribution.

Figure 3 illustrates a qualitative feel for some of the approximation techniques for a gene linkage model (Tanner, 1996) that has a parameter $\theta \in [0, 1]$. A spreadsheet implementation of the EM algorithm identified the MAP $\tilde{\theta}$. The asymptotic normal approximation of Theorem 2 provides a reasonable estimate of the mode and variance of the true posterior distribution, but does not model skewness well, particularly if $\tilde{\theta}$ is near the boundary or if few data points are available. Data augmentation results in a smoother kernel estimator than the empirical histogram estimator usually studied with MCMC methods. The MCMC estimator could be smoothed like the data augmentation to provide a much more accurate representation (Gilks et al., 1996) with an additional computational cost for smoothing.

For simple examples a spreadsheet is fine, but more powerful tools are needed in general to implement Bayesian inference. The BUGS and WinBUGS packages implement Gibbs sampling and some Metropolis sam-

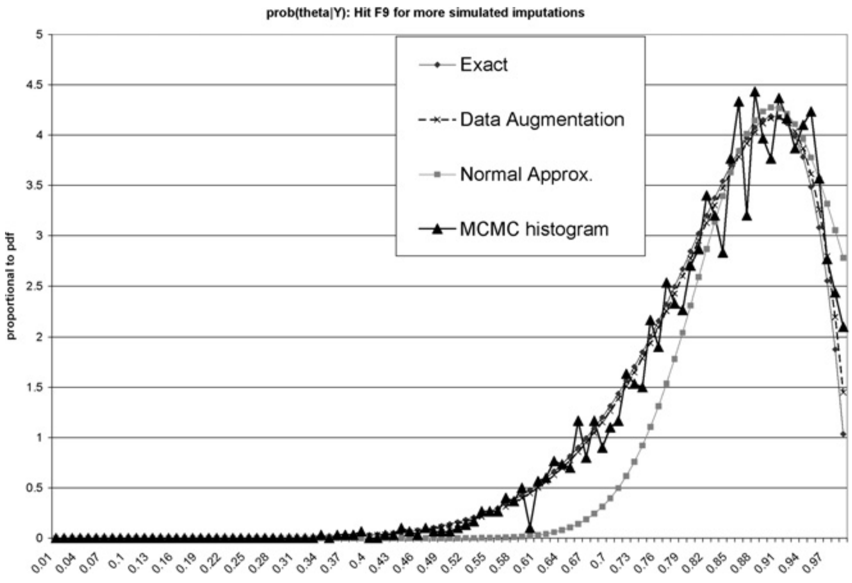


Fig. 3. Different approximations for the posterior p.d.f.

pling, and are available on the WWW (Spiegelhalter et al., 1996). BOA, for Bayesian output analysis (Smith, 2004), is a set of MCMC diagnostic tools for convergence and data analysis that functions with the R or S-PLUS statistical packages. Gauss and Matlab are also commonly used to program MCMC methods.

At present, it is possible to input randomized input parameters to some commercial discrete-event simulation packages to implement the BMA algorithm of Figure 2, but interfaces are not yet fully user friendly. A user-friendly tool to implement the BMA and other uncertainty analysis needs in commercial discrete-event simulation packages would be helpful. Uncertainty analysis for other Monte Carlo applications has been available as a spreadsheet tool for some time (e.g., Winston, 2000).

3 Input distribution and model selection

Selecting an input distribution to model a sequence of random quantities $X_1, X_2, \ldots$ is often more complicated than inferring a parameter of a single parametric distribution, as described in Section 1. There is often a finite number q of candidate distributions proposed to model a given source of randomness, with continuous parameters $\theta_m = (\vartheta_{m1}, \ldots, \vartheta_{md_m})$, where d_m is the dimension of θ_m, for $m = 1, \ldots, q$. For example, service times might be modeled by exponential, log-normal or gamma distributions ($q = 3$). Denote by $p(x|m, \theta_m)$ the probability density function (p.d.f.) for X, given m and θ_m.

The classical approach in simulation (Law and Kelton, 2000) for input selection is to (a) find the MLE of each parameter, (b) perform a goodness-of-fit test, and (c) pick the 'best' fitting candidate distribution and input the MLE into the simulation. The Bayesian approach addresses some controversial aspects of the classical approach. Critiques of classical techniques in general include: goodness-of-fit and P-value criteria are difficult to interpret and inconclusive at best, and misleading at worst (Berger and Pericchi, 1996); use of a single distribution and parameter underestimates the uncertainty in the distribution's functional form and parameter (Draper, 1995); with few data points, few candidate distributions are rejected, and with many data points, all distributions are rejected (Raftery, 1995); there is no coherent method for selecting among nonrejected distributions; and classical techniques can reject the a posteriori most probable distribution (Lindley, 1957; Berger and Delampady, 1987). *In the simulation context, input uncertainty can make standard confidence intervals for the mean output almost meaningless if the classical approach is employed* (Chick, 2001; Barton and Schruben, 2001).

A Bayesian approach is no different than the approach in Section 1, except that a prior probability distribution needs to be placed on the model/parameter combination, $\pi(M = m, \theta_m)$, a mixed discrete-continuous model. If data $\mathbf{x}_n$ becomes available, the BMA then requires sampling from the joint posterior $p(m, \theta_m|\mathbf{x}_n)$. This can be accomplished by composition, sampling the input model then the parameter with $p(m|\mathbf{x}_n)p(\theta_m|m,\mathbf{x}_n)$. *While improper, noninformative prior distributions can formally be used when there is a single candidate model and enough data makes the posterior proper, this cannot be done when there are multiple candidate distributions.* The reason is that

$$p(M=m|\mathbf{x}_n)=\frac{\int p(\mathbf{x}_n|\theta_m,m)\pi(\theta_m|m)\pi(M=m)\,\mathrm{d}\theta_m}{\sum_{i=1}^{q}\int p(\mathbf{x}_n|\theta_i,i)\pi(\theta_i|i)\pi(M=i)\,\mathrm{d}\theta_i}$$

can be changed at will by changing an improper prior $\pi(\theta_m|M)$ to an equally valid improper prior $c\pi(\theta_m|M)$ for an arbitrary constant c. Proper (and therefore informative) prior distributions $\pi(\theta_m|M=m)$ are needed for each m in order to make Bayes' rule well defined. Chick (2001) implements the BMA with this model selection for discrete-event simulations, and suggests a method for assessing $\pi(\theta_m|M=m)$ based on moment methods. O'Hagan (1995), Berger and Pericchi (1996) proposed automated prior distribution selection techniques for the model selection problem that use variations on the theme of using part of the data with a noninformative prior.

Chick (2001) illustrated Bayesian input modeling in a stochastic simulation context when $q>1$, and suggested a method-of-moments approach for assessing prior distributions for the unknown parameters of each candidate distribution. Zouaoui and Wilson (2004) noted a decoupling of stochastic uncertainty from two types of structural uncertainty (about candidate models and their parameters) under special conditions, and provided a variance reduction for the BMA and numerical analysis. Richardson and Green (1997) and Cheng and Currie (2003) present techniques for the nonregular case when a candidate itself is a mixture distribution (e.g., a mixture of 2 or 3 normal distributions).

Selecting models according to $p(M|\mathcal{E})$ is consistent in that if one of the entertained models is actually the true model, then the true model is selected if enough data is observed and some regularity conditions hold. When the true model is not among those being considered, Bayesian model selection chooses the model that is closest to the true model in terms of Kullback–Leibler divergence (Berk, 1966; Bernardo and Smith, 1994; Dmochowski, 1999).

4 Joint input–output models

Simulation is interested in both stochastic uncertainty, or randomness that occurs when all model parameters are known, and structural uncertainty, or uncertainty about model inputs when a real system is being simulated. This section describes an input–output model that quantifies the uncertainty in simulation outputs due to input uncertainty, data collection plans for reducing input uncertainty in a way that effectively reduces output uncertainty, mechanisms to select computer inputs to improve estimates of the system response, and mechanisms to help infer probability distributions for input parameters, given information about output parameters (the inverse problem).

Recall Figure 1. It is impossible to simulate all possible values of continuous input parameters in finite time on a finite number of computers. Metamodels describe the value of g at untested input values (see Chapter 18). This is useful when the simulation model requires extensive computation. Metamodel parameters $\boldsymbol{\psi} = (\phi_1, \phi_2, \dots, \phi_{n_m})$ may include regression coefficients, or parameters of a Gaussian random function (GRF) model of the mean response (Cressie, 1993; Santner et al., 2003; van Beers and Kleijnen, 2003). Since the metamodel parameters are unknown, they are described as a random variable $\boldsymbol{\Psi}$. The metamodel is

$$Y = g(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e, \boldsymbol{\theta}_c; \mathbf{U}, \boldsymbol{\Psi}). \tag{12}$$

This formulation allows for Y to be predicted for unsimulated values of $\boldsymbol{\theta} = (\boldsymbol{\theta}_p, \boldsymbol{\theta}_e, \boldsymbol{\theta}_c)$ via the response model and $\boldsymbol{\Psi}$. Field data from a modeled system can be used (1) to infer the input parameters $\boldsymbol{\theta}_c$, along the lines of Sections 1 and 3, or (2) to understand the distribution of the outputs Y. Output from multiple simulation runs are used to infer $\boldsymbol{\Psi}$. This formulation generalizes several previous models that focused on $g(\boldsymbol{\theta}_p, \mathbf{u})$ (Cheng and Holland, 1997), or nonstochastic models that do not model randomness from $\mathbf{u}$ (Santner et al., 2003). The model of Chick (1997) differed in that simulation output was not considered to be exchangeable with a real system's output (a calibration issue).

4.1 Bayesian metamodeling

Here we discuss inference for the normal linear model and Gaussian random function (GRF) metamodels. The normal linear model is

$$Y = \sum_{\ell=1}^{p} g_\ell(\boldsymbol{\theta})\beta_\ell + Z(\boldsymbol{\theta}; \mathbf{U}) = \mathbf{g}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{\beta} + Z(\boldsymbol{\theta}; \mathbf{U})$$

for *known* regression functions $g_1, \dots, g_p$, unknown regression coefficients $\boldsymbol{\beta}$, and independent zero-mean random noise $Z(\cdot)$ with sampling variance σ^2. If σ^2 does not depend upon $\boldsymbol{\theta}$, then the conjugate prior is an inverse gamma distribution for σ^2 and a conditionally normal distribution for $\boldsymbol{\beta}$ given σ^2, if all factors are active (Bernardo and Smith, 1994). George and McCulloch (1996) and Cheng (1999) discuss techniques for estimating which factors are active with what probability. Ng and Chick (2004) describe an entropy-based experimental design criterion to identify which factors are active and reduce parameter uncertainty simultaneously.

If the g_i represent the individual dimensions of the unknown parameters $(\boldsymbol{\theta}_p, \boldsymbol{\theta}_e)$, the β_ℓ are gradients with respect to the inputs. If further the model has only statistical input parameters $\boldsymbol{\theta}_p$ for which data can be collected (but not parameters $\boldsymbol{\theta}_e$ for which no data is available), Ng and Chick (2001) and

Zouaoui and Wilson (2003) indicated that output uncertainty can be decoupled asymptotically or under special conditions.

$$V_{tot} = \mathrm{Var}[\overline{Y}|\mathcal{E}] \approx \text{stochastic} + \text{structural uncertainty}$$
$$\approx \frac{\hat{\sigma}_0^2}{m} + \frac{\boldsymbol{\beta}^{\mathrm{T}} \widehat{H}_{\boldsymbol{\theta}_p}^{-1} \boldsymbol{\beta}}{n},$$

where $\hat{\sigma}_0^2$ is the estimate of the variance from m replications, the MLE $\hat{\boldsymbol{\theta}}_p$ and estimate $\widehat{H}_{\boldsymbol{\theta}_p}^{-1}$ of the information in one observation are based on n data points, and technical conditions hold (e.g., those for Theorem 2). This adapted a frequentist result of Cheng and Holland (1997). Ng and Chick (2001) applied the result to uncertainty due to multiple input parameters, to provide sampling plans to collect further data (e.g., for arrival rates or for service time distributions) to reduce input parameter uncertainty in a way that optimally reduces output uncertainty in some sense. Ng and Chick (2006) extended that analysis by accounting for uncertainty in $\boldsymbol{\beta}$ as well, providing analytical results for members of the exponential family of distributions, and giving a numerical analysis.

Figure 4 summarizes qualitatively how ignoring input parameter uncertainty can significantly degrade confidence interval (CI) coverage. The experiment involved simulating the parameter estimation process for several simulation input parameters, inputting those parameters into a simulation, then generating a nominal 95% CI (see Ng and Chick, 2006, for details). The values at $B = 0$ in all graphs represent the CI coverage and mean half width if no additional data is collected after that first round of simulation. The top row of graphs gives the empirical coverage and half width if the CI is based on the estimated variance V_{tot} involving both stochastic and structural uncertainty. The bottom row is based on computing the CI by inputting only the MLE of the input parameters into the simulation, and using only stochastic uncertainty $V_{stoch} = \hat{\sigma}_0^2/m$ to estimate the variance of the estimator. The values at $B > 0$ describe how the coverage would change if additional data for the several different input parameters were collected in an optimal way. Optimal here is defined by collecting data for the different parameters in a way that minimizes V_{tot} if the effect of additional samples is presumed to reduce V_{tot} as in Equation (11). There is a slight degradation in coverage, perhaps due to the fact that the simulation model was nonlinear and the approximation for V_{tot} is based on a local linear approximation. The bigger story from this example is that ignoring input parameter uncertainty can lead to almost meaningless CI statements if the mean of the simulated response is a nonlinear function of the input parameters. The effect is worse with fewer data points to estimate parameters. To date, much simulation research seeks analytical results for stochastic models, or mechanisms to reduce the variance of estimators due to stochastic noise. Those results need to be complemented with an *understanding of how performance depends on input uncertainty, and methods to reduce input uncertainty to effectively reduce output uncertainty*. Bayesian tools can help.

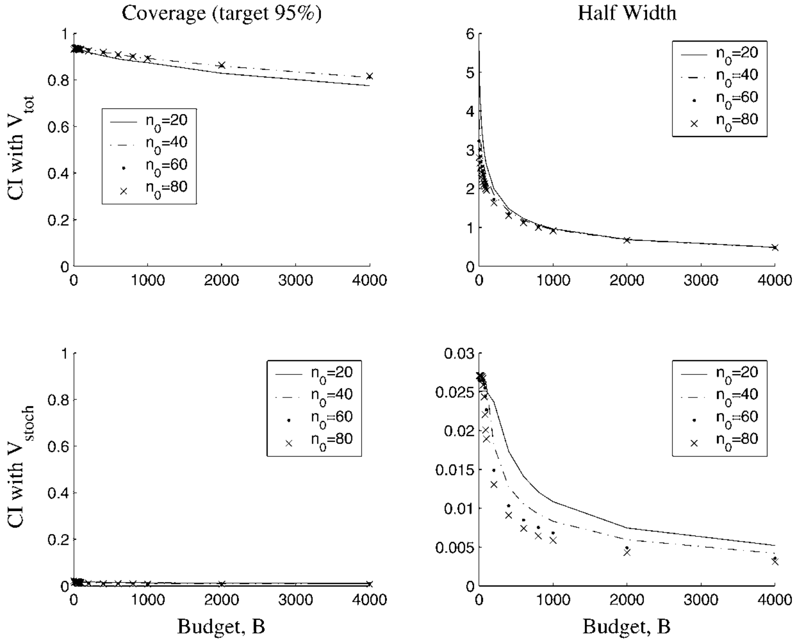


Fig. 4. Empirical coverage for a nominal 95% CI is poor if parameter uncertainty is ignored.

GRFs are well-known response models in deterministic simulations, particularly in geostatistics (Cressie, 1993; Santner et al., 2003), but are less well known in stochastic simulation. GRFs provide flexibility that the linear model does not, and are useful when g takes a long time to compute. The GRF for an unknown nonstochastic g (no random number stream $\mathbf{u}$) is

$$Y(\boldsymbol{\theta}) = \sum_{\ell=1}^{p} g_\ell(\boldsymbol{\theta})\beta_\ell + Z(\boldsymbol{\theta}) = \mathbf{g}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{\beta} + Z(\boldsymbol{\theta}) \tag{13}$$

for known regression functions $g_1, \ldots, g_p$ of $\mathbb{R}^d$, and unknown regression coefficients $\boldsymbol{\beta}$. The zero-mean random second-order process $Z(\boldsymbol{\theta})$ is such that for any distinct inputs $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m$, the vector $(Y_1, \ldots, Y_m)$ has a multivariate normal distribution, conditional on $\boldsymbol{\beta}$. GRFs are determined by their mean $\mathbf{g}^{\mathrm{T}}(\boldsymbol{\theta})\boldsymbol{\beta}$ and (auto)covariance function $C^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \mathrm{Cov}(Y(\boldsymbol{\theta}_1), Y(\boldsymbol{\theta}_2))$, defined independent of $\boldsymbol{\beta}$. It is common to assume strong stationarity ($(Y_1, \ldots, Y_m)$ and $(Y_1 + \mathbf{h}, \ldots, Y_m + \mathbf{h})$ have the same distribution), so $C^*(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = C(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)$.

Inference for $g(\boldsymbol{\theta})$ at values of $\boldsymbol{\theta}_{r+1}$ not yet input to a simulation model is more flexible than linear regression via the correlation function $R(\mathbf{h}) = C(\mathbf{h})/C(0)$ for $\mathbf{h} \in \mathbb{R}^d$. See Santner et al. (2003) for examples. Kriging, a geostatistics term, is a best linear unbiased prediction (BLUP) for $g(\boldsymbol{\theta}_{r+1})$. An assessment of the uncertainty in $g(\boldsymbol{\theta}_{r+1})$ can be used as an experimental design technique to choose inputs to reduce response uncertainty (Santner et al., 2003). Example usage of GRFs includes input selection for efficient response surface estimation; percentile estimation; and model calibration (Sacks et al., 1989; O'Hagan et al., 1999; Kennedy and O'Hagan, 2001; Santner et al., 2003; Oakley, 2004). Stochastic observations can be modeled by assuming a measurement error, giving a so-called 'nugget effect' (Santner et al., 2003). Van Beers and Kleijnen (2003) found that a special case of Equation (13) was a useful metamodel of a stochastic process simulation. *GRFs provide an effective mechanism for reducing the computational effort to get a good response estimate by selecting simulation inputs on areas where the mean response has the greatest uncertainty. More work is needed for GRFs in the stochastic simulation context.*

4.2 Inference of input parameters from output information

Uncertainty analysis examines the distribution of $g(\boldsymbol{\theta}_c, \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e; \mathbf{U})$ induced by the distribution of $(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e)$ (whether g depends upon $\mathbf{U}$ or is deterministic). The question of calibration is whether the realization or distribution of $(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e)$ can be inferred from data or probability assessments about Y.

For example, the arrival rate λ and service rate μ of an $M/M/c$ queue are inputs to a simulation model. If they are unknown, their distribution can be inferred from field data. The distribution of λ, μ induces a distribution on system outputs, such as the average queue content $\overline{Q}$, and the 'clock speed' as measured by the autocorrelation of the queue occupancy process. The inverse problem is whether inputs λ and μ can be inferred from observation on the outputs such as the queue occupancy at discrete times $Q(\tau_1), \ldots, Q(\tau_n)$. McGrath et al. (1987), McGrath and Singpurwalla (1987), Armero and Bayarri (1997) comment on the potential to infer input parameters for queues from outputs, but mostly evaluate the expected value of information for inferring λ, μ from interarrival and service time data, along with potentially one observation $Q(\tau_1)$. The problem is hard because the likelihood function involves transient state transition probabilities, which are complicated for many stochastic processes. Larson (1990) describes frequentist inference with incomplete arrival time data, but using transactional data on arrivals and departures.

For epidemic models, O'Neill (2002) provides a review of tools and techniques. Chick et al. (2003) illustrate the inference of epidemic model (input) parameters from (output) data on disease incidence and germ contamination by estimating the likelihood model for outputs, given inputs, using approximations of the stationary distribution to help assess the likelihood function. Kennedy and O'Hagan (2001) provide a GRF model to infer $(\boldsymbol{\theta}_p, \boldsymbol{\theta}_e)$ given observations of $Y(\boldsymbol{\theta}_c)$ to help calibrate input parameters given output observations, assuming a nonstochastic response function g and that output data can be observed with random error. Also see Craig et al. (2001). Part of the problem of inferring input parameters from outputs arises from allowing the computer output to be an imperfect model of reality (e.g., with bias modeled as a Gaussian random field). The joint input–output model of Chick (1997) did not allow for the inference of input parameters from system outputs because it considered input models to be adequate for representing reality, but did not consider the outputs of the model to be exchangeable with the corresponding observations from the real system. The reason is that the lack of detail in computer models may not reflect the reality of the actual system. The joint input–output model augmented with an explicit bias metamodel helps quantify model validation concerns. The bias term accounts for the potential that model output might not be perfectly exchangeable with computer model output, conditional upon the values of the input parameters matching the values of the corresponding parameters in the real system.

Kraan and Bedford (2003) assess subjective prior distributions for inputs to a nonstochastic g that best induces a desired distribution for the outputs, in the sense of Kullback–Leibler divergence, even without data for Bayesian inference.

5 Ranking and selection

This section compares and contrasts frequentist and Bayesian approaches to ranking and selection. The objective of ranking and selection is to select the best of a finite number k of systems, where best is defined in terms of the expected performance of a system (e.g. Chapter 17). In the notation of Section 4, $\boldsymbol{\theta}_c$ assumes one of a discrete set of values indexed by $i \in \{1, 2, \ldots, k\}$, and the goal is to identify the system i that maximizes $w_i = \mathrm{E}[g(\boldsymbol{\theta}_{ci}, \boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e; \mathbf{U})]$. The means are inferred from observing simulation replications

$$Y_{ir} = w_i(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p, \boldsymbol{\theta}_e) + \sigma_i(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p, \boldsymbol{\theta}_e) z(\boldsymbol{\theta}_c, \boldsymbol{\theta}_p, \boldsymbol{\theta}_e; \mathbf{U}_{ri}) \tag{14}$$

for $i = 1, 2, \ldots, k$, $r = 1, 2, \ldots,$ where the function g is rewritten in terms of the mean w_i, standard deviation σ_i, and a zero-mean unit-variance noise z. The usual formulation of the problem does not explicitly specify a dependence upon $(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e)$, so we leave those parameters implicit in this section, and the response from different systems is not related with a metamodel $\boldsymbol{\psi}$ (although continuous optimization problems may use $\boldsymbol{\psi}$, see Chapter 18).

Both two-stage and sequential procedures are possible. In a two-stage procedure, a first stage consisting of r_0 replications for each system are run, initial estimates of $(\mathbf{w}, \boldsymbol{\sigma})$ are obtained in order to determine how many more samples to collect for each system, and then a second stage collects the samples and selects a system based on all output. Sequential procedures repeatedly allocate samples in a series of consecutive stages.

Different approaches to ranking and selection use different criteria for allocating samples and for measuring the quality of a selection. Frequentist approaches, like indifference zone (IZ) methods, provide correct selection guarantees over repeated applications of a selection procedure (see Chapter 17). Bayesian approaches, like the value of information procedures (VIP, Chick and Inoue, 2001a), and optimal computing budget allocation (OCBA, Chen, 1996), use posterior distributions to quantify the evidence for correct selection. These approaches are described and distinguished below.

Typical assumptions common to all three approaches are that the means and variances of the output of each system may differ, and that the output is independent and normally distributed, *conditional* on w_i and σ_i^2, for $i = 1, \ldots, k$,

$$\{Y_{ij}\colon j = 1, 2, \ldots\} \overset{\text{i.i.d.}}{\sim} \operatorname{Normal}\bigl(w_i, \sigma_i^2\bigr).$$

Although the normality assumption is not always applicable, it is often possible to batch a large number of outputs so that independence and normality is approximately satisfied (we return to correlated output in Section 5.3). It will be easier at times to refer to the precision $\lambda_i = 1/\sigma_i^2$ instead of the variance. Set $\mathbf{w} = (w_1, \ldots, w_k)$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)$. Let $w_{[1]} \leqslant w_{[2]} \leqslant \cdots \leqslant w_{[k]}$ be the ordered means. In practice, the ordering $[\cdot]$ is unknown, and the best system, system $[k]$, is to be identified with simulation. A problem instance ('configuration') is denoted by $\boldsymbol{\chi} = (\mathbf{w}, \boldsymbol{\sigma}^2)$. Let r_i be the number of simulation replications for system i run so far. Let $\bar{y}_i = \sum_{j=1}^{r_i} y_{ij}/r_i$ be the sample mean and $\hat{\sigma}_i^2 = \sum_{j=1}^{r_i}(y_{ij} - \bar{y}_i)^2/(r_i - 1)$ be the sample variance. Let $\bar{y}_{(1)} \leqslant \bar{y}_{(2)} \leqslant \cdots \leqslant \bar{y}_{(k)}$ be the ordering of the sample means based on all replications seen so far. Equality occurs with probability 0 in contexts of interest here. The quantities r_i, $\bar{y}_i$, $\hat{\sigma}_i^2$ and (i) may change as more replications are observed.

Because output is random, correct selection cannot be guaranteed with probability 1 with a finite number of replications. A correct selection occurs when the selected system, system $\mathfrak{D}$, is the best system $[k]$. Selection is based on a procedure's estimates $\hat{w}_i$ of w_i, for $i = 1, \ldots, k$, after all replications are observed. Usually overall sample means are the estimates, $\hat{w}_i = \bar{y}_i$, and the system with the best sample mean is selected as best, $\mathfrak{D} = (k)$, although some procedures may vary due to screening (Goldsman et al., 2002) or weighted samples (Dudewicz and Dalal, 1975).

The commonality between the three approaches stops there, and the differences now begin. The IZ (Chapter 17) approach seeks to guarantee a bound for the evidence for correct selection, with respect to repeated applications of the procedure to a given problem instance, for all problem instances within a specified class. For normally distributed output, the most widely-used criterion is a lower bound P^* for the probability of correct selection (Bechhofer et al., 1995), subject to the indifference zone constraint that the best system be at least a pre-specified amount $\delta^* > 0$ better than the others. Formally, the probability of correct selection ($\mathrm{PCS_{IZ}}$) is the probability that the system selected as best (system $\mathfrak{D}$) is the system with the highest mean (system $[k]$), conditional on the problem instance. The probability is with respect to the simulation output Y_{ij} from the procedure (the realizations y_{ij} determine $\mathfrak{D}$).

$$\mathrm{PCS_{IZ}}(\chi) \overset{\text{def}}{=} \Pr\bigl(w_{\mathfrak{D}} = w_{[k]} | \mathbf{w}, \boldsymbol{\sigma}^2\bigr). \tag{15}$$

The validity of an IZ procedure comes from proving statements like

$$\mathrm{PCS}_{\mathrm{IZ}}(\boldsymbol{\chi}) \geqslant P^* \text{ for all } \boldsymbol{\chi} = \left(\mathbf{w}, \boldsymbol{\sigma}^2\right) \text{ such that } w_{[k]} \geqslant w_{[k-1]} + \delta^*. \tag{16}$$

See Bechhofer et al. (1995) for other examples of the IZ approach with other distributions and indifference-zone structures. Early IZ procedures were statistically conservative in the sense of requiring many replications. More recent work with screening or sequential IZ procedures reduces the number of replications (Nelson et al., 2001; Goldsman et al., 2002).

Bayesian procedures model the evidence for correct selection with the posterior distribution of the unknown means and variances, given the data seen so far from a single application of the procedure. The basic Bayesian formulation is developed before the differences between the VIP and OCBA approaches are described. Given all output $\mathcal{E}$ seen so far for a single application of a procedure, the posterior probability of correct selection is

$$\mathrm{PCS}_{\mathrm{Bayes}} = \Pr(W_{\mathfrak{D}} = W_{[k]} | \mathcal{E}) \tag{17}$$

$$= 1 - \mathrm{E}\left[\mathcal{L}_{0\text{–}1}(\mathfrak{D}, \mathbf{W}) | \mathcal{E}\right], \tag{18}$$

where the 0–1 loss function $\mathcal{L}_{0\text{–}1}(\mathfrak{D}, \mathbf{w})$ equals 1 if $w_{\mathfrak{D}} = w_{[k]}$ and 0 otherwise. The expectation is taken over the decision, $\mathfrak{D}$, and the posterior distribution of the unknown means and variances. Assuming a noninformative prior distribution for the unknown mean and variance, the posterior marginal distribution for the unknown means W_i given $r > 3$ samples is $\mathrm{St}(\bar{y}_i, r_i/\hat{\sigma}_i^2, \nu_i)$, a shifted Student t distribution with mean $\bar{y}_i$, degrees of freedom $\nu_i = r_i - 1$, and variance $(\hat{\sigma}_i^2/r_i)\nu_i/(\nu_i - 2)$ (de Groot, 1970; Chick and Inoue, 2001a).

A comparison of Equations (15) and (17) emphasizes the difference between the PCS based on frequentist methods and Bayesian methods. Frequentist methods provide worst-case bounds for $\mathrm{PCS}_{\mathrm{IZ}}(\boldsymbol{\chi})$ subject to constraints on $\boldsymbol{\chi}$, and $\mathrm{PCS}_{\mathrm{IZ}}(\boldsymbol{\chi})$ is estimated by counting the fraction of correct selections from repeated applications of the procedure. Bayesian methods provide a measure of evidence for correct selection given the data seen in a single application of a procedure. Equation (18) emphasizes the link to the Bayesian decision-theoretic methods in Section 1.2. That link can be extended by generalizing the opportunity cost to the selection context here, $\mathcal{L}_{\mathrm{oc}}(\mathfrak{D}, \mathbf{w}) = w_{[k]} - w_{\mathfrak{D}}$. The loss is 0 when the best system is selected, and is the difference between the best and the selected system otherwise. This is an alternate measure of evidence for correct selection that makes more sense than PCS when simulation output represents financial value. The posterior expectation of the opportunity cost of a potentially incorrect selection is

$$\mathrm{EOC}_{\mathrm{Bayes}} = \mathrm{E}\left[\mathcal{L}_{\mathrm{oc}}(\mathfrak{D}, \mathbf{W}) | \mathcal{E}\right] = \mathrm{E}[W_{[k]} - W_{\mathfrak{D}} | \mathcal{E}]. \tag{19}$$

The frequentist $\mathrm{EOC}_{\mathrm{IZ}}(\boldsymbol{\chi}) = \mathrm{E}[w_{[k]} - W_{\mathfrak{D}} | \mathbf{w}, \boldsymbol{\sigma}^2]$ differs by having the expectation taken only over randomized $\mathfrak{D}$ for a given $\boldsymbol{\chi}$. This formalism is sufficient to describe the basic ideas behind the VIP and OCBA selection procedures.

5.1 Value of information procedures (VIPs)

Value of Information Procedures (VIPs) allocate additional samples in order to improve the expected value of information (EVI) of those samples with respect to a loss function. Chick and Inoue (2001a) provide four procedures. Both two-stage and sequential procedures are given for both the 0–1 loss function and opportunity cost. The EVI of additional samples are measured using the predictive distribution of additional output, along with the expected loss from Equations (18) or (19), with ideas paralleling those in Section 1.2.

After the first stage of sampling of $r_i = r_0$ replications per system in a VIP, the posterior distribution of the unknown mean and variance of each system is used as a prior distribution for the second stage. If noninformative prior distributions are used for the unknown mean and variance, the unknown means have a t distribution as described after Equation (18). The goal is to determine the second-stage allocation $\mathbf{r}' = (r'_1, r'_2, \ldots, r'_k)^{\mathrm{T}}$ that minimizes the *expected loss* to a decision maker after all replications have been run. Let $\mathbf{x}_{r'_i} = (x_{i,r_0+1}, \ldots, x_{i,r_0+r'_i})$ denote the second-stage output for system i, let $\mathbf{x}_{\mathbf{r}'} = (\mathbf{x}_{r'_1}, \ldots, \mathbf{x}_{r'_k})$ denote all second-stage output, and let $\mathfrak{D}(\mathbf{x}_{\mathbf{r}'})$ be the system with the highest overall sample mean after both stages. Given $\mathbf{x}_{\mathbf{r}'}$ and a loss function $\mathcal{L}$, the expected loss is $\mathrm{E}[\mathcal{L}(\mathfrak{D}(\mathbf{x}_{\mathbf{r}'}), \mathbf{W})|\mathbf{x}_{\mathbf{r}'}]$. Since $\mathbf{r}'$ is chosen before the second stage, we take the expectation with respect to the predictive distribution of $\mathbf{X}_{\mathbf{r}'}$. Let $\mathbf{c} = (c_1, \ldots, c_k)$ be the cost per replication of each system. The total cost of replications plus the expected loss for selecting the system with the best overall sample mean is (cf. Equation (7))

$$\rho^*(\mathbf{r}') \overset{\mathrm{def}}{=} \mathbf{c}\mathbf{r}' + \mathrm{E}\big[\mathrm{E}\big[\mathcal{L}\big(\mathfrak{D}(\mathbf{X}_{\mathbf{r}'}), \mathbf{W}\big)|\mathbf{X}_{\mathbf{r}'}\big]\big]. \tag{20}$$

If there is a computing budget constraint (e.g., for CPU time), samples are allocated to solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{r}'} \rho^*(\mathbf{r}') \\ &\quad \text{s.t.} \quad \mathbf{c}\mathbf{r}' = B, \\ &\qquad\quad\ \ r'_i \geqslant 0 \quad \text{for } i = 1, \ldots, k. \end{aligned} \tag{21}$$

Gupta and Miescke (1994) solved a special case of Problem (21). If $k = 2$ and $c_1 = c_2$, the optimal second-stage allocation minimizes the absolute difference of the posterior precision for the mean of each system, regardless of whether the 0–1 loss or opportunity cost is used. For the opportunity cost, $k \geqslant 2$, $c_1 = \cdots = c_k = 1$, and known precision, Gupta and Miescke (1996) provide an optimal allocation if $B = 1$.

Those special cases are not sufficient to address the complexity of problems found in simulation, and approximations are required to obtain readily computable allocations. Chick and Inoue (2001a) derived asymptotically optimal allocations that minimize a bound on the expected loss in Equation (20), a formulation that allows for unequal, unknown variances, different sampling costs, and a balance between sampling costs and the EVI of the samples. *The bound is obtained by examining the $k-1$ pairwise comparisons between the system with the highest first-stage sample mean and each other system.* The asymptotic approximation is like that for Equation (10). If $k = 2$, the bound is tight for the opportunity cost but is loose for the 0–1 loss due to an extra asymptotic approximation. *Sequential procedures* require one more approximation, as the number of replications of each system may be different after a given stage. This means that the EVI requires assessing the difference $W_i - W_j$ of t distributed random variables with different degrees of freedom (the Behrens–Fisher problem). The Welch (1938) approximation can be used to approximate the EVI and expected loss in Equation (21) (Chick and Inoue, 2001a). The resulting EOC (expected opportunity cost) and PCS VIPs minimize the following measures of incorrect selection:

$$\mathrm{EOC}_{\mathrm{Bonf}} = \sum_{i\neq\mathfrak{D}} \mathrm{E}\big[\max\{0, W_i - W_{\mathfrak{D}}\}|\mathcal{E}\big],$$

$$1 - \mathrm{PCS}_{\mathrm{Bonf}} = \sum_{i\neq\mathfrak{D}} \Pr\big(\{W_i \geqslant W_{\mathfrak{D}}\}|\mathcal{E}\big).$$

5.2 OCBA procedures

The third approach is the optimal computing budget allocation (OCBA) (Chen, 1996). The OCBA is based on several approximations, including the thought experiment in Equation (11) that approximates how additional replications would affect uncertainty about each W_i. Samples are allocated sequentially in a greedy manner to maximize an approximation to $\mathrm{PCS}_{\mathrm{Bayes}}$ at each stage. The approximations made by the original OCBA essentially assume that (a) the system with the best sample mean based on replications seen so far is to be selected, (b) a normal distribution can approximate uncertainty about each W_i, and (c) the effect of an additional small number of replications r' beyond the r_i done so far for system i, but none for the other systems, changes the uncertainty about the means to

$$p(\widetilde{W}_i) \sim \mathrm{Normal}\left(\bar{y}_i, \frac{\hat{\sigma}_i^2}{r_i + r'}\right),$$

$$p(\widetilde{W}_j) \sim \mathrm{Normal}\left(\bar{y}_j, \frac{\hat{\sigma}_j^2}{r_j}\right) \quad \text{for } j \neq i.$$

This induces an *estimated approximate probability of correct selection* that approximates the probability of correct selection with respect to $p(\widetilde{\mathbf{W}})$ and Slepian's inequality (Chapter 17).

$$\mathrm{EAPCS}_i = \prod_{j:\, j\neq(k)} \Pr\big(\{\widetilde{\mathbf{W}}\colon \widetilde{W}_j < \widetilde{W}_{(k)}\}|\mathcal{E}\big). \tag{22}$$

A small number r' of replications are allocated at each stage to the q systems with the largest $\mathrm{EAPCS}_i - \mathrm{PCS}_{\mathrm{Slep}}$, where the posterior evidence for correct selection is approximated with Slepian's inequality,

$$\mathrm{PCS}_{\mathrm{Slep}} = \prod_{j:\ j \neq (k)} \Pr\bigl(\{\mathbf{W}\colon\ W_j < W_{(k)}\} | \mathcal{E}\bigr),$$

and using the Welch approximation for the differences $W_j - W_{(k)}$. Chen et al. (2006, references therein) explored several variations on this theme, including varying r' and q; and the use of either normal or t distributions for W_i.

5.3 Comments

Surprisingly few papers to date compare representatives from each of the IZ, VIP and OCBA approaches. Chick and Inoue (2001a) found that VIPs compared favorably with the Combined Procedure of Nelson et al. (2001), when $\mathrm{PCS}_{\mathrm{IZ}}$ was measured against the average number of replications per system. Inoue et al. (1999) compared the VIP, OCBA and modified versions of the IZ procedure of Rinott (1978). Both two-stage and sequential VIPs performed well empirically over a broad class of problems. The VIP based on opportunity cost fared best with respect to several performance criteria. The VIP based on the 0–1 loss performed slightly less well than the opportunity cost procedure, even with respect to $\mathrm{PCS}_{\mathrm{IZ}}$, because of an extra approximation in its derivation. The fully sequential OCBA was also empirically very effective. The OCBA performed less well when run as a two-stage procedure, rather than sequentially, or when the values of r' and q were large (Inoue et al., 1999).

Branke et al. (2005) evaluate a variety of procedures, and introduce new 'stopping rules' for the VIP and OCBA procedures. The sequential stopping rule $\mathcal{S}$ samples τ replications per stage of sampling (after the first) until a fixed sampling budget is exhausted. That rule was used by Inoue et al. (1999). The new EOC stopping rule samples τ replications per stage (after the first) until a Bonferroni-like bound for the EOC is reached ($\mathrm{EOC}_{\mathrm{Bonf}} \leqslant \beta^*$ for some user-specified β^*). Other stopping rules, such as for $\mathrm{PCS}_{\mathrm{Slep}}$ are implemented similarly. Figure 5 gives a typical result that compares VIPs and OCBA procedures endowed with these stopping rules ($\tau = 1$), and a version of a leading sequential IZ procedure, $\mathcal{KN}$++ (Goldsman et al., 2002), adapted for independent output. As the number of replications or target posterior EOC values are changed for VIP and OCBA procedures, a different average number of replications and evidence for correct selection are observed. The curves are dependent upon the specific problem instance, which in this case is a single problem instance with evenly spaced means and a common variance, but some observations can be made.

In the top graphs of Figure 5, lower curves mean that more evidence per correct selection is obtained per average number of replications. For $\mathcal{KN}$++, δ^* was fixed to certain levels equal to, less than, and greater than the true difference in means between the best and second best (0.5 in this case), and

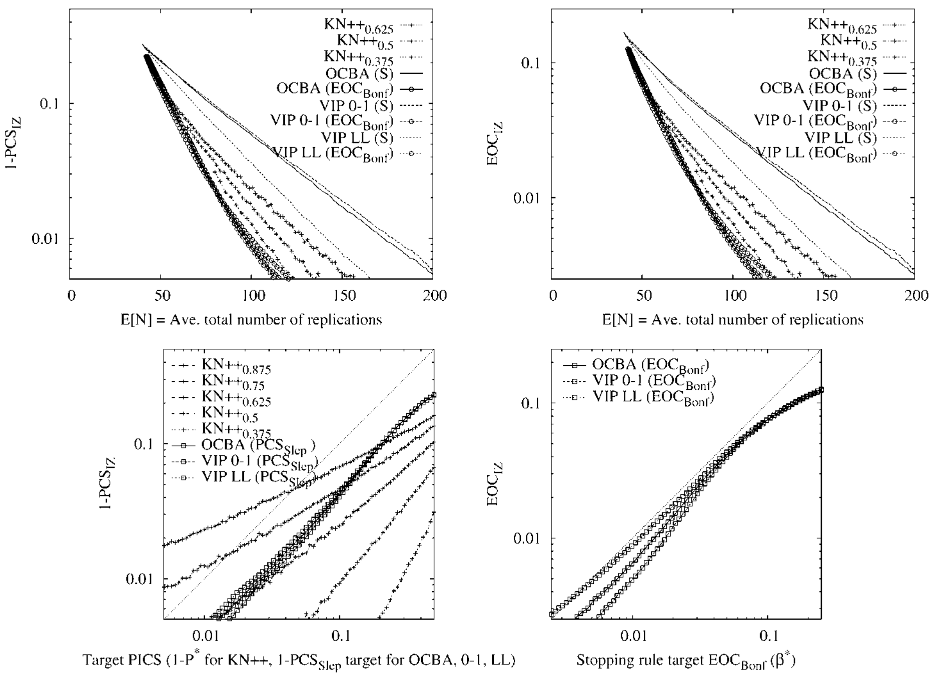


Fig. 5. Estimated evidence for correct selection as a function of the average number of replications per stage (top row), and relationship of parameter target versus empirical result (bottom row), for several selection procedures ($k = 10$ systems, $w_1 - w_i = (i - 1)/2$; $\sigma_i^2 = 1$ for $i = 1, 2, \ldots, k$, with $r_0 = 5$, estimates based on 10^5 macroreplications).

the PCS goal P^* was varied to obtain similar curves. The $\mathcal{S}$ stopping rule allows a practitioner to completely control the number of replications. Procedure $\mathcal{KN}$++ improves efficiency here by allowing a procedure to stop early if the evidence for correct selection is high early, or to continue sampling if more information is needed. The Bayesian VIP and OCBA procedures similarly can adapt the amount of sampling as a function of the posterior evidence for correct selection so far (here measured with $\mathrm{EOC}_{\mathrm{Bonf}}$), but improved upon $\mathcal{KN}$++ here due to more flexibility as to which systems get sampled. The bottom graphs in Figure 5 show the relationship between the targeted and empirically observed evidence for correct selection. The diagonal represents perfect calibration. The OCBA and VIP are slightly conservative (below the line), meaning that the observed $\mathrm{PCS}_{\mathrm{IZ}}$ and $\mathrm{EOC}_{\mathrm{IZ}}$ for this problem instance is somewhat 'better' than the corresponding Bayesian target levels. The graphs for $\mathcal{KN}$++ are straight, but may significantly underdeliver or overdeliver relative to the desired $\mathrm{PCS}_{\mathrm{IZ}}$ depending upon whether δ^* is selected to be larger, similar to, or smaller than the difference in means between the top two systems. A small δ^* results in excessive sampling for a given target P^*. For $\mathcal{KN}$++, the desired PCS and empirical $\mathrm{PCS}_{\mathrm{IZ}}$ tracked each other better for slippage configurations (all nonbest systems have the same mean) when δ^* was exactly the difference between the best and second best. Branke et al. (2005) provide a more thorough analysis and empirical study.

Common random numbers (CRN) can be used to sharpen contrasts across systems in two stage procedures. To date, the IZ approach provides more choices for procedures that accommodate CRN than the Bayesian approach. Kim and Nelson (2001) describe recent progress for IZ procedures with CRN. Chick and Inoue (2001b) present two-stage VIP procedures that handle CRN for both the opportunity cost and 0–1 loss functions. The procedures allow for simulating only a subset of systems during a second stage, but are not sequential. Because of correlation across systems, information is gained about the systems not simulated during the second stage using missing data techniques. Fu et al. (2004) examine CRN for the OCBA with sequential sampling.

CRN is typically implemented by using the same random number generators $\mathbf{u}_r$ to synchronize the stochastic uncertainty between simulated systems, so that the outputs Y_{ri} and Y_{rj} are positively correlated for replication r of systems i and j. When there is also input uncertainty, so that $(\boldsymbol{\Theta}_p, \boldsymbol{\Theta}_e)$ are described with probability rather than assumed to be a fixed input value, a second method for inducing CRN exists. Common values of $(\boldsymbol{\theta}_{pr}, \boldsymbol{\theta}_{er})$ are used for the rth simulation of each system. This can be useful for sharpening contrasts for systems even when input uncertainty is modeled.

Cross-fertilization between the approaches is also possible. Chick and Wu (2005) applied the opportunity cost idea from the VIP approach to the IZ framework to develop a two-stage procedure with a frequentist expected opportunity cost bound. The net effect is to replace the two parameters P^*, δ^* of the IZ approach with a single parameter Δ, the maximum acceptable expected opportunity cost, so that the frequentist expected opportunity cost $\mathrm{E}[w_{[k]} - w_{\mathfrak{D}} | \boldsymbol{\chi} = (\mathbf{w}, \boldsymbol{\sigma}^2)] \leqslant \Delta$ for all problem instances $\boldsymbol{\chi}$ (not just those in an indifference zone).

6 Discussion and future directions

Bayesian methods apply to simulation experiments in a variety of ways, including uncertainty analysis, ranking and selection, input distribution modeling, response surface modeling, and experimental design. One main theme is to represent all uncertainty with probability distributions, to update probability using Bayes' rule, and to use the expected value of information as a technique to make sampling decisions (e.g., the opportunity cost and 0–1 loss functions for selection procedures, or the Kullback–Leibler divergence for parameter estimation for linear response models). The other main theme is to use simulation to efficiently estimate quantities of interest for a Bayesian analysis. Asymptotic approximations are often helpful when exact optimal solutions are difficult to obtain. Research opportunities include:

- Input modeling and uncertainty analysis: kernel estimation for conditional means, with variability due to input uncertainty; improved modeling of prior distributions for statistical input parameters to obtain better models of uncertainty for simulation outputs (e.g., the conjugate prior distributions for parameters of an $M/M/1$ queue result in the absence of moments for quantities like the stationary average queue length, even conditioning on stability), including results for the calibration/inverse problem.
- Response modeling: extending the Gaussian random function work in the world of stochastic simulation; sampling plans for input parameter inference to optimally reduce output uncertainty, including nonasymptotic results, to help understand what data is most important to collect to infer the value of inputs for simulations.
- Ranking and selection: VIP procedures based on nonasymptotic EVI allocations for samples; analytical or empirical work to evaluate the IZ, VIP and OCBA approaches.
- Experimental designs: Estimating quantiles or other nonexpected value goals; nonGaussian output for ranking and selection and GRFs.
- Computational methods: improvements in MCMC and other sampling methods for posterior distributions.

Acknowledgement

Portions of this chapter were published by Chick (2004) in the *Proceedings of the Winter Simulation Conference*.

References

Andradóttir, S., Bier, V.M. (2000). Applying Bayesian ideas in simulation. *Simulation Theory and Practice* 8, 253–280.

Andradóttir, S., Glynn, P.W. (2004). Computing Bayesian means using simulation. Technical report, Georgia Tech, ISYE.

Armero, C., Bayarri, M. (1997). A Bayesian analysis of a queueing system with unlimited service. *Journal of Statistical Planning and Inference* 58, 241–261.

Barlow, R.E., Mendel, M.B. (1992). De Finetti-type representations for life distributions. *Journal of the American Statistical Association* 87, 1116–1122.

Barton, R.R., Schruben, L.W. (2001). Simulating real systems, submitted for publication.

Bechhofer, R.E., Santner, T.J., Goldsman, D.M. (1995). *Design and Analysis for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.

Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis*, 2nd edition. Springer-Verlag, New York.

Berger, J.O. (1994). An overview of robust Bayesian analysis. *TEST* 3, 5–124.

Berger, J.O., Delampady, M. (1987). Testing precise hypotheses (with discussion). *Statistical Science* 2, 317–352.

Berger, J.O., Pericchi, L.R. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association* 91, 109–122.

Berk, R. (1966). Limiting behaviour of posterior distributions when the model is incorrect. *Annals of Mathematical Statistics* 37 (1), 51–58.

Bernardo, J.M., Smith, A.F.M. (1994). *Bayesian Theory*. Wiley, Chichester, UK.

Billingsley, P. (1986). *Probability and Measure*, 2nd edition. Wiley, New York.

Branke, J., Chick, S.E., Schmidt, C. (2005). Selecting a selection procedure. Working paper, INSEAD, Technology Management Area, Fontainebleau, France.

Chen, C.-H. (1996). A lower bound for the correct subset-selection probability and its application to discrete event simulations. *IEEE Transactions on Automatic Control* 41 (8), 1227–1231.

Chen, C.-H., Yücesan, E., Dai, L., Chen, H. (2006). Efficient computation of optimal budget allocation for discrete event simulation experiment. *IIE Transactions*, in press.

Chen, H., Schmeiser, B. (1995). Monte Carlo estimation for guaranteed-coverage non-normal tolerance intervals. *Journal of Statistical Computation and Simulation* 51 (2–4), 223–238.

Chen, H.-C., Chen, C.-H., Lin, J., Yücesan, E. (1999). An asymptotic allocation for simultaneous simulation experiments. In: Farrington, P.A., Nembhard, H.B., Sturrock, D., Evans, G. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 359–366.

Cheng, R.C.H. (1999). Regression metamodelling in simulation using Bayesian methods. In: Farrington, P.A., Nembhard, H.B., Sturrock, D., Evans, G. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 330–335.

Cheng, R.C.H., Currie, C.S. (2003). Prior and candidate models in the Bayesian analysis of finite mixtures. In: Chick, S., Sánchez, P., Ferrin, D., Morrice, D. (Eds.), *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 392–398.

Cheng, R.C.H., Holland, W. (1997). Sensitivity of computer simulation experiments to errors in input data. *Journal of Statistical Computing and Simulation* 57, 219–241.

Chick, S.E. (1997). Bayesian analysis for simulation input and output. In: Andradóttir, S., Healy, K., Withers, D., Nelson, B. (Eds.), *Proceedings of the 1997 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 253–260.

Chick, S.E. (2001). Input distribution selection for simulation experiments: Accounting for input uncertainty. *Operations Research* 49 (5), 744–758.

Chick, S.E. (2004). Bayesian methods for discrete event simulation. In: Ingalls, R., Rossetti, M., Smith, J., Peters, B. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 89–100.

Chick, S.E., Inoue, K. (2001a). New two-stage and sequential procedures for selecting the best simulated system. *Operations Research* 49 (5), 732–743.

Chick, S.E., Inoue, K. (2001b). New procedures for identifying the best simulated system using common random numbers. *Management Science* 47 (8), 1133–1149.

Chick, S.E., Inoue, K. (2002). Corrigendum: New selection procedures. *Operations Research* 50 (3), 566.

Chick, S.E., Mendel, M.B. (1998). New characterizations of the no-aging property and the ℓ_1-isotropic models. *Journal of Applied Probability* 35 (4), 903–910.

Chick, S.E., Wu, Y. (2005). Selection procedures with frequentist expected opportunity cost bounds. *Operations Research* 53 (5), 867–878.

Chick, S.E., Soorapanth, S., Koopman, J.S. (2003). Inferring infection transmission parameters that influence water treatment decisions. *Management Science* 49 (7), 920–935.

Craig, P., Goldstein, M., Rougier, J., Seheult, A.H. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association* 96, 717–729.

Cressie, N.A. (1993). *Statistics for Spatial Data*. Wiley, New York.

de Finetti, B. (1990). *Theory of Probability, vol. 2*. Wiley, New York.

de Groot, M.H. (1970). *Optimal Statistical Decisions*. McGraw-Hill, New York.

Dmochowski, J. (1999). Intrinsic priors via Kullback–Leibler geometry. In: *Bayesian Statistics 5*. Oxford University Press, Oxford, UK, pp. 543–549.

Draper, D. (1995). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society B* 57 (1), 45–97.

Dudewicz, E.J., Dalal, S.R. (1975). Allocation of observations in ranking and selection with unequal variances. *Sankhyā B* 37 (1), 28–78.

Edwards, A.W.F. (1984). *Likelihood*. Cambridge University Press, Cambridge, UK.

Evans, M., Swartz, T. (1995). Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science* 10 (3), 254–272.

Fu, M.C., Hu, J.-Q., Chen, C.-H., Xiong, X. (2004). Optimal computing budget allocation under correlated sampling. In: Ingalls, R., Rossetti, M., Smith, J., Peters, B. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 595–603.

George, E.I., McCulloch, R.E. (1996). Stochastic search variable selection. In: Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (Eds.), *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, pp. 203–214.

Gilks, W.R., Best, N.G., Tan, K.K.C. (1995). Adaptive rejection Metropolis sampling. *Applied Statistics* 44, 455–472.

Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (Eds.) (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Glynn, P. (1986). Problems in Bayesian analysis of stochastic simulation. In: Wilson, J.R., Henriksen, J.O., Roberts, S.D. (Eds.), *Proceedings of the 1986 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 376–383.

Goldsman, D., Kim, S.-H., Marshall, W.S., Nelson, B.L. (2002). Ranking and selection for steady-state simulation: Procedures and perspectives. *INFORMS Journal on Computing* 14 (1), 2–19.

Gupta, S.S., Miescke, K.J. (1994). Bayesian look ahead one stage sampling allocations for selecting the largest normal mean. *Statistical Papers* 35, 169–177.

Gupta, S.S., Miescke, K.J. (1996). Bayesian look ahead one-stage sampling allocations for selecting the best population. *Journal of Statistical Planning and Inference* 54, 229–244.

Inoue, K., Chick, S.E., Chen, C.-H. (1999). An empirical evaluation of several methods to select the best system. *ACM Transactions on Modeling and Computer Simulation* 9 (4), 381–407.

Jaynes, E.T. (1983). *E.T. Jaynes: Papers on Probability, Statistics and Statistical Physics*. Reidel, Dordrecht, The Netherlands.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London A* 186, 453–461.

Kahneman, D., Slovic, P., Tversky, A. (1982). *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press, Cambridge.

Kass, R.E. (1989). The geometry of asymptotic inference. *Statistical Science* 4 (3), 188–234.

Kass, R.E., Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association* 91 (435), 1343–1370.

Kennedy, M., O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society B* 63, 425–464.

Kim, S.-H., Nelson, B.L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM TOMACS* 11, 251–273.

Kraan, B., Bedford, T. (2003). Probabilistic inversion of expert judgements in the quantification of model uncertainty. Research Paper No. 2003/12, Strathclyde Business School, Glasgow.

Laplace, P.S. (1812). *Theorie analytique des probabilites*. Courcier, Paris.

Larson, R.C. (1990). The queue inference engine: Deducing queue statistics from transactional data. *Management Science* 36, 586–601.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Lee, S.-H., Glynn, P. (1999). Computing the distribution function of a conditional expectation via Monte Carlo: Discrete conditioning spaces. In: Farrington, P.A., Nembhard, H.B., Sturrock, D., Evans, G. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1654–1663.

Lindley, D.V. (1957). A statistical paradox. *Biometrika* 44, 187–192.

Lindley, D.V. (1972). *Bayesian Statistics, A Review*. Society of Industrial and Applied Mathematics, Philadelphia.

McGrath, M.F., Singpurwalla, N.D. (1987). A subjective Bayesian approach to the theory of queues, II – Inference and information in $M/M/1$ queues. *Queueing Systems* 1, 335–353.

McGrath, M.F., Gross, D., Singpurwalla, N.D. (1987). A subjective Bayesian approach to the theory of queues I – Modeling. *Queueing Systems* 1, 317–333.

Nahmias, S. (2000). *Production and Operations Analysis*, 4th edition. McGraw-Hill, New York.

Naylor, J., Smith, A. (1982). Applications of a method for the efficient computation of posterior distributions. *Applied Statistics* 31, 214–225.

Nelson, B.L., Schmeiser, B.W., Taaffe, M.R., Wang, J. (1997). Approximation-assisted point estimation. *Operations Research Letters* 20 (3), 109–118.

Nelson, B.L., Swann, J., Goldsman, D., Song, W. (2001). Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research* 49 (6), 950–963.

Ng, S.-H., Chick, S.E. (2001). Reducing input distribution uncertainty for simulations. In: Peters, B., Smith, J., Rohrer, M., Madeiros, D. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 364–371.

Ng, S.-H., Chick, S.E. (2004). Design of follow-up experiments for improving model discrimination and parameter estimation. *Naval Research Logistics* 51 (8), 1129–1148.

Ng, S.-H., Chick, S.E. (2006). Reducing parameter uncertainty for stochastic systems. *ACM Transactions on Modeling and Computer Simulation* 15 (1), in press.

Oakley, J. (2004). Estimating percentiles of uncertain computer code outputs. *Applied Statistics* 53, 83–93.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison (with discussion). *Journal of the Royal Statistical Society Series B* 56, 99–118.

O'Hagan, A., Kennedy, M., Oakley, J. (1999). Uncertainty analysis and other inference tools for complex computer codes (with discussion). In: *Bayesian Statistics 6*. Oxford University Press, Oxford, UK, pp. 503–524.

O'Neill, P.D. (2002). A tutorial introduction to Bayesian inference for stochastic epidemic models using Markov chain Monte Carlo methods. *Mathematical Biosciences* 180, 103–114.

Raftery, A.E. (1995). Bayesian model selection in social research (with discussion by A. Gelman, D.B. Rubin, and R.M. Hauser, and a rejoinder). In: Marsden, P.V. (Ed.), *Sociological Methodology 1995*. Blackwell, Cambridge, MA, pp. 111–196.

Richardson, S., Green, P. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B* 59 (4), 731–792.

Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics A* 7 (8), 799–811.

Sacks, J., Welch, W., Mitchell, T., Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science* 4, 409–435.

Santner, T.J., Williams, B.J., Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York.

Savage, L.J. (1972). *The Foundations of Statistics*. Dover Publications, New York.

Scott, E.M. (1996). Uncertainty and sensitivity studies of models of environmental systems. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 255–259.

Smith, B.J. (2004). Bayesian output analysis program (BOA) v. 1.1 user's manual. Technical report, University of Iowa. Available at *http://www.public-health.uiowa.edu/boa/*.

Spiegelhalter, D.J., Best, N.G., Gilks, W.R., Inskip, H. (1996). Hepatitis B: A case study in MCMC methods. In: Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (Eds.), *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London, pp. 21–44.

Steckley, S.G., Henderson, S.G. (2003). A kernel approach to estimating the density of a conditional expectation. In: Chick, S., Sánchez, P., Ferrin, D., Morrice, D. (Eds.), *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 383–391.

Tanner, M.A. (1996). *Tools for Statistical Inference: Methods for Exploration of Posterior Distributions and Likelihood Functions*, 3rd edition. Springer-Verlag, New York.

The Mathworks (2002). Getting started with MATLAB. Natick, MA, USA.

van Beers, W., Kleijnen, J.P.C. (2003). Kriging for interpolation in random simulation. *Journal of the Operations Research Society* 54, 255–262.

Welch, B.L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika* 25, 350–362.

Winston, W. (2000). *Financial Models Using Simulation and Optimization*, 2nd edition. Palisade Corporation, Newfield, NY.

Zouaoui, F., Wilson, J.R. (2003). Accounting for parameter uncertainty in simulation input modeling. *IIE Transactions* 35, 781–792.

Zouaoui, F., Wilson, J.R. (2004). Accounting for input-model and input-parameter uncertainties in simulation. *IIE Transactions* 36 (11), 1135–1151.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13010-8

Chapter 10

A Hilbert Space Approach to Variance Reduction

Roberto Szechtman

Department of Operations Research, Naval Postgraduate School, Monterey, CA 93943, USA
E-mail: rszechtm@nps.edu

Abstract

In this chapter we explain variance reduction techniques from the Hilbert space standpoint, in the terminating simulation context. We use projection ideas to explain how variance is reduced, and to link different variance reduction techniques. Our focus is on the methods of control variates, conditional Monte Carlo, weighted Monte Carlo, stratification, and Latin hypercube sampling.

1 Introduction

The goal of this chapter is to describe variance reduction techniques from a Hilbert space perspective in the terminating simulation setting, with the focal point lying on the method of control variates. Several variance reduction techniques have an intuitive geometric interpretation in the Hilbert space setting, and it is often possible to obtain rather deep probabilistic results with relatively little effort by framing the relevant mathematical objects in an appropriate Hilbert space. The procedure employed to reach most results in this context consists of three stages:

1. Find a pertinent space endowed with an inner product.
2. Apply Hilbert space results toward a desired conclusion.
3. Translate the conclusions back into probabilistic language.

The key geometric idea used in Stage 2 is that of "projection": Given an element y in a Hilbert space H and a subset M of the Hilbert space, it holds under mild conditions that there exists a unique element in M that is closest to y. The characterization of this element depends on the space H determined by Stage 1, on y, and on M; in essence it is found by dropping a perpendicular from y to M.

The Hilbert space explanation of control variates, and to a somewhat lesser extent that of conditional Monte Carlo, is closely related to that of other variance reduction techniques; in this chapter we bridge these connections whenever they arise. From the projection perspective, it is often possible to link variance reduction techniques for which the relevant Hilbert space H and element $y \in H$ are the same. The articulation is done by judiciously choosing the subset M for each particular technique.

We do not attempt to provide a comprehensive survey of control variates or of the other techniques covered in this chapter. As to control variates, several publications furnish a broader picture; see, for example, Lavenberg and Welch (1981), Lavenberg et al. (1982), Wilson (1984), Rubinstein and Marcus (1985), Venkatraman and Wilson (1986), Law and Kelton (2000), Nelson (1990), Loh (1995) and Glasserman (2004). For additional material on other variance reduction techniques examined here, refer to the items in the References section and to references therein.

This chapter is organized as follows: Section 2 is an overview of control variates. In Section 3 we review Hilbert space theory and present several examples that serve as a foundation for the rest of the chapter. Section 4 is about control variates in Hilbert space. The focus of Section 5 is on the method of conditional Monte Carlo, and on combinations of conditional Monte Carlo with control variates. Section 6 describes how control variates and conditional Monte Carlo can reduce variance cooperatively. The subject of Section 7 is the method of weighted Monte Carlo. In Sections 8 and 9 we describe stratification techniques and Latin hypercube sampling, respectively. The last section presents an application of the techniques we investigate. As stated above, the focus of this chapter is in interpreting and connecting various variance reduction techniques in a Hilbert space framework.

2 Problem formulation and basic results

We study efficiency improvement techniques for the computation of an unknown scalar parameter α that can be represented as $\alpha = EY$, where Y is a random variable called the response variable, in the terminating simulation setting. Given n independent and identically distributed (i.i.d.) replicates $Y_1, \ldots, Y_n$ of Y produced by the simulation experiment, the standard estimator for α is the sample mean

$$\overline{Y} = \frac{1}{n}\sum_{k=1}^{n} Y_k.$$

The method of control variates (CVs) arises when the simulationist has available a random column vector $\mathbf{X} = (X_1, \ldots, X_d) \in \mathbb{R}^d$, called the control, such that $\mathbf{X}$ is jointly distributed with Y, $\mathrm{E}\mathbf{X} = \boldsymbol{\mu}_\mathbf{x}$ is known, and it is possible to obtain i.i.d. replicates $(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)$ of $(Y, \mathbf{X})$ as a simulation output. Under these conditions, the CV estimator is defined by

$$\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}) = \overline{Y} - \boldsymbol{\lambda}^{\mathrm{T}}(\overline{\mathbf{X}} - \boldsymbol{\mu}_\mathbf{x}), \tag{1}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d) \in \mathbb{R}^d$ is the vector of control variates coefficients; “T” denotes transpose, vectors are defined as columns, and vectors and matrices are written in bold.

The following holds throughout this chapter:

Assumption 1. $\mathrm{E}(Y^2 + \sum_{i=1}^d X_i^2) < \infty$ and the covariance of $(Y, \mathbf{X})$, defined by

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_y^2 & \boldsymbol{\Sigma}_{y\mathbf{x}} \\ \boldsymbol{\Sigma}_{\mathbf{x}y} & \boldsymbol{\Sigma}_{\mathbf{xx}} \end{pmatrix},$$

is nonsingular.

The first part of Assumption 1 is satisfied in most settings of practical interest. Furthermore, when $\boldsymbol{\Sigma}$ is singular it is often possible to make it nonsingular by reducing the number of controls $\mathbf{X}$; see the last paragraph of Example 5.

Naturally, $\boldsymbol{\lambda}$ is chosen to minimize $\operatorname{Var} \widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda})$, which is the same as

$$\text{minimizing } \sigma_y^2 - 2\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{x}y} + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{xx}}\boldsymbol{\lambda}. \tag{2}$$

The first- and second-order optimality conditions for this problem imply that there exists a unique optimal solution given by

$$\boldsymbol{\lambda}^* = \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}y}. \tag{3}$$

With this choice of $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ the CV estimator variance is

$$\operatorname{Var} \widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*) = \operatorname{Var} \overline{Y}\bigl(1 - R_{y\mathbf{x}}^2\bigr), \tag{4}$$

where

$$R_{y\mathbf{x}}^2 = \frac{\boldsymbol{\Sigma}_{y\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}y}}{\sigma_y^2}$$

is the square of the multiple correlation coefficient between Y and $\mathbf{X}$. CVs reduce variance because $0 \leqslant R_{y\mathbf{x}}^2 \leqslant 1$ implies $\operatorname{Var} \widehat{Y}_{\mathrm{CV}} \leqslant \operatorname{Var} \overline{Y}$ in (4). The central limit theorem (CLT) for $\widehat{Y}_{\mathrm{CV}}$ asserts that, under Assumption 1,

$$n^{1/2}\bigl(\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*) - \alpha\bigr) \Rightarrow \mathrm{N}\bigl(0, \sigma_{\mathrm{CV}}^2\bigr),$$

where $\sigma_{\mathrm{CV}}^2 = \sigma_y^2(1 - R_{y\mathbf{x}}^2)$, “$\Rightarrow$” denotes convergence in distribution and $\mathrm{N}(0, \sigma^2)$ is a zero-mean Normal random variable with variance σ^2.

In general, however, the covariance structure of the random vector $(Y, \mathbf{X})$ may not be fully known prior to the simulation. This difficulty can be overcome by using the available samples to estimate the unknown components of $\boldsymbol{\Sigma}$, which can then be used to estimate $\boldsymbol{\lambda}^*$. Let $\boldsymbol{\lambda}_n$ be an estimator of $\boldsymbol{\lambda}^*$ and suppose that $\boldsymbol{\lambda}_n \Rightarrow \boldsymbol{\lambda}^*$ as $n \to \infty$. Then, under Assumption 1,

$$n^{1/2}\bigl(\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}_n) - \alpha\bigr) \Rightarrow \mathrm{N}\bigl(0, \sigma_{\mathrm{CV}}^2\bigr) \tag{5}$$

as $n \to \infty$; see Glynn and Szechtman (2002) for details. Equation (5) means that estimating $\boldsymbol{\lambda}^*$ causes no loss of efficiency as $n \to \infty$, if $\boldsymbol{\lambda}_n \Rightarrow \boldsymbol{\lambda}$.

Thus far we have only considered linear control variates of the form $\overline{Y} - \boldsymbol{\lambda}^{\mathrm{T}}(\overline{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{x}})$. In some applications, however, the relationship between the response and the CVs is nonlinear, examples of which are: $\overline{Y}\exp(\boldsymbol{\lambda}^{\mathrm{T}}(\overline{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{x}}))$, $\overline{Y}\,\overline{X}/\mu_x$, and $\overline{Y}\mu_x/\overline{X}$. In order to have a general representation of CVs we introduce a function $f : \mathbb{R}^{d+1} \to \mathbb{R}$ that is continuous at $(y, \boldsymbol{\mu}_{\mathbf{x}})$ with $f(y, \boldsymbol{\mu}_{\mathbf{x}}) = y$. This property ensures that $f(\overline{Y}, \overline{\mathbf{X}}) \to \alpha$ a.s. if $(\overline{Y}, \overline{\mathbf{X}}) \to (\alpha, \boldsymbol{\mu}_{\mathbf{x}})$ a.s., so we only consider such functions.

The limiting behavior of $f(\overline{Y}, \overline{\mathbf{X}})$ is characterized in Glynn and Whitt (1989, Theorem 9) under the assumption that the i.i.d. sequence $\{(Y_n, \mathbf{X}_n) \colon n \geqslant 1\}$ satisfies the CLT $\sqrt{n}((\overline{Y}, \overline{\mathbf{X}}) - (\alpha, \boldsymbol{\mu}_{\mathbf{x}})) \Rightarrow \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$, and that f is continuously differentiable in a neighborhood of $(\alpha, \boldsymbol{\mu}_{\mathbf{x}})$ with first partial derivatives not all zero at $(\alpha, \boldsymbol{\mu}_{\mathbf{x}})$. Then

$$\sqrt{n}\bigl(f(\overline{Y}, \overline{\mathbf{X}}) - \alpha\bigr) \Rightarrow \mathrm{N}\bigl(0, \sigma_f^2\bigr) \tag{6}$$

as $n \to \infty$, where σ_f^2 is given by

$$\sigma_f^2 = \sigma_y^2 + 2\nabla_{\mathbf{x}} f(\alpha, \boldsymbol{\mu}_{\mathbf{x}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{x}y} + \nabla_{\mathbf{x}} f(\alpha, \boldsymbol{\mu}_{\mathbf{x}})^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{xx}} \nabla_{\mathbf{x}} f(\alpha, \boldsymbol{\mu}_{\mathbf{x}}), \tag{7}$$

and $\nabla_{\mathbf{x}} f(y, \mathbf{x}) \in \mathbb{R}^d$ is the vector with ith component $\partial f/\partial x_i(y, \mathbf{x})$.

The asymptotic variance σ_f^2 is minimized, according to Equation (2) with $\nabla_{\mathbf{x}} f(\alpha, \boldsymbol{\mu}_{\mathbf{x}})$ in lieu of $\boldsymbol{\lambda}$, by selecting f^* such that $\nabla_{\mathbf{x}} f^*(\alpha, \boldsymbol{\mu}_{\mathbf{x}}) = -\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \boldsymbol{\Sigma}_{\mathbf{x}y}$ in (7); that is, $\sigma_{f^*}^2 = \sigma_y^2(1 - R_{y\mathbf{x}}^2)$. In other words, nonlinear CVs have at best the same asymptotic efficiency as $\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*)$. Notice, however, that for small sample sizes it could happen that nonlinear CVs achieve more (or less) variance reduction than linear CVs.

The argument commonly used to prove this type of result is known as the Delta method; see Chapter 2 or, for a more detailed treatment, Serfling (1980, p. 122). The reason why $\sqrt{n}(f(\overline{Y}, \overline{\mathbf{X}}) - \alpha)$ converges in distribution to a normal random variable is that f is linear in a neighborhood of $(\alpha, \boldsymbol{\mu}_{\mathbf{x}})$ because f is differentiable there, and a linear function of a normal random variable is again normal.

To conclude this section, for simplicity let the dimension $d = 1$ and suppose that only an approximation of μ_x, say $\gamma = \mu_x + \varepsilon$ for some scalar ε, is known. This is the setting of biased control variates (BCV). The BCV estimator is given by

$$\widehat{Y}_{\mathrm{BCV}}(\lambda) = \overline{Y} - \lambda(\overline{X} - \gamma).$$

The bias of $\widehat{Y}_{\mathrm{BCV}}(\lambda)$ is $\lambda\varepsilon$, and the mean-squared error is

$$\mathrm{E}\bigl(\widehat{Y}_{\mathrm{BCV}}(\lambda) - \alpha\bigr)^2 = \mathrm{Var}\,\overline{Y} + \lambda^2\,\mathrm{E}(\overline{X} - \gamma)^2 - 2\lambda\,\mathrm{Cov}(\overline{Y}, \overline{X}).$$

Mean-squared error is minimized by

$$\lambda_n = \frac{\operatorname{Cov}(\overline{Y}, \overline{X})}{\mathrm{E}(\overline{X} - \gamma)^2} \tag{8}$$

and

$$\mathrm{E}\big(\widehat{Y}_{\mathrm{BCV}}(\lambda_n) - \alpha\big)^2 = \operatorname{Var} \overline{Y}\left(1 - \frac{\operatorname{Cov}(\overline{Y}, \overline{X})^2}{\operatorname{Var} \overline{Y}\, \mathrm{E}(\overline{X} - \gamma)^2}\right),$$

which is (4) when $\varepsilon = 0$; see Schmeiser et al. (2001) for more details on BCVs.

3 Hilbert spaces

We present basic ideas about Hilbert spaces, mainly drawn from Kreyszig (1978), Bollobas (1990), Zimmer (1990), Williams (1991) and Billingsley (1995). We encourage the reader to consult those references for proofs, and for additional material. We illustrate the concepts with a series of examples that serve as foundational material for the rest of the chapter.

An inner product space is a vector space V with an inner product $\langle x, y\rangle$ defined on it. An inner product on V is a mapping of $V \times V$ into $\mathbb{R}$ such that for all vectors x, y, z and scalars α, β we have

(i) $\langle \alpha x + \beta y, z\rangle = \alpha\langle x, z\rangle + \beta\langle y, z\rangle$;
(ii) $\langle x, x\rangle \geqslant 0$, with equality if and only if $x = 0$;
(iii) $\langle x, y\rangle = \langle y, x\rangle$.

An inner product defines a norm on X given by

$$\|x\| = \sqrt{\langle x, x\rangle}. \tag{9}$$

A Hilbert space H is a complete inner product space, complete meaning that every Cauchy sequence in H has a limit in H.

The next three examples present the Hilbert spaces that will be employed throughout this chapter.

Example 1. Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and

$$\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P}) = \left\{Y \in (\Omega, \mathcal{F}, \mathcal{P})\colon \mathrm{E}Y^2 = \int_\Omega Y(\omega)^2\, \mathrm{d}\mathcal{P}(\omega) < \infty\right\}$$

the space of square-integrable random variables defined on $(\Omega, \mathcal{F}, \mathcal{P})$. For $X, Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, the inner product is defined by

$$\langle X, Y\rangle = \mathrm{E}(XY) = \int_\Omega X(\omega)Y(\omega)\, \mathrm{d}\mathcal{P}(\omega), \tag{10}$$

and the norm is given by

$$\|Y\| = \sqrt{\mathrm{E}Y^2} = \left(\int_\Omega Y(\omega)^2 \,\mathrm{d}\mathcal{P}(\omega)\right)^{1/2}, \tag{11}$$

by (9). It can be easily verified that the inner product defined by (10) has properties (i), (ii) and (iii). The space $\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ is complete under this norm (Billingsley, 1995, p. 243). Note that

$$\operatorname{Var} Y = \|Y - \mathrm{E}Y\|^2. \tag{12}$$

Example 2. The space $\mathbb{R}^n$ is the set of vectors $\mathbf{x} = (x_1, \dots, x_n)$ in $\mathbb{R}^n$, and can be made into a Hilbert space by defining the inner product for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ as

$$\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{j=1}^{n} x_j y_j. \tag{13}$$

The norm induced by (13) is

$$\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle} = \left(\sum_{j=1}^{n} x_j^2\right)^{1/2}.$$

The space $\mathbb{R}^n$ is complete under this norm (Bollobas, 1990, p. 133).

Example 3. Consider independent random variables X_i with distribution function $F_i(x_i)$, $1 \leqslant i \leqslant d$, and define $F(\mathbf{x}) = \prod_{i=1}^{d} F_i(x_i)$ for $\mathbf{x} = (x_1, \dots, x_d)$. Write $\mathbf{X} = (X_1, \dots, X_d)$ and let $f : \mathbb{R}^d \to \mathbb{R}$ be a Borel-measurable function in $\mathcal{L}^2(\mathrm{d}F)$, the space of square integrable functions with respect to F. This space can be made into a Hilbert space by defining the inner product

$$\langle f, g\rangle = \int f(\mathbf{x}) g(\mathbf{x}) \,\mathrm{d}F(\mathbf{x}), \tag{14}$$

for any $f, g \in \mathcal{L}^2(\mathrm{d}F)$. The norm induced by (14) is

$$\|f\| = \left(\int f(\mathbf{x})^2 \,\mathrm{d}F(\mathbf{x})\right)^{1/2},$$

and $\mathcal{L}^2(\mathrm{d}F)$ is complete under this norm (Billingsley, 1995, p. 243).

The notion of orthogonality among elements lies at the heart of Hilbert space theory, and extends the notion of perpendicularity in Euclidean space. Two elements x, y are orthogonal if

$$\langle x, y\rangle = 0.$$

From here, there is just one step to the Pythagorean theorem:

Result 1 (Pythagorean theorem, Kreyszig, 1978). *If $x_1, \dots, x_n$ are pairwise orthogonal vectors of an inner product space V then*

$$\left\| \sum_{i=1}^{n} x_i \right\|^2 = \sum_{i=1}^{n} \|x_i\|^2. \tag{15}$$

Let us write

$$x^{\perp} = \{ y \in V \colon \langle x, y \rangle = 0 \}$$

for the set of orthogonal vectors to $x \in V$, and

$$S^{\perp} = \{ y \in V \colon \langle x, y \rangle = 0, \forall x \in S \}$$

for $S \subset V$. Finally, a set $(x_1, \dots, x_n)$ of elements in V is orthogonal if all its elements are pairwise orthogonal.

We often work with a subspace S of a Hilbert space H defined on X, by which we mean a vector subspace of X with the inner product restricted to $S \times S$. It is important to know when S is complete, and therefore a Hilbert space. It is easy to prove that S is complete if and only if S is closed in H.

Consider a nonempty subset M of an inner product space V. Central to the concepts discussed in this chapter is to know when, given a point y in V, there exists a unique point $x \in M$ that minimizes the distance from y to M, where the distance from y to M is defined to be $d(y, M) = \inf_{v \in M} \|y - v\|$. The following result provides an answer to this problem.

Result 2 (Projection theorem, Kreyszig, 1978). *Let M be a complete subspace of an inner product space V, and let $y \in V$ be fixed. Then there exists a unique $x = x(y) \in M$ such that*

$$d(y, M) = \|y - x\|.$$

Furthermore, every $y \in V$ has a unique representation of the form

$$y = x + z,$$

where $x \in M$ and $z \in M^{\perp}$. Then

$$\langle x, y - x \rangle = \langle x, z \rangle = 0. \tag{16}$$

The second part of Result 2 implies that if M is a complete subspace of an inner product space V, then there exists a function $P_M \colon V \to M$ defined by $P_M y = x$. We call P_M the orthogonal projection of V onto M, see Figure 1.

Among the properties that the projection functional enjoys are:

(a) $P_M x = x$, for all $x \in M$;
(b) $P_M z = 0$, for all $z \in M^{\perp}$;
(c) $\|I - P_M\| \leqslant 1$.

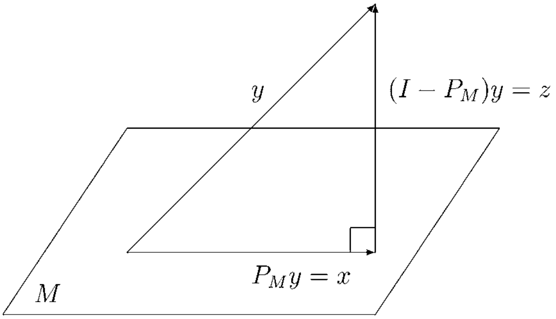


Fig. 1. Orthogonal projection.

Applying Result 2 to Examples 1–3 leads to several variance reduction techniques in the Hilbert space setting. The following example forms the basis for the method of conditional Monte Carlo.

Example 4. In the setting of Example 1, consider a sub-σ-algebra $\mathcal{G}$ of $\mathcal{F}$. The set of square integrable random variables defined on $\mathcal{L}^2(\Omega, \mathcal{G}, \mathcal{P})$ is a complete subspace of $\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$. Therefore, for $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ fixed, there exists an element $Z \in \mathcal{L}^2(\Omega, \mathcal{G}, \mathcal{P})$ that is the closest point to Y in $\mathcal{L}^2(\Omega, \mathcal{G}, \mathcal{P})$ and for which

$$\langle Y - Z, W \rangle = 0, \tag{17}$$

for $W \in \mathcal{L}^2(\Omega, \mathcal{G}, \mathcal{P})$ arbitrary. Choosing $W = I_B, B \in \mathcal{G}$, Equation (17) shows that Z is the conditional expectation of Y given $\mathcal{G}$, $Z = \mathrm{E}(Y|\mathcal{G})$. We also can write $P_{\mathcal{G}} Y = \mathrm{E}(Y|\mathcal{G})$; see Williams (1991) for more details.

Observe that Equation (17) and the Pythagorean theorem imply that

$$\|Y\|^2 = \|Y - Z\|^2 + \|Z\|^2. \tag{18}$$

Using Equation (12), centering Y so that $\mathrm{E}Y = 0$, we have that Equation (18) is the variance decomposition formula (Law and Kelton, 2000)

$$\operatorname{Var} Y = \mathrm{E} \operatorname{Var}(Y|\mathcal{G}) + \operatorname{Var} \mathrm{E}(Y|\mathcal{G}), \tag{19}$$

where $\operatorname{Var}(Y|\mathcal{G}) = \mathrm{E}(Y^2|\mathcal{G}) - (\mathrm{E}(Y|\mathcal{G}))^2$.

We continue with an example with a view toward control variates.

Example 5. For elements $X_1, \ldots, X_d \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ with zero mean (otherwise re-define $X_i := X_i - \mathrm{E}X_i$), define $M = \{Z \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})\colon Z = \sum_{i=1}^d \beta_i X_i,$ for all $\beta_i \in \mathbb{R}, i = 1, \ldots, d\}$. For $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, Result 2 then guarantees the existence of constants $\beta_1^* = \beta_1^*(Y), \ldots, \beta_d^* = \beta_d^*(Y)$ such that

$$P_M(Y - \mathrm{E}Y) = \sum_{i=1}^d \beta_i^* X_i \quad \text{and}$$

$$(I - P_M)(Y - \mathrm{E}Y) = (Y - \mathrm{E}Y) - \sum_{i=1}^{d} \beta_i^* X_i. \tag{20}$$

If $Y - \mathrm{E}Y \in M$, using property (a) of the projection operator we obtain $(I - P_M)(Y - \mathrm{E}Y) = 0$, so that

$$\mathrm{Var}\left(Y - \sum_{i=1}^{d} \beta_i^* X_i\right) = 0. \tag{21}$$

If the elements $X_1, \ldots, X_d$ form an orthogonal set, applying Equation (15) we have

$$\begin{aligned}\left\|(I - P_M)(Y - \mathrm{E}Y)\right\|^2 &= \|Y - \mathrm{E}Y\|^2 - \left\|P_M(Y - \mathrm{E}Y)\right\|^2 \\ &= \|Y - \mathrm{E}Y\|^2 - \sum_{i=1}^{d} \beta_i^{*2} \|X_i\|^2,\end{aligned}$$

so that

$$\mathrm{Var}\left(Y - \sum_{i=1}^{d} \beta_i^* X_i\right) = \mathrm{Var}\, Y - \sum_{i=1}^{d} \beta_i^{*2} \,\mathrm{Var}\, X_i. \tag{22}$$

When the elements $X_1, \ldots, X_d$ are not mutually orthogonal but linearly independent, the Gram–Schmidt process (Billingsley, 1995, p. 249) yields an orthogonal set with the same span as $X_1, \ldots, X_d$. In case the X_i's are linearly dependent, at least one X_i can be expressed as a linear combination of the others, and eliminated from the set $X_1, \ldots, X_d$. By noticing that $\mathrm{Cov}(X_i, X_j) = \langle X_i, X_j \rangle$ we gather that $X_1, \ldots, X_d$ are linearly independent if and only if their covariance matrix is nonsingular, and $X_1, \ldots, X_d$ are mutually orthogonal if and only if their covariance matrix has all its entries equal to zero except for positive numbers on the diagonal.

We can extend Example 5 to the setting of biased control variates.

Example 6. Let $X \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, and $M = \{Z \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})\colon Z = \beta(X - \gamma), \forall \beta \in \mathbb{R}\}$, $\gamma \neq \mathrm{E}X$. Fix an element $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ and let $\alpha = \mathrm{E}Y$. Project $Y - \alpha$ on M:

$$\begin{aligned}&P_M(Y - \alpha) = \beta^*(X - \gamma) \quad \text{and} \\ &(I - P_M)(Y - \alpha) = Y - \alpha - \beta^*(X - \gamma),\end{aligned}$$

for some $\beta^* = \beta^*(Y) \in \mathbb{R}$. As in the last example we have

$$\left\|(I - P_M)(Y - \alpha)\right\|^2 = \|Y - \alpha\|^2 - \left\|P_M(Y - \alpha)\right\|^2$$

and, since $\langle (I - P_M)(Y - \alpha), X - \gamma\rangle = 0$, it follows that

$$\beta^* = \frac{\langle Y - \alpha, X - \gamma\rangle}{\|X - \gamma\|^2}.$$

The Pythagorean theorem applied to the denominator in the last equation yields

$$\|X - \gamma\|^2 = \|X - \mathrm{E}X\|^2 + \|\mathrm{E}X - \gamma\|^2, \tag{23}$$

which is known as the bias-variance decomposition formula.

The following example is geared to the method of control variates when the optimal control coefficient is estimated from the sample data.

Example 7. Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ be elements of $\mathbb{R}^n$ (cf. Example 2), and define $S = \{\mathbf{z} \in \mathbb{R}^n\colon \mathbf{z} = \beta\mathbf{x}, \forall\beta \in \mathbb{R}\}$. By Result 2 we have

$$P_S\mathbf{y} = \beta_n\mathbf{x} \quad \text{and} \quad (I - P_S)\mathbf{y} = \mathbf{y} - \beta_n\mathbf{x},$$

for some $\beta_n = \beta_n(\mathbf{y})$. Because $\langle \mathbf{y} - \beta_n\mathbf{x}, \mathbf{x}\rangle = 0$, for $\|\mathbf{x}\| > 0$,

$$\beta_n = \frac{\langle \mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|^2}.$$

We now set the stage for the method of Latin hypercube sampling.

Example 8. Building on Example 3, let $M = \{h \in \mathcal{L}^2(\mathrm{d}F)\colon h(\mathbf{x}) = \sum_{i=1}^d h_i(x_i)\}$ be the subspace of $\mathcal{L}^2(\mathrm{d}F)$ spanned by the linear combinations of univariate functions $h_1, \ldots, h_d$. Because M is complete, appealing to Result 2 establishes the existence of an element $h^* = h^*(f) \in M$, $h^*(\mathbf{x}) = \sum_{i=1}^d h_i^*(x_i)$, such that $\|f - h^*\| = \inf_{h\in M} \|f - h\|$ for $f \in \mathcal{L}^2(\mathrm{d}F)$ fixed, and of a projection operator P_M: $P_M f = h^*$. Similarly, for $M_i = \{h \in \mathcal{L}^2(\mathrm{d}F)\colon h(\mathbf{x}) = h_i(x_i)\}$, $1 \leqslant i \leqslant d$, there exists $g_i^* = g_i^*(f) \in M_i : \|f - g_i^*\| = \inf_{h\in M_i} \|f - h\|$ and a projection P_i: $P_i f = g_i^*$. To complete the picture, define the subspace $M_0 = \{\beta \in \mathbb{R}\colon |\beta| < \infty\}$ which induces the projection P_0: $P_0 f = g_0^*$, for g_0^*: $\|f - g_0^*\| = \inf_{\beta\in M_0} \|f - \beta\|$. We now have:

- Let $\mathcal{F}_i = \sigma(\{\mathbb{R} \times \mathbb{R} \times \cdots \times \mathbb{R} \times B \times \mathbb{R} \times \cdots \times \mathbb{R} : B \in \mathcal{B}\})$, where $\mathcal{B}$ is the Borel σ-field, and let $\mathcal{F}_0$ be the trivial σ-algebra $\{\emptyset, \mathbb{R}\}$. For each P_i, we know that $\langle f - P_i f, h\rangle = 0$ for any $h \in M_i$; choosing $h(\mathbf{x}) = h_i(x_i) = I_B(x_i)$, $B \in \mathcal{B}$, shows that $P_i f = \mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$ for $1 \leqslant i \leqslant d$, and $P_0 f = \mathrm{E}(f(\mathbf{X})|\mathcal{F}_0) = \mathrm{E}f(\mathbf{X})$.
- Suppose $P_0 f = 0$. Then $(I - P_M)f \in M_i^\perp$ implies $g_i^* = P_i f = P_i P_M f = h_i^*$, which results in

$$P_M = \sum_{i=1}^d P_i \quad \text{and} \quad P_M f = \sum_{i=1}^d \mathrm{E}\big(f(\mathbf{X})|\mathcal{F}_i\big). \tag{24}$$

For general $P_0 f \neq 0$, (24) becomes

$$P_M = P_0 + \sum_{i=1}^{d} (P_i - P_0) \quad \text{and}$$

$$P_M f = \mathrm{E}f(\mathbf{X}) + \sum_{i=1}^{d} \bigl(\mathrm{E}\bigl(f(\mathbf{X})|\mathcal{F}_i\bigr) - \mathrm{E}f(\mathbf{X})\bigr). \tag{25}$$

The next result, a variant of Result 2, will be useful when we consider the method of weighted Monte Carlo.

Result 3 (Kreyszig, 1978). *Suppose $M \neq \emptyset$ is a closed convex subset of a Hilbert space H. Then, for $x \in H$ fixed, $x_1 = P_M(x)$ is the (unique) closest point in M to x if and only if*

$$\langle x - x_1, y - x_1 \rangle \leqslant 0 \quad \forall y \in M. \tag{26}$$

Later in the chapter we will deal with sequences of projections, say (P_n), defined on a Hilbert space H that are monotone increasing in that

$$\|P_i x\| \leqslant \|P_{i+1} x\| \quad \text{for } i = 1, 2, \ldots,$$

and $x \in H$ arbitrary. Using the completeness of H it can be shown that (P_n) converges in the following sense:

Result 4 (Kreyszig, 1978). *Let (P_n) be a monotone increasing sequence of projections P_n defined on a Hilbert space H. Then, for any $x \in H$,*

$$\|P_n x - P x\| \to 0,$$

and the limit operator P is a projection on H.

An immediate application of this result is the following example.

Example 9. Suppose that $(\mathcal{F}_n)$ is an increasing sequence $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_\infty$ of σ-algebras such that $\mathcal{F}_\infty = \sigma(\bigcup_{n=1}^{\infty} \mathcal{F}_n)$. Then associated with every $\mathcal{L}^2(\Omega, \mathcal{F}_n, \mathcal{P})$ there exists a projection $P_n : \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P}) \to \mathcal{L}^2(\Omega, \mathcal{F}_n, \mathcal{P})$, and the sequence of projections (P_n) is monotone increasing. Let P_∞ be the projection that results from applying Result 4: $\|P_n W - P_\infty W\| \to 0$ for any $W \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, $\mathcal{F}_\infty \subseteq \mathcal{F}$. Because $(I - P_\infty)W \in \mathcal{L}^2(\Omega, \mathcal{F}_n, \mathcal{P})^\perp$, we have

$$\int_B W \,\mathrm{d}\mathcal{P} = \int_B P_\infty W \,\mathrm{d}\mathcal{P} \tag{27}$$

for any $B \in \mathcal{F}_n$. A standard π–λ argument (Durrett, 1996, p. 263) shows that (27) holds for $B \in \mathcal{F}_\infty$ arbitrary; that is, $P_\infty W = \mathrm{E}(W|\mathcal{F}_\infty)$. The conclusion is

$$\bigl\|\mathrm{E}(W|\mathcal{F}_\infty) - \mathrm{E}(W|\mathcal{F}_n)\bigr\| \to 0 \tag{28}$$

as $n \to \infty$.

We will appeal to Example 9 when dealing with stratification techniques. A variation of the last example is

Example 10. Suppose that X is random variable with known and finite moments $\mathrm{E}X^i$, $i = 1, 2, \ldots$. Define a sequence of complete subspaces (M_d) of $\mathcal{L}^2(\Omega, \sigma(X), \mathcal{P})$ by

$$M_d = \Bigg\{ Z \in \mathcal{L}^2\big(\Omega, \sigma(X), \mathcal{P}\big): \\ Z = \sum_{i=1}^{d} \beta_i \big(X^i - \mathrm{E}X^i\big), \forall \beta_i \in \mathbb{R}, 1 \leqslant i \leqslant d \Bigg\}$$

for $d = 1, 2, \ldots$. Clearly $M_1 \subseteq M_2 \subseteq \cdots \subseteq M_\infty$, where $M_\infty = \bigcup_{i=1}^{\infty} M_i$. Associated with each M_d there is, by Result 2, a projection operator P_d with range on M_d such that, for $W \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, $\sigma(X) \subseteq \mathcal{F}$, with $\mathrm{E}W = 0$,

$$P_d W = \sum_{i=1}^{d} \beta_i^* \big(X^i - \mathrm{E}X^i\big) \tag{29}$$

for some constants $\beta_i^* = \beta_i^*(W)$, $1 \leqslant i \leqslant d$, possibly dependent on d (although this is not apparent from the notation). Because the sequence of operators (P_d) is (clearly) monotone increasing, Result 4 ensures the existence of a projection P_∞ in $\mathcal{L}^2(\Omega, \sigma(X), \mathcal{P})$ that satisfies

$$\|P_\infty W - P_d W\| \to 0 \tag{30}$$

as $d \to \infty$. Proceeding like in the last example it follows that $P_\infty W = \mathrm{E}(W|X)$, for $W \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ arbitrary. In other words,

$$\left\| \mathrm{E}(W|X) - \sum_{i=1}^{d} \beta_i^* \big(X^i - \mathrm{E}X^i\big) \right\| \to 0 \tag{31}$$

as $d \to \infty$, by Equations (29) and (30).

We will use the last example in Section 6 to show how conditional Monte Carlo and control variates reduce variance cooperatively.

The rest of the chapter is devoted to provide an interpretation of this section's examples in terms of variance reduction techniques. We start with the method of control variates.

4 A Hilbert space approach to control variates

We build on the sequence of examples from the previous section; Glynn and Szechtman (2002) is a relevant reference for the issues discussed in this section.

Consider the setting of Example 5: $X_1, \dots, X_d$ are zero-mean square integrable random variables with nonsingular covariance matrix $\mathbf{\Sigma_{xx}}$ (although this was not needed in Example 5), and defined on the same probability space as the response Y, $\mathrm{E}Y^2 < \infty$. The goal is to estimate $\alpha = \mathrm{E}Y$ by averaging n i.i.d. replicates of $Y - \sum_{i=1}^d \lambda_i X_i$ to obtain $\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda})$ as in Equation (1). Clearly, $\operatorname{Var} \widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}) = (1/n) \operatorname{Var}(Y - \sum_{i=1}^d \lambda_i X_i)$.

From Example 5, $\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*)$ is the remainder from projecting Y on M; as M "grows" the norm of the remainder decreases. Also, because the scalars $\lambda_1^*, \dots, \lambda_d^*$ that minimize $\operatorname{Var}(Y - \sum_{i=1}^d \lambda_i X_i)$ are also the numbers that result from projecting Y into M, from Result 2 and Equation (16) we know that

$$\left\langle Y - \sum_{i=1}^d \lambda_i^* X_i, Z \right\rangle = 0 \quad \forall Z \in M.$$

In particular,

$$\left\langle Y - \sum_{i=1}^d \lambda_i^* X_i, \lambda_k^* X_k \right\rangle = 0 \quad \text{for } k = 1, \dots, d. \tag{32}$$

Therefore, since $\operatorname{Cov}(Y, X_j) = \langle Y, X_j \rangle$ and $\operatorname{Cov}(X_i, X_j) = \langle X_i, X_j \rangle$ for $1 \leqslant i, j \leqslant d$,

$$\lambda_i^* = \left(\mathbf{\Sigma_{xx}^{-1}} \mathbf{\Sigma_{x}}_y\right)_i \quad \text{for } i = 1, \dots, d,$$

in concordance with Equation (3). When $X_1, \dots, X_d$ is an orthogonal set, Equation (32) yields

$$\lambda_i^* = \frac{\langle Y, X_i \rangle}{\langle X_i, X_i \rangle} = \frac{\operatorname{Cov}(Y, X_i)}{\operatorname{Var} X_i}, \quad i = 1, \dots, d. \tag{33}$$

We re-interpret the results from Examples 5–7 in the CV context:

(a) $\operatorname{Var} \widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*) \leqslant \operatorname{Var} \overline{Y}$ because

$$\begin{aligned}
&\operatorname{Var}\left(Y - \sum_{i=1}^d \lambda_i^* X_i\right) \\
&\quad = \left\|(I - P_M)(Y - \mathrm{E}Y)\right\|^2 \quad \text{by Equation (20)} \\
&\quad \leqslant \|I - P_M\|^2 \|Y - \mathrm{E}Y\|^2 \\
&\quad \leqslant \operatorname{Var} Y,
\end{aligned}$$

by using the projection operator properties of the last section.

(b) If Y can be expressed as a linear combination of $X_1, \dots, X_d$, then $\operatorname{Var}(Y - \sum_{i=1}^{d} \lambda_i X_i) = 0$ for some $\lambda_1, \dots, \lambda_d$; this is Equation (21).

(c) If the controls $X_1, \dots, X_d$ are mutually orthogonal, then

$$\operatorname{Var}\left(Y - \sum_{i=1}^{d} \lambda_i X_i\right) = \operatorname{Var} Y - \sum_{i=1}^{d} \frac{\operatorname{Cov}(Y, X_i)^2}{\operatorname{Var} X_i}$$
$$= \operatorname{Var} Y\left(1 - \sum_{i=1}^{d} \rho_{yx_i}^2\right),$$

by Equations (22) and (33), where ρ_{yx_i} is the correlation coefficient between Y and X_i, $i = 1, \dots, d$.

(d) With biased control variates in mind, apply Example 6 to the elements $(\overline{Y} - \alpha)$ and $(\overline{X} - \gamma)$ to get the optimal BCV coefficient

$$\lambda_n = \frac{\operatorname{Cov}(\overline{Y}, \overline{X})}{\operatorname{E}(\overline{X} - \gamma)^2},$$

as expected from (8). That is, BCVs as presented in Section 2 arise from taking the remainder of the projection of $\overline{Y} - \alpha$ on the span of $\overline{X} - \gamma$. Because of (23) we have

$$\operatorname{Var} \widehat{Y}_{\mathrm{CV}}\big(\lambda^*\big) \leqslant \operatorname{MSE} \widehat{Y}_{\mathrm{BCV}}(\lambda_n).$$

(e) Consider the setting of Example 7: There exists a zero-mean control variate $X \in \mathbb{R}$, and the output of the simulation are the sample points $\mathbf{y} = (y_1, \dots, y_n)$ and $\mathbf{x} = (x_1, \dots, x_n)$. Let $\tilde{\mathbf{y}} = (y_1 - \bar{y}, \dots, y_n - \bar{y})$, and define the estimator

$$\widehat{Y}_{\mathrm{CV}}(\lambda_n) = \frac{1}{n} \sum_{j=1}^{n} (y_j - \lambda_n x_j),$$

where $\lambda_n = \langle \mathbf{x}, \tilde{\mathbf{y}} \rangle / \|\mathbf{x}\|^2$. From Example 7 we know that λ_n arises from projecting $\tilde{\mathbf{y}}$ on the span of $\mathbf{x}$: $P_S \tilde{\mathbf{y}} = \lambda_n \mathbf{x}$. Now, the sample variance is

$$\frac{1}{n} \sum_{j=1}^{n} \big(y_j - \lambda_n x_j - \widehat{Y}_{\mathrm{CV}}(\lambda_n)\big)^2$$
$$= \frac{1}{n} \big\|(I - P_S)\tilde{\mathbf{y}}\big\|^2 - \big(\widehat{Y}_{\mathrm{CV}}(\lambda_n) - \bar{y}\big)^2$$
$$= \frac{1}{n} \big(\|\tilde{\mathbf{y}}\|^2 - \lambda_n^2 \|\mathbf{x}\|^2\big) + \mathrm{O}\big(n^{-1}\big)$$
$$= \frac{1}{n} \|\tilde{\mathbf{y}}\|^2 \big(1 - \rho_{\tilde{\mathbf{y}},\mathbf{x}}^2\big) + \mathrm{O}\big(n^{-1}\big),$$

where $\rho_{\tilde{\mathbf{y}},\mathbf{x}} = \langle \mathbf{x}, \tilde{\mathbf{y}} \rangle / \|\mathbf{x}\| \|\tilde{\mathbf{y}}\|$, which makes precise the variance reduction achieved by projecting $\tilde{\mathbf{y}}$ on the span of $\mathbf{x}$ relative to the crude estimator sample variance $\|\tilde{\mathbf{y}}\|^2/n$.

Finally, we remark that there is no impediment in extending items (a)–(e) to the multi-response setting, where Y is a random vector.

5 Conditional Monte Carlo in Hilbert space

In this section we address the method of conditional Monte Carlo, paying special attention to its connection with control variates; we follow Avramidis and Wilson (1996) and Loh (1995).

Suppose $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ and that we wish to compute $\alpha = \mathrm{E}Y$. Let $X \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ be such that $\mathrm{E}(Y|X)$ can be analytically or numerically computed. Then

$$\widehat{Y}_{\mathrm{CMC}} = \frac{1}{n} \sum_{j=1}^{n} \mathrm{E}(Y|X_i)$$

is an unbiased estimator of α, where the $\mathrm{E}(Y|X_i)$ are found by first obtaining i.i.d. samples X_i and then computing $\mathrm{E}(Y|X_i)$. We call $\widehat{Y}_{\mathrm{CMC}}$ the conditional Monte Carlo (CMC) estimator of α; remember that according to Example 4, $\widehat{Y}_{\mathrm{CMC}}$ results by projecting Y on $\mathcal{L}^2(\Omega, \sigma(X), \mathcal{P})$. The variability of $\widehat{Y}_{\mathrm{CMC}}$ is given by

$$\mathrm{Var}\, \widehat{Y}_{\mathrm{CMC}} = \frac{1}{n} \mathrm{Var}\, \mathrm{E}(Y|X),$$

with Equation (19) implying that $\mathrm{Var}\, \hat{Y}_{\mathrm{CMC}} \leqslant \mathrm{Var}\, \overline{Y}$. Specifically, CMC eliminates the $\mathrm{E}\,\mathrm{Var}(Y|X)$ term from the variance of Y.

Sampling from $Y - \lambda(Y - \mathrm{E}(Y|X))$ also provides an unbiased estimator of α for any $\lambda \in \mathbb{R}$. By Equation (3),

$$\begin{aligned} \lambda^* &= \frac{\langle Y, (I - P_{\sigma(X)})Y \rangle}{\|(I - P_{\sigma(X)})Y\|^2} \\ &= \frac{\langle P_{\sigma(X)}Y + (I - P_{\sigma(X)})Y, (I - P_{\sigma(X)})Y \rangle}{\|(I - P_{\sigma(X)})Y\|^2} \\ &= 1. \end{aligned} \tag{34}$$

This shows that CMC is optimal from a CV perspective. Avramidis and Wilson (1996) and Loh (1995) generalize this approach: Let Z be a zero-mean random variable in $\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ and X a random variable in $\mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ for which both $\mathrm{E}(Y|X)$ and $\mathrm{E}(Z|X)$ can be determined. Then sampling from

$$Y - \lambda_1\bigl(Y - \mathrm{E}(Y|X)\bigr) - \lambda_2\, \mathrm{E}(Z|X) - \lambda_3 Z \tag{35}$$

can be used to form the standard means based estimator for α, for all $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$. Repeating the logic leading to (34) we obtain

$$\lambda_1^* = 1, \qquad \lambda_2^* = \frac{\operatorname{Cov}(\mathrm{E}(Y|X), \mathrm{E}(Z|X))}{\operatorname{Var}\mathrm{E}(Z|X)} \quad \text{and} \quad \lambda_3^* = 0.$$

The conclusion is that

$$\operatorname{Var}\biggl(\mathrm{E}(Y|X) - \frac{\operatorname{Cov}(\mathrm{E}(Y|X), \mathrm{E}(Z|X))}{\operatorname{Var}\mathrm{E}(Z|X)}\,\mathrm{E}(Z|X)\biggr) \leqslant \begin{cases} \operatorname{Var} Y, \\ \operatorname{Var}\mathrm{E}(Y|X), \\ \operatorname{Var}\bigl(Y - \frac{\operatorname{Cov}(Y,Z)}{\operatorname{Var} Z} Z\bigr), \\ \operatorname{Var}\bigl(Y - \frac{\operatorname{Cov}(Y,\mathrm{E}(Z|X))}{\operatorname{Var}\mathrm{E}(Z|X)}\,\mathrm{E}(Z|X)\bigr), \\ \operatorname{Var}\bigl(\mathrm{E}(Y|X) - \frac{\operatorname{Cov}(\mathrm{E}(Y|X),Z)}{\operatorname{Var} Z} Z\bigr). \end{cases}$$

In particular, Loh (1995) considers the case of $Z = X$ almost surely in (35), and Avramidis and Wilson (1996) fix $\lambda_1 = 1$ and $\lambda_3 = 0$ in (35). From the norm perspective,

$$\begin{aligned} &\bigl\|\mathrm{E}(Y|X) - \alpha - \lambda_2^*\,\mathrm{E}(Z|X)\bigr\|^2 \\ &\quad = \|Y - \alpha\|^2 - \bigl\|Y - \mathrm{E}(Y|X)\bigr\|^2 - \bigl\|\lambda_2^*\,\mathrm{E}(Z|X)\bigr\|^2 \end{aligned}$$

makes precise the variance eliminated when sampling from $\mathrm{E}(Y|X) - \lambda_2^*\,\mathrm{E}(Z|X)$.

6 Control variates and conditional Monte Carlo from a Hilbert space perspective

We now discuss how CMC and CV can be combined to reduce variance cooperatively; the results of this section appear in Loh (1995) and Glynn and Szechtman (2002).

Suppose the setting of Example 10: There exists a random variable $X \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$ such that the moments $\mathrm{E}X^i$ are known with $\mathrm{E}|X|^i < \infty$ for all $i = 1, 2, \ldots$. Given a random variable $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$, the goal is to find $\alpha = \mathrm{E}Y$ by running a Monte Carlo simulation that uses the knowledge about the moments of X to increase simulation efficiency. Suppose we can sample from either

(a) $\mathrm{E}(Y|X) - \sum_{i=1}^{d} \lambda_i^*(X^i - \mathrm{E}X^i)$

or

(b) $Y - \sum_{i=1}^{d} \lambda_i^*(X^i - \mathrm{E}X^i)$

to form the standard estimator for α, where the λ_i^* are determined by applying Equation (3) on $\mathrm{E}(Y|X)$ and on the controls $(X^1 - \mathrm{E}X^1, \ldots, X^d - \mathrm{E}X^d)$. From the developments of Example 10 it is a short step to:

(a) Take $W = \mathrm{E}(Y|X) - \alpha$ in Equation (31), consequently

$$\mathrm{Var}\left(\mathrm{E}(Y|X) - \sum_{i=1}^{d} \lambda_i^*\left(X^i - \mathrm{E}X^i\right)\right) \to 0 \tag{36}$$

as $d \to \infty$.

(b) The triangle inequality and $W = Y - \alpha$ in Equation (31) result in

$$\mathrm{Var}\left(Y - \sum_{i=1}^{d} \lambda_i^*\left(X^i - \mathrm{E}X^i\right)\right) \to \mathrm{E}\,\mathrm{Var}(Y|X) \tag{37}$$

as $d \to \infty$.

The interpretation of (a) is that CV and CMC reduce variance concurrently: $\mathrm{E}(Y|X)$ eliminates the $\mathrm{E}\,\mathrm{Var}(Y|X)$ part of $\mathrm{Var}\,Y$, while $\sum_{i=1}^{d} \lambda_i^*(X^i - \mathrm{E}X^i)$ asymptotically cancels $\mathrm{Var}\,\mathrm{E}(Y|X)$. The effect of $\sum_{i=1}^{d} \lambda_i^*(X^i - \mathrm{E}X^i)$ in part (b) is to asymptotically eliminate the variance component due to $\mathrm{E}(Y|X)$ when using $\mathrm{E}(Y|X)$ in the simulation is not possible.

7 Weighted Monte Carlo

In this section we consider the asymptotic behavior of weighted Monte Carlo (WMC) estimators, for a large class of objective functions. We rely on Glasserman and Yu (2005) and Glasserman (2004), which make precise the connection between WMC and CVs for separable convex objective functions. Initial results, under weaker assumptions and just for one class of objectives were obtained in Szechtman and Glynn (2001) and in Glynn and Szechtman (2002). Applications of weighted estimators to model calibration in the finance context are presented in Avellaneda et al. (2001) and in Avellaneda and Gamba (2000).

Consider the standard CV setting: $(Y_1, \mathbf{X}_1), \ldots, (Y_n, \mathbf{X}_n)$ are i.i.d. samples of jointly distributed random elements $(Y, \mathbf{X}) \in (\mathbb{R}, \mathbb{R}^d)$ with nonsingular covariance matrix $\boldsymbol{\Sigma}$ and, without loss of generality, $\mathrm{E}\mathbf{X} = \mathbf{0}$ componentwise. The goal is to compute $\alpha = \mathrm{E}Y$ by Monte Carlo simulation, using information about the means $\mathrm{E}\mathbf{X}$ to reduce estimator variance. Let $f: \mathbb{R} \to \mathbb{R}$ be a strictly convex and continuously differentiable function and suppose that the weights $w_{1,n}^*, \ldots, w_{n,n}^*$,

$$\text{minimize} \quad \sum_{k=1}^{n} f(w_{k,n}) \tag{38}$$

$$\text{subject to} \quad \frac{1}{n}\sum_{k=1}^{n} w_{k,n} = 1, \tag{39}$$

$$\frac{1}{n}\sum_{k=1}^{n} w_{k,n}\mathbf{X}_k = \mathbf{0}. \tag{40}$$

Then, the WMC estimator of α takes the form

$$\widehat{Y}_{\mathrm{WMC}} = \frac{1}{n}\sum_{k=1}^{n} w^*_{k,n} Y_k.$$

The following observations review some key properties of WMC: The weight applied to each replication i is $w^*_{i,n}/n$, rather than the weight $1/n$ used to form the sample mean $\overline{Y}$. A feasible set of weights is one that makes $\widehat{Y}_{\mathrm{WMC}}$ unbiased (cf. constraint (39)), and that forces the weighted average of the control samples to match their known mean (cf. constraint (40)). For every n sufficiently large $\mathbf{0}$ ($= \mathrm{E}\mathbf{X}$) belongs to the convex hull of the replicates $\mathbf{X}_1, \ldots, \mathbf{X}_n$, and therefore the constraint set is nonempty. The objective function in (38), being strictly convex, ensures uniqueness of the optimal solution if the optimal solution is finite. If $w_{k,n} \geqslant 0, 1 \leqslant k \leqslant n$, were additional constraints, a feasible set of weights $w_{1,n}, \ldots, w_{n,n}$ would determine a probability mass function $(1/n)\sum_{k=1}^{n} \delta_{\mathbf{X}_k}(\cdot) w_{k,n}$, where $\delta_{\mathbf{x}}(\mathbf{z}) = 1$ if $\mathbf{z} = \mathbf{x}$ and is equal to zero otherwise. However, as discussed in Hesterberg and Nelson (1998), $\mathrm{P}(w_{k,n} < 0) = \mathrm{o}(n^{-p})$ uniformly in $1 \leqslant k \leqslant n$ if $\mathrm{E}(\|\mathbf{X}\|^p) < \infty$ indicates that the nonnegativity constraints are asymptotically not binding; see Szechtman and Glynn (2001) for an example of this scenario.

There are different f's depending on the application setting. For example: $f(w) = -\log w$ results in maximizing empirical likelihood; discussed in Szechtman and Glynn (2001). The function $f(w) = -w \log w$ yields an entropy maximization objective; this is the subject of Avellaneda and Gamba (2000) and Avellaneda et al. (2001). The important case of $f(w) = w^2$ is considered next, the optimization problem being to

$$\text{minimize} \quad \sum_{k=1}^{n} w^2_{k,n}$$

$$\text{subject to} \quad \frac{1}{n}\sum_{k=1}^{n} w_{k,n} = 1, \tag{41}$$

$$\frac{1}{n}\sum_{k=1}^{n} w_{k,n}\mathbf{X}_k = \mathbf{0}.$$

Solving the optimization problem yields (Glasserman and Yu, 2005) optimal weights given by

$$w^*_{k,n} = 1 - \overline{\mathbf{X}}^{\mathrm{T}}\mathbf{M}^{-1}(\mathbf{X}_k - \overline{\mathbf{X}}) \quad \text{for } k = 1, \ldots, n, \tag{42}$$

where $\mathbf{M} \in \mathbb{R}^{d\times d}$ is the matrix with elements $M_{i,j} = (1/n)\sum_{k=1}^{n}(X_{i,k} - \overline{X})(X_{j,k} - \overline{X})$; $\mathbf{M}^{-1}$ exists for all n large enough because $\mathbf{M} \to \mathbf{\Sigma_{xx}}$ a.s. componentwise. Rearranging terms immediately gives

$$\frac{1}{n}\sum_{k=1}^{n} w^{*}_{k,n} Y_k = \widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}_n), \tag{43}$$

with the benefit that the optimal weights do not depend on the Y_k, which makes this approach advantageous when using CVs for quantile estimation; see Hesterberg and Nelson (1998) for details.

Regarding Hilbert spaces, consider the space $\mathbb{R}^n$ (cf. Example 2) and the set

$$A(n) = \Biggl\{\mathbf{w}(n) = (w_{1,n}, \ldots, w_{n,n}) \in \mathbb{R}^n\colon \frac{1}{n}\sum_{k=1}^{n} w_{k,n} = 1 \text{ and } \frac{1}{n}\sum_{k=1}^{n} w_{k,n}\mathbf{X}_k = \mathbf{0}\Biggr\}.$$

It can be verified that $A(n)$ meets the conditions of Result 3 for every n sufficiently large, and consequently it is fair to ask: What element in $A(n)$ is closest to $\mathbf{1}(n) = (1, \ldots, 1) \in \mathbb{R}^n$? That is, which element $\mathbf{w}^{*}(n) = (w^{*}_{1,n}, \ldots, w^{*}_{n,n}) \in \mathbb{R}^n$

$$\begin{aligned} &\text{minimizes} \quad \bigl\|\mathbf{1}(n) - \mathbf{w}(n)\bigr\| \\ &\text{subject to} \quad \mathbf{w}(n) \in A(n)? \end{aligned}$$

This problem yields the same solution as problem (41); doing simple algebra it is easy to verify that $\mathbf{w}^{*}(n)$ with components as in Equation (42) satisfies condition (26). The conclusion is that $\mathbf{w}^{*}(n)$ is the closest point in $A(n)$ to $\mathbf{1}(n)$, the vector of crude sample weights. Would this Hilbert space approach to WMC work with f's that are not quadratic? Yes, as long as the metric induced by the inner product meets the defining properties of a metric.

The main result concerning WMC and CV, proved in Glasserman and Yu (2005) under certain conditions on $\mathbf{X}$, Y, f, and the Lagrange multipliers associated with constraints (39) and (40), is that

$$\widehat{Y}_{\mathrm{WMC}} = \widehat{Y}_{\mathrm{CV}} + \mathrm{O}_p\bigl(n^{-1}\bigr)$$

and

$$\sqrt{n}\bigl(\widehat{Y}_{\mathrm{WMC}} - \alpha\bigr) \Rightarrow \mathrm{N}\bigl(0, \sigma^2_{\mathrm{WMC}}\bigr)$$

as $n \to \infty$ where

$$\sigma^2_{\mathrm{WMC}} = \sigma^2_{\mathrm{CV}},$$

and $\mathrm{O}_p(a_n)$ stands for a sequence of random variables $(\xi_n\colon\ n \geqslant 1)$ such that for all $\varepsilon > 0$ and some constant δ, $P(|\xi_n| \geqslant a_n\delta) < \varepsilon$. The last result provides support to the statement that $\widehat{Y}_{\mathrm{WMC}}$ and $\widehat{Y}_{\mathrm{CV}}$ are asymptotically identical.

8 Stratification techniques

In this section we discuss stratification methods emphasizing the connection with the Hilbert space and CVs ideas already developed. Refer to Fishman (1996), Glasserman et al. (1999) and Glynn and Szechtman (2002) for more details.

Suppose that we wish to compute $\alpha = \mathrm{E}Y$, for some random variable $Y \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$. Let $X \in \mathcal{L}^2(\Omega, \mathcal{F}, \mathcal{P})$. The method of stratification arises when there is a collection of disjoint sets ("strata") $(\mathcal{A}_i\colon 1 \leqslant i \leqslant d)$ in the range of X such that $\mathrm{P}(X \in \bigcup_{i=1}^d \mathcal{A}_i) = 1$ and $\mathrm{P}(X \in \mathcal{A}_i) = p_i$ is known for every $1 \leqslant i \leqslant d$. Then, assuming that one can obtain i.i.d. replicates $(Y_{i,k}\colon 1 \leqslant k \leqslant n_i)$ from $\mathrm{P}(Y \in \cdot | X \in \mathcal{A}_i)$, $1 \leqslant i \leqslant d$, the estimator of α given by

$$\sum_{i=1}^{d} \frac{p_i}{n_i} \sum_{k=1}^{n_i} Y_{i,k} \tag{44}$$

is unbiased, where n_i is the number of replicates sampled from $\mathrm{P}(Y \in \cdot | X \in \mathcal{A}_i)$.

For a total number of replications $n = \sum_{i=1}^d n_i$, proportional stratification allocates $n_i = np_i$ samples to strata $\mathcal{A}_i$, $1 \leqslant i \leqslant d$, where for simplicity we assume that the np_i are integers. The estimator of Equation (44) is then called the proportional stratification (PS) estimator

$$\widehat{Y}_{\mathrm{PS}} = \frac{1}{n} \sum_{i=1}^{d} \sum_{k=1}^{np_i} Y_{i,k},$$

with variance given by

$$\begin{aligned}\operatorname{Var} \widehat{Y}_{\mathrm{PS}} &= \frac{1}{n} \sum_{i=1}^{d} p_i \operatorname{Var}(Y | Z = i) \\ &= \frac{1}{n} \mathrm{E} \operatorname{Var}(Y|Z),\end{aligned} \tag{45}$$

where the random variable $Z = \sum_{i=1}^d i I(X \in \mathcal{A}_i)$.

One implication of Equation (45) is that if Y is not constant inside each strata, then $\operatorname{Var} \widehat{Y}_{\mathrm{PS}} > 0$, so that proportional stratification does not eliminate the variability of Y inside strata, but rather the variability of $\mathrm{E}(Y|Z)$ across strata. In addition, Equation (45), jointly with the variance decomposition formula (19), quantifies the per-replication variance reduction achieved by proportional stratification: $\mathrm{E} \operatorname{Var}(Y|Z) = \operatorname{Var} Y - \operatorname{Var} \mathrm{E}(Y|Z)$. Observe that although PS is relatively simple to implement, it does not provide the optimal sample allocation n_i per strata; see Glasserman (2004, p. 217) for more details.

From a CV perspective, PS acts like applying $\mathrm{E}(Y|Z) - \alpha$ as a CV on Y; $\widehat{Y}_{\mathrm{PS}}$ achieves the same variance reduction as that obtained by averaging i.i.d. replications of $Y - (\mathrm{E}(Y|Z) - \alpha)$. Of course, sampling from the distribution of $Y - (\mathrm{E}(Y|Z) - \alpha)$ is impractical because α is unknown.

Regarding Hilbert spaces, Equation (45) is simply

$$n \operatorname{Var} \widehat{Y}_{\mathrm{PS}} = \big\| (I - P_{\sigma(Z)})Y \big\|^2. \tag{46}$$

In addition, $\widehat{Y}_{\mathrm{PS}}$ satisfies the following CLT:

$$n^{1/2}\big(\widehat{Y}_{\mathrm{PS}} - \alpha\big) \Rightarrow \mathrm{N}\big(0, \sigma^2_{\mathrm{PS}}\big) \quad \text{as } n \to \infty,$$

where $\sigma^2_{\mathrm{PS}} = \mathrm{E} \operatorname{Var}(Y|Z)$, which enables the construction of asymptotically valid confidence intervals for α.

Post-stratification offers an alternative to proportional stratification when sampling from $\mathrm{P}(Y \in \cdot | X \in \mathcal{A}_i)$ is not possible, but when it is possible to sample from the distribution of (X, Y). Specifically, we construct the unbiased estimator

$$\widehat{Y}_{\mathrm{pST}} = \sum_{i=1}^{d} p_i \frac{\sum_{k=1}^{n} Y_k I(X_k \in \mathcal{A}_i)}{\sum_{j=1}^{n} I(X_j \in \mathcal{A}_i)}.$$

Using the Delta method (cf. Section 2) it is easy to prove a CLT for $\widehat{Y}_{\mathrm{pST}}$

$$n^{1/2}\big(\widehat{Y}_{\mathrm{pST}} - \alpha\big) \Rightarrow \mathrm{N}\big(0, \sigma^2_{\mathrm{pST}}\big)$$

as $n \to \infty$, where $\sigma^2_{\mathrm{pST}} = \mathrm{E} \operatorname{Var}(Y|Z)$. However, for every stratum we know a priori that $\mathrm{E}I(X \in \mathcal{A}_i) = p_i$, which suggests the use of the vector $(I(X \in \mathcal{A}_i) - p_i\colon 1 \leqslant i \leqslant d)$ as a control. More specifically, an unbiased CV estimator is given by

$$\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}) = \frac{1}{n} \sum_{j=1}^{n} \left(Y_j - \sum_{i=1}^{d} \lambda_i \big(I(X_j \in \mathcal{A}_i) - p_i \big) \right).$$

Using Equation (3), the optimal coefficients λ_i^*, $1 \leqslant i \leqslant d$, are immediately found to be

$$\lambda_i^* = \mathrm{E}(Y|Z = i) \quad \text{for } 1 \leqslant i \leqslant d.$$

That is,

$$\widehat{Y}_{\mathrm{CV}}\big(\boldsymbol{\lambda}^*\big) = \frac{1}{n} \sum_{j=1}^{n} \big(Y_j - \big(\mathrm{E}(Y|Z_j) - \alpha\big)\big),$$

and

$$\operatorname{Var} \widehat{Y}_{\mathrm{CV}}\big(\boldsymbol{\lambda}^*\big) = \frac{1}{n} \mathrm{E} \operatorname{Var}(Y|Z).$$

Therefore, $n(\operatorname{Var}\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*) - \operatorname{Var}\widehat{Y}_{\mathrm{pST}}) \to 0$ as $n \to \infty$. With a little more effort, it can be shown that

$$n^{1/2}\big(\widehat{Y}_{\mathrm{pST}} - \widehat{Y}_{\mathrm{CV}}\big(\boldsymbol{\lambda}^*\big)\big) \Rightarrow 0$$

as $n \to \infty$. In other words, $\widehat{Y}_{\mathrm{pST}}$ and $\hat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}^*)$ have the same distribution up to an error of order $\mathrm{o}_p(n^{-1/2})$ as $n \to \infty$, where $\mathrm{o}_p(a_n)$ denotes a sequence of random variables (ξ_n: $n \geqslant 1$) such that $a_n^{-1}\xi_n \Rightarrow 0$ as $n \to \infty$.

Given the strata ($\mathcal{A}_i$: $1 \leqslant i \leqslant d$), it is always possible to find finer strata that further reduce estimator variance. In the case of proportional stratification, suppose that it is possible to split each stratum $\mathcal{A}_i$ into integer $n_i = np_i$ strata ($\mathcal{A}_{i,k}$: $1 \leqslant k \leqslant n_i$) such that $P(X \in \mathcal{A}_{i,k}) = 1/n$; i.e., the bivariate random vector $V_n = \sum_{i=1}^{d}\sum_{k=1}^{n_i}(i,k)I(X \in \mathcal{A}_{i,k})$ is uniformly distributed on the lattice $\{(i,k)\colon 1 \leqslant i \leqslant d, 1 \leqslant k \leqslant n_i\}$. Assume in addition that it is possible to sample from $P(Y \in \cdot|X \in \mathcal{A}_{i,k})$. Then the refined proportional stratification (rST) estimator is

$$\widehat{Y}_{\mathrm{rST}} = \frac{1}{n}\sum_{i=1}^{d}\sum_{k=1}^{np_i} Y_{i,k},$$

where the $Y_{i,k}$ are sampled from $\mathrm{P}(Y \in \cdot|X \in \mathcal{A}_{i,k})$. Proceeding as in (45), we arrive at

$$\operatorname{Var}\widehat{Y}_{\mathrm{rST}} = \frac{1}{n}\operatorname{E}\operatorname{Var}(Y|V_n).$$

The fact that $\|\mathrm{E}(Y|V_n) - \mathrm{E}Y\|^2 = \|\mathrm{E}(Y|V_n) - \mathrm{E}(Y|Z)\|^2 + \|\mathrm{E}(Y|Z) - \mathrm{E}Y\|^2$ shows that $\operatorname{Var}\mathrm{E}(Y|V_n) \geqslant \operatorname{Var}\mathrm{E}(Y|Z)$, and therefore,

$$\operatorname{Var}\widehat{Y}_{\mathrm{rST}} \leqslant \operatorname{Var}\widehat{Y}_{\mathrm{PS}}.$$

With regards to Example 9, the conditions leading to Equation (28) apply, so that as $n \to \infty$,

$$\operatorname{Var}\big(\mathrm{E}(Y|X) - \mathrm{E}(Y|V_n)\big) \to 0$$

and

$$\operatorname{Var}\big(Y - \mathrm{E}(Y|V_n)\big) \to \operatorname{E}\operatorname{Var}(Y|X).$$

In particular, $n\operatorname{Var}\widehat{Y}_{\mathrm{rST}} \to \operatorname{E}\operatorname{Var}(Y|X)$. This result should come as no surprise because as n grows we get to know the full distribution of X, not unlike the setting of Equations (36) and (37): rST presumes knowledge of an increasing sequence of σ-algebras that converge to $\sigma(X)$, whereas in Equations (36) and (37) we have information about the full sequence of moments of X.

As to control variates, rST produces the same estimator variance as the standard CV estimator formed by i.i.d. sampling from $Y-(\mathrm{E}(Y|X)-\alpha)$ as $n \to \infty$. Similar to (46), we can write $n\operatorname{Var}\widehat{Y}_{\mathrm{rST}} \to \|(I-P_{\sigma(X)})Y\|^2$ as $n \to \infty$. Finally, the CLT satisfied by $\widehat{Y}_{\mathrm{rST}}$ is

$$n^{1/2}\big(\widehat{Y}_{\mathrm{rST}} - \alpha\big) \Rightarrow \mathrm{N}\big(0, \sigma^2_{\mathrm{rST}}\big)$$

as $n \to \infty$, where $\sigma^2_{\mathrm{rST}} = \mathrm{E}\,\mathrm{Var}(Y|X)$.

To conclude this section, we mention the link between post-stratification and WMC. Write

$$w^*_{k,n} = \sum_{i=1}^{d} \frac{p_i I(X_k \in \mathcal{A}_i)}{\sum_{j=1}^{n} I(X_j \in \mathcal{A}_i)} \tag{47}$$

for $1 \leqslant k \leqslant n$, then the WMC estimator $\widehat{Y}_{\mathrm{WMC}} = \sum_{k=1}^{n} w^*_{k,n} Y_k$ equals $\widehat{Y}_{\mathrm{pST}}$. It can be confirmed that the weights given in Equation (47) are the solution of the optimization problem with objective function $\min \sum_{k=1}^{n} w^2_{k,n}$ and constraints $\sum_{k=1}^{n} w_{k,n} I(X_k \in \mathcal{A}_i) = p_i$, $1 \leqslant i \leqslant d$, and $\sum_{k=1}^{n} w_{k,n} = 1$. Interpreting at face value, $\mathbf{w}^*_n$ with elements as in (47) is the closest point in the set determined by the constraints to the vector consisting of n ones.

9 Latin hypercube sampling

We now discuss the method of Latin hypercube sampling (LHS) from a Hilbert space and CV perspective. McKay et al. (1979), Stein (1987), Owen (1992) and Loh (1996) are standard references for LHS. We rely on Mathé (2000), which gives a good account of LHS from a Hilbert space point of view. Avramidis and Wilson (1996) is also a valuable reference for the issues we consider.

Suppose the setting of Examples 3 and 8: We have mutually independent random variables $X_1, \ldots, X_d$, each with known distribution function F_i, and the goal is to compute

$$\alpha = \mathrm{E} f(\mathbf{X}) = \int f(\mathbf{x})\,\mathrm{d}F(\mathbf{x})$$

via simulation, where $f : \mathbb{R}^d \to \mathbb{R}$ is a square integrable function with respect to $F(\mathbf{x}) = \prod_{d=1}^{n} F_i(x_i)$, $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{X} = (X_1, \ldots, X_d)$.

LHS generates samples of $\mathbf{X}$ as follows:

(i) Tile $[0,1)^d$ into n^d hypercubes $\Delta_{l_1,\ldots,l_d} = \prod_{i=1}^{d} [\frac{l_i - 1}{n}, \frac{l_i}{n})$, $l_i = 1, \ldots, n$, $i = 1, \ldots, d$, each of volume n^{-d}.

(ii) Generate d uniform independent permutations $(\pi_1(\cdot), \ldots, \pi_d(\cdot))$ of $\{1, \ldots, n\}$.

(iii) Use the output of (ii) to choose n hypercubes from (i): $\Delta_{\pi_1(k),\ldots,\pi_d(k)}$ for the kth tile, $k = 1, \ldots, n$.

(iv) Uniformly select a point from within each $\Delta_{\pi_1(k),\ldots,\pi_d(k)}$, and generate $X_{i,k}$ by inverting F_i at that point.

Notice that (i)–(iv) are

$$X_{i,k} = F_i^{-1}\left(\frac{\pi_i(k) - 1 + U_i(k)}{n}\right), \quad 1 \leqslant i \leqslant d \text{ and } 1 \leqslant k \leqslant n, \tag{48}$$

where the $U_i(k)$ are i.i.d. uniform on $[0, 1]$. The LHS estimator is the average of the n samples $f(\mathbf{X}_k)$, each $\mathbf{X}_k = (X_{1,k}, \ldots, X_{d,k})$ obtained according to (48)

$$\widehat{Y}_{\mathrm{LHS}} = \frac{1}{n}\sum_{k=1}^{n} f(\mathbf{X}_k).$$

As in refined stratification, given a sample size n, LHS assigns one sample to each strata $\mathcal{A}_{i,k}$ given by

$$\mathcal{A}_{i,k} = \left[F_i^{-1}\left(\frac{k-1}{n}\right), F_i^{-1}\left(\frac{k}{n}\right)\right), \quad 1 \leqslant i \leqslant d, 1 \leqslant k \leqslant n,$$

with the sample uniformly distributed within the strata. Where refined proportional stratification applied to a particular X_i asymptotically eliminates the variance due to $\mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$ (cf. Example 8 for the definition of $\mathcal{F}_i$) along just one dimension i, LHS asymptotically eliminates $\operatorname{Var}\sum_{i=1}^{d}\mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$ at the same rate used by rST used to eliminate only $\operatorname{Var}\mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$.

Figure 2 illustrates LHS with $d = 2, n = 4$. Each dot in the lower left square is a sample from $(\pi_i(k) - 1 + U_i(k))/n$. The position of each dot within a square is uniformly distributed according to the $U_i(k)$; the permutations $\pi_i(k)$ ensure that there is just one dot per row and per column. The lower right region depicts F_1 and the four equiprobable strata for X_1, with one sample point $F_1^{-1}((\pi_1(k) - 1 + U_1(k))/n)$ per strata $\mathcal{A}_{1,k}$; the final output are the samples $X_{1,1}, \ldots, X_{1,4}$. In the upper left F_2 is pictured with the axes inverted; it has the same explanation as that of F_1, with the final output being the samples $X_{2,1}, \ldots, X_{2,4}$.

Stein (1987) demonstrates that when $f \in \mathcal{L}^2(\mathrm{d}F)$,

$$\operatorname{Var}\widehat{Y}_{\mathrm{LHS}} = \frac{1}{n}\operatorname{Var}\left(f(\mathbf{X}) - \sum_{i=1}^{d}\mathrm{E}\big(f(\mathbf{X})|\mathcal{F}_i\big)\right) + \mathrm{o}\big(n^{-1}\big) \tag{49}$$

as $n \to \infty$, which makes precise the variance reduction achieved by LHS, up to order $\mathrm{o}(n^{-1})$.

The CLT satisfied by $\widehat{Y}_{\mathrm{LHS}}$, proved in Owen (1992) under the condition that $f(F^{-1}(\cdot))$ is bounded on $[0, 1]^d$ is

$$n^{1/2}\big(\widehat{Y}_{\mathrm{LHS}} - \alpha\big) \Rightarrow \mathrm{N}\big(0, \sigma_{\mathrm{LHS}}^2\big) \tag{50}$$

as $n \to \infty$, where

$$\sigma_{\mathrm{LHS}}^2 = \operatorname{Var}\left(f(\mathbf{X}) - \sum_{i=1}^{d}\mathrm{E}\big(f(\mathbf{X})|\mathcal{F}_i\big)\right).$$

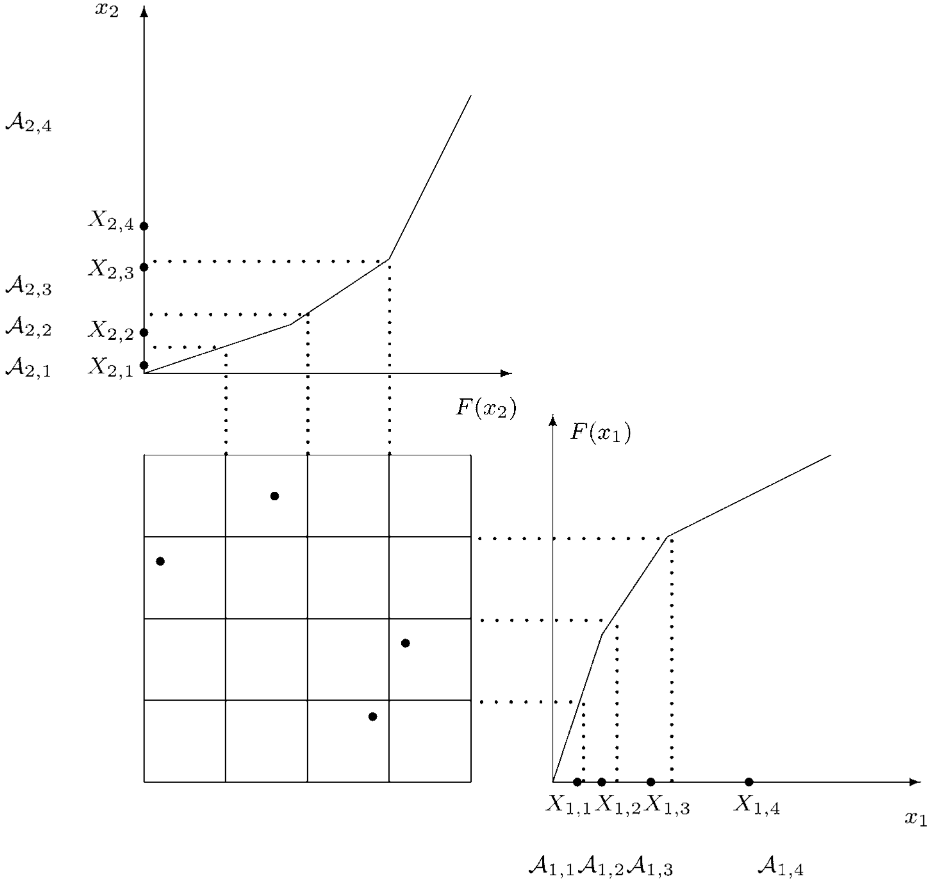


Fig. 2. Latin hypercube sampling.

Equation (50) provides theoretical support for the construction of a valid confidence interval for α, whose width depends on σ_{LHS}. This term is generally not known prior to the simulation, nor easily estimated from the simulation data. Section 3 of Owen (1992) deals with the estimation of σ^2_{LHS} from LHS data, and shows how to use the LHS samples to find an estimator for σ^2_{LHS} which is within $n^{-1/2}$ in probability of σ^2_{LHS} as $n \to \infty$. This permits the formation of an asymptotically valid confidence interval for α.

As to standard Monte Carlo, Owen (1997) proves that LHS is no less efficient because

$$\operatorname{Var} \widehat{Y}_{\mathrm{LHS}} \leqslant \frac{\operatorname{Var} f(\mathbf{X})}{n-1}$$

for all $n \geqslant 2$ and $d \geqslant 2$. In other words, even if the variance eliminated by LHS, $\operatorname{Var} \sum_{i=1}^{d} \mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$, is small, LHS with n samples is no less efficient than standard Monte Carlo with $n-1$ replications. Of course, the only measure of efficiency in this argument is variance, a more complete analysis would take into account the computational cost of generating sample variates.

Example 8 leads into the Hilbert interpretation of LHS. In particular, Equations (25) and (49) imply

$$\begin{aligned}\lim_{n\to\infty} n \operatorname{Var} \widehat{Y}_{\mathrm{LHS}} &= \operatorname{Var}\left(f(\mathbf{X}) - \sum_{i=1}^{d} \mathrm{E}\big(f(\mathbf{X})|\mathcal{F}_l\big)\right)\\ &= \big\|(I-P_M)f\big\|^2. \qquad (51)\end{aligned}$$

That is, LHS takes the remainder from projecting f on the subspace M determined by the span of the linear combinations of univariate functions. Using the last equation, LHS eliminates variance because

$$\begin{aligned}\operatorname{Var} f(\mathbf{X}) &= \big\|(I-P_0)f\big\|^2\\ &= \big\|P_M(I-P_0)f\big\|^2 + \big\|(I-P_M)(I-P_0)f\big\|^2\\ &\geqslant \big\|(I-P_M)(I-P_0)f\big\|^2\\ &= \big\|(I-P_M)f\big\|^2,\end{aligned}$$

where

$$\big\|P_M(I-P_0)f\big\|^2 = \sum_{i=1}^{d} \big\|P_i(I-P_0)f\big\|^2 = \sum_{i=1}^{d} \operatorname{Var} \mathrm{E}\big(f(\mathbf{X})|\mathcal{F}_i\big)$$

is the variance eliminated by LHS. If $f \in M$, then Equation (51) also shows that $n \operatorname{Var} \widehat{Y}_{\mathrm{LHS}} \to 0$ as $n \to \infty$; in other words, LHS asymptotically eliminates all the variance of f if f is a sum of univariate functions.

Much more can be said about $\operatorname{Var} f(\mathbf{X})$. Suppose for simplicity that the X_i are Uniform[0, 1] random variables, so that $\alpha = \int_0^1 f(\mathbf{x})\,\mathrm{d}\mathbf{x}$. Let $u \subseteq \{1, 2, \ldots, d\}$ and define $\mathrm{d}\mathbf{x}^{-u} = \prod_{j\notin u} \mathrm{d}x_j$. Then, if f is square integrable, there is a unique recursion

$$f_u(\mathbf{x}) = \int f(\mathbf{x})\,\mathrm{d}\mathbf{x}^{-u} - \sum_{v\subset u} f_v(\mathbf{x}) \qquad (52)$$

with the property that $\int_0^1 f_u(\mathbf{x})\,\mathrm{d}x_j = 0$ for every $j \in u$ and

$$f(\mathbf{x}) = \sum_{u\subseteq\{1,2,\ldots,d\}} f_u(\mathbf{x});$$

see, for example, Jiang (2003) for a proof. Recursion (52) actually is the Gram–Schmidt process, and it splits f into 2^d orthogonal components such that

$$\int f_u(\mathbf{x}) f_v(\mathbf{x})\,\mathrm{d}\mathbf{x} = 0$$

for $u \neq v$, and

$$\sigma^2 = \sum_{|u|>0} \sigma_u^2, \tag{53}$$

where $\sigma^2 = \int (f(\mathbf{x}) - \alpha)^2 \, \mathrm{d}\mathbf{x}$ is the variance of $f(\mathbf{X})$ and $\sigma_u^2 = \int f_u^2(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ is the variance of $f_u(\mathbf{X})$. The conclusion in this context is that LHS eliminates the variance of the f_u for all u with $|u| = 1$. The setting of this paragraph is known as functional ANOVA; see Chapter 13 for more details on this topic.

Considering control variates, we can say that the estimator formed by averaging i.i.d. replicates of $f(\mathbf{X}) - \sum_{i=1}^d (\mathrm{E}(f(\mathbf{X})|\mathcal{F}_i) - \alpha)$ has the same asymptotic variance as $\widehat{Y}_{\mathrm{LHS}}$. Moreover, given a zero-mean control variate $h(\mathbf{X})$, h a deterministic function, obtain n Latin hypercube samples of $\mathbf{X}$ using (48), and form the combined LHS + CV estimator

$$\widehat{Y}_{\mathrm{LHS+CV}}(\lambda) = \frac{1}{n} \sum_{k=1}^{n} \bigl(f(\mathbf{X}_k) - \lambda h(\mathbf{X}_k)\bigr).$$

Then, $n(\mathrm{Var}\, \widehat{Y}_{\mathrm{LHS+CV}}(\lambda) - \mathrm{Var}\, \widehat{Y}_{\mathrm{LHS}}) \to 0$ for any control of the type $h(\mathbf{X}) = \sum_{i=1}^d h_i(X_i)$ because $h \in M$ and property (a) of the projection operator together imply $(I - P_M)(f - \lambda h) = (I - P_M)f$ for all $\lambda \in \mathbb{R}$. Using Equations (4) and (49),

$$\mathrm{Var}\, \widehat{Y}_{\mathrm{LHS+CV}}\bigl(\lambda^*\bigr) = \mathrm{Var}\, \widehat{Y}_{\mathrm{LHS}}\bigl(1 - \rho^2\bigr),$$

where ρ^2 is the square of the correlation coefficient between

$$f(\mathbf{X}) - \sum_{i=1}^{d} \mathrm{E}\bigl(f(\mathbf{X})|\mathcal{F}_i\bigr) \quad \text{and} \quad h(\mathbf{X}) - \sum_{i=1}^{d} \mathrm{E}\bigl(h(\mathbf{X})|\mathcal{F}_i\bigr).$$

In other words, a good CV for $f(\mathbf{X})$ in the LHS context is one that maximizes the absolute value of the correlation coefficient of its nonadditive part with the nonadditive part of $f(\mathbf{X})$. Notice that λ^* is the optimal CV coefficient associated with the response $f(\mathbf{X}) - \sum_{i=1}^d \mathrm{E}(f(\mathbf{X})|\mathcal{F}_i)$ and the CV $h(\mathbf{X}) - \sum_{i=1}^d \mathrm{E}(h(\mathbf{X})|\mathcal{F}_i)$; refer to Owen (1992) for the estimation of λ^* from sample data.

Additional guidance for effective CVs in the LHS setting is provided by (53): Choose a CV with nonadditive part that is highly correlated with f_u's, $|u| > 1$, that have σ_u^2 large.

As regards weighted Monte Carlo, consider using LHS to generate replicates $\mathbf{X}_1, \ldots, \mathbf{X}_n$, and the optimization problem

$$\text{minimize} \quad \sum_{k=1}^{n} w_{k,n}^2$$

$$\text{subject to} \quad \sum_{k=1}^{n} w_{k,n} = 1, \tag{54}$$

$$\sum_{k=1}^{n} w_{k,n} I(X_{i,k} \in \mathcal{A}_{i,j}) = \frac{1}{n} \quad \text{for } i = 1, \dots, d \text{ and } j = 1, \dots, n.$$

Clearly $w^*_{k,n} = 1/n$, $k = 1, \dots, n$, is feasible for (54), and it is also optimal by the developments of Section 7, so that the WMC estimator $\sum_{k=1}^{n} w^*_{k,n} f(\mathbf{X}_k)$ coincides with $\widehat{Y}_{\mathrm{LHS}}$ for every $n \geqslant 1$. For $d = 1$, problem (54) furnishes a WMC estimator that equals $\widehat{Y}_{\mathrm{rST}}$; in other words, for $d = 1$ LHS yields the same variance reduction as rST in the limit as $n \to \infty$.

10 A numerical example

In this section we present a numerical example that supports many of the results discussed in the chapter. Consider the stochastic activity network (SAN) of Loh (1995) depicted in Figure 3 (see also the SAN discussion in Chapter 1) with arcs X_1, X_2, X_3, X_4, X_5 that are independent random variables that represent activity durations. The problem of numerically computing the expected duration of the shortest path that leads from the source node a to the sink node z involves estimating $\alpha = \mathrm{E}Y$, where $Y = \min\{X_1 + X_2, X_1 + X_3 + X_5, X_4 + X_5\}$. For the purposes of this example, we assume that the X_i's are exponentially distributed with parameters $\mu_1 = 1.1$, $\mu_2 = 2.7$, $\mu_3 = 1.1$, $\mu_4 = 2.5$, $\mu_5 = 1.2$.

Given an inner sample size n, we wish to appraise the variability of:

- The crude Monte Carlo estimator $\overline{Y}$.
- The control variates estimator $\widehat{Y}_{\mathrm{CV}}(\boldsymbol{\lambda}_n)$, using the first moments of the X_i's as control variates.
- The conditional Monte Carlo estimator $\widehat{Y}_{\mathrm{CMC}}$. Because the durations of the three paths that lead from a to z are conditionally independent given X_1 and X_5, $\mathrm{E}(Y|X_1, X_5)$ can be found analytically; see Loh (1995, p. 103).
- The weighted Monte Carlo estimator $\widehat{Y}_{\mathrm{WMC}}$, with $f(w) = -\log w$ in Equation (38).
- The stratification estimator $\widehat{Y}_{\mathrm{rST}}$, where we stratify on X_1.
- The Latin hypercube estimator $\widehat{Y}_{\mathrm{LHS}}$ applied to $X_1, \dots, X_5$.

In order to compare the variance of these estimators, we repeat $m = 1000$ times the simulation to obtain $\overline{Y}(m), \dots, \widehat{Y}_{\mathrm{LHS}}(m)$ by averaging $\overline{Y}, \dots, \widehat{Y}_{\mathrm{LHS}}$ over m. The (sample) standard deviations of these six estimators are $s(m, n)$, $s_{\mathrm{CV}}(m, n)$, $s_{\mathrm{CMC}}(m, n)$, $s_{\mathrm{WMC}}(m, n)$, $s_{\mathrm{rST}}(m, n)$ and $s_{\mathrm{LHS}}(m, n)$.

The results are summarized in Table 1. As expected, $s_{\mathrm{CV}}(m, n) \approx s_{\mathrm{WMC}}(m, n)$, and $s_{\mathrm{LHS}}(m, n) < s_{\mathrm{rST}}(m, n)$ for each n. Notice that $s_{\mathrm{rST}}(m, n)$ and

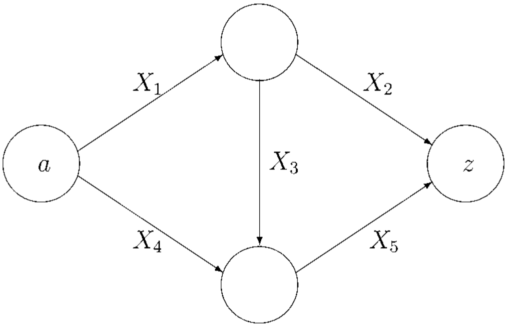


Fig. 3. Stochastic activity network.

Table 1.
SAN numerical example

Parameter	Sample size n		
	100	1000	10000
$s(m, n)$	0.0529	0.0162	0.0053
$s_{\text{CV}}(m, n)$	0.0392	0.0112	0.0036
$s_{\text{CMC}}(m, n)$	0.0328	0.0102	0.0033
$s_{\text{WMC}}(m, n)$	0.0395	0.0116	0.0038
$s_{\text{rST}}(m, n)$	0.0456	0.0144	0.0044
$s_{\text{LHS}}(m, n)$	0.03	0.0093	0.0030

$s_{\text{LHS}}(m, n)$ behave like a constant divided by $n^{1/2}$ for each n, which indicates that rST and LHS achieve their variance reduction potential by $n = 100$.

11 Conclusions

We presented various variance reduction techniques for terminating simulations in the Hilbert space setting, establishing connections with CV, CMC and WMC whenever possible. It is the geometric interpretation of Result 2 that makes this approach especially tractable.

There are, however, several topics missing from our coverage where Hilbert space theory might yield valuable insights. Consider for instance the case of variance reduction techniques for steady-state simulations that have a suitable martingale representation; see, for example, Henderson and Glynn (2002). It is well known that square integrable martingale differences have a simple interpretation in the Hilbert space framework, which suggests that it might be possible to obtain additional insights when dealing with such techniques. Another area of interest is the Hilbert space formulation of CVs in the multi-response setting, where Y is a random vector; see Rubinstein and Marcus (1985) for relevant results. The combination of importance sampling (cf. Chapter 12) with CVs also can be studied in the Hilbert space setting; see, for example, Hesterberg (1995) and Owen and Zhou (1999).

Acknowledgements

I would like to thank the editors for their valuable comments and suggestions. This work was supported by NPS Grant RIP BORY4.

References

Avellaneda, M., Gamba, R. (2000). Conquering the Greeks in Monte Carlo: Efficient calculation of the market sensitivities and hedge-ratios of financial assets by direct numerical simulation. In: Avellaneda, M. (Ed.), *Quantitative Analysis in Financial Markets, vol. III*. World Scientific, Singapore, pp. 336–356.

Avellaneda, M., Buff, R., Friedman, C., Grandchamp, N., Kruk, L., Newman, J. (2001). Weighted Monte Carlo: A new technique for calibrating asset-pricing models. *International Journal of Theoretical and Applied Finance* 4, 1–29.

Avramidis, A.N., Wilson, J.R. (1996). Integrated variance reduction techniques for simulation. *Operations Research* 44, 327–346.

Billingsley, P. (1995). *Probability and Measure*, 3rd edition. Wiley, New York.

Bollobas, B. (1990). *Linear Analysis*. Cambridge University Press, Cambridge.

Durrett, R. (1996). *Probability: Theory and Examples*, 2nd edition. Duxbury Press, Belmont.

Fishman, G.S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, New York.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glasserman, P., Yu, B. (2005). Large sample properties of weighted Monte Carlo. *Operations Research* 53 (2), 298–312.

Glasserman, P., Heidelberger, P., Shahabuddin, P. (1999). Asymptotically optimal importance sampling and stratification for path-dependent options. *Mathematical Finance* 9, 117–152.

Glynn, P.W., Szechtman, R. (2002). Some new perspectives on the method of control variates. In: Fang, K.T., Hickernell, F.J., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 27–49.

Glynn, P.W., Whitt, W. (1989). Indirect estimation via $L = \lambda W$. *Operations Research* 37, 82–103.

Henderson, S., Glynn, P.W. (2002). Approximating martingales for variance reduction in Markov process simulation. *Mathematics of Operations Research* 27, 253–271.

Hesterberg, T.C. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics* 37 (2), 185–194.

Hesterberg, T.C., Nelson, B.L. (1998). Control variates for probability and quantile estimation. *Management Science* 44, 1295–1312.

Jiang, T. (2003). Data driven shrinkage strategies for quasi-regression. PhD thesis, Stanford University.

Kreyszig, E. (1978). *Introductory Functional Analysis with Applications*. Wiley, New York.

Lavenberg, S.S., Moeller, T.L., Welch, P.D. (1982). Statistical results on control variates with application to queuing network simulation. *Operations Research* 30, 182–202.

Lavenberg, S.S., Welch, P.D. (1981). A perspective on the use of control variables to increase the efficiency of Monte Carlo simulations. *Management Science* 27, 322–335.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Loh, W.L. (1996). On Latin hypercube sampling. *The Annals of Statistics* 24 (5), 2058–2080.

Loh, W.W. (1995). On the method of control variates. PhD thesis, Stanford University.

Mathé, P. (2000). Hilbert space analysis of Latin hypercube sampling. *Proceedings of the American Mathematical Society* 129 (5), 1477–1492.

McKay, M.D., Conover, W.J., Beckman, R.J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2), 239–245.

Nelson, B.L. (1990). Control variate remedies. *Operations Research* 38, 974–992.

Owen, A.B. (1992). A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society B* 54 (2), 541–551.

Owen, A.B. (1997). Monte Carlo variance of scrambled equidistribution quadrature. *SIAM Journal of Numerical Analysis* 34 (5), 1884–1910.

Owen, A.B., Zhou, Y. (1999). Safe and effective importance sampling. Technical report, Stanford University.

Rubinstein, R.Y., Marcus, R. (1985). Efficiency of multivariate control variates in Monte Carlo simulation. *Operations Research* 33, 661–677.

Schmeiser, B.W., Taaffe, M.R., Wang, J. (2001). Biased control-variate estimation. *IIE Transactions* 33, 219–228.

Serfling, R.J. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley, New York.

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics* 29, 143–151.

Szechtman, R., Glynn, P.W. (2001). Constrained Monte Carlo and the method of control variates. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 394–400.

Venkatraman, S., Wilson, J.R. (1986). The efficiency of control variates in multiresponse simulation. *Operations Research Letters* 5, 37–42.

Williams, D. (1991). *Probability with Martingales*. Cambridge University Press, Cambridge.

Wilson, J.R. (1984). Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences* 4, 277–312.

Zimmer, R.J. (1990). *Essential Results of Functional Analysis*. The University of Chicago Press, Chicago.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13011-X

Chapter 11

Rare-Event Simulation Techniques: An Introduction and Recent Advances

S. Juneja

School of Technology and Computer Science, Tata Institute of Fundamental Research, India
E-mail: juneja@tifr.res.in

P. Shahabuddin

Department of Industrial Engineering and Operations Research, Columbia University, USA

Abstract

In this chapter we review some of the recent developments for efficient estimation of rare-events, most of which involve application of importance sampling techniques to achieve variance reduction. The zero-variance importance sampling measure is well known and in many cases has a simple representation. Though not implementable, it proves useful in selecting good and implementable importance sampling changes of measure that are in some sense close to it and thus provides a unifying framework for such selections. Specifically, we consider rare events associated with: (1) multi-dimensional light-tailed random walks, (2) with certain events involving heavy-tailed random variables and (3) queues and queueing networks. In addition, we review the recent literature on development of adaptive importance sampling techniques to quickly estimate common performance measures associated with finite-state Markov chains. We also discuss the application of rare-event simulation techniques to problems in financial engineering. The discussion in this chapter is nonmeasure theoretic and kept sufficiently simple that the key ideas are accessible to beginners. References are provided for more advanced treatments.

1 Introduction

Rare-event simulation involves estimating extremely small but important probabilities. Such probabilities are of importance in various applications: In modern packet-switched telecommunications networks, in order to reduce delay variation in carrying real-time video traffic, the buffers within the switches are of limited size. This creates the possibility of packet loss if the buffers overflow. These switches are modeled as queueing systems and it is important to estimate the extremely small loss probabilities in such queueing systems (see, e.g., Chang et al., 1994; Heidelberger, 1995). Managers of portfolios of loans need to maintain reserves to protect against rare events involving large losses due to multiple loan defaults. Thus, accurate measurement of the probability of large losses is of utmost importance to them (see, e.g., Glasserman and Li, 2005). In insurance settings, the overall wealth of the insurance company is modeled as a stochastic process. This incorporates the incoming wealth due to insurance premiums and outgoing wealth due to claims. Here the performance measures involving rare events include the probability of ruin in a given time frame or the probability of eventual ruin (see, e.g., Asmussen, 1985, 1989, 2000). In physical systems designed for a high degree of reliability, the system failure is a rare event. In such cases the related performance measures of interest include the mean time to failure, and the fraction of time the system is down or the 'system unavailability' (see, e.g., Goyal et al., 1992). In many problems in polymer statistics, population dynamics and percolation, statistical physicists need to estimate probabilities of order 10^{-50} or rarer, often to verify conjectured asymptotics of certain survival probabilities (see, e.g., Grassberger, 2002; Grassberger and Nadler, 2000).

Importance sampling is a Monte Carlo simulation variance reduction technique that has achieved dramatic results in estimating performance measures associated with certain rare events (see, e.g., Glynn and Iglehart, 1989, for an introduction). It involves simulating the system under a change of measure that accentuates paths to the rare-event and then unbiasing the resultant output from the generated path by weighing it with the 'likelihood ratio' (roughly, the ratio of the original measure and the new measure associated with the generated path). In this chapter we primarily highlight the successes achieved by this technique for estimating rare-event probabilities in a variety of stochastic systems.

We refer the reader to Heidelberger (1995) and Asmussen and Rubinstein (1995) for earlier surveys on rare-event simulation. In this chapter we supplement these surveys by focusing on the more recent developments[1]. These include a brief review of the literature on estimating rare events related to multidimensional light-tailed random walks (roughly speaking, light-tailed random variables are those whose tail distribution function decays at an exponential rate or faster, while for heavy-tailed random variables it decays at a slower rate, e.g., polynomially). These are important as many mathematical models of interest involve a complex interplay of constituent random walks, and the way rare events happen in random walks settings provides insights for the same in more complex models.

[1] The authors confess to the lack of comprehensiveness and the unavoidable bias towards their research in this survey. This is due to the usual reasons: Familiarity with this material and the desire to present the authors viewpoint on the subject.

We also briefly review the growing literature on adaptive importance sampling techniques for estimating rare events and other performance measures associated with Markov chains. Traditionally, a large part of rare-event simulation literature has focused on implementing *static* importance sampling techniques. (By static importance sampling we mean that a fixed change of measure is used throughout the simulation, while adaptive importance sampling involves updating and learning an improved change of measure based on the simulated sample paths.) Here, the change of measure is selected that emphasizes the most likely paths to the rare event. In many cases large deviations theory is useful in identifying such paths (for an introduction see, e.g., Dembo and Zeitouni, 1998; Shwartz and Weiss, 1995). Unfortunately, one can prove the effectiveness of such static importance sampling distributions only in special and often simple cases. There also exists a substantial literature highlighting cases where static importance sampling distributions with intuitively desirable properties lead to large, and even infinite, variance. In view of this, adaptive importance sampling techniques are particularly exciting as at least in the finite state Markov chain settings, they appear to be quite effective in solving a large class of problems.

Heidelberger (1995) provides an excellent review of reliability and queueing systems. In this chapter, we restrict our discussion to only a few recent developments in queueing systems.

A significant portion of our discussion focuses on the probability that a Markov process observed at a hitting time to a set lies in a rare subset. Many commonly encountered problems in rare-event simulation literature are captured in this framework. The importance sampling zero-variance estimator of small probabilities is well known, but unimplementable as it involves a priori knowledge of the probability of interest. Importantly, in this framework, the Markov process remains Markov under the zero-variance change of measure (although explicitly determining it remains at least as hard as determining the original probability of interest). This Markov representation is useful as it allows us to view the process of selecting a good importance sampling distribution from a class of easily implementable ones as identifying a distribution that is in some sense closest to the zero-variance measure. In the setting of stochastic processes involving random walks this often amounts to selecting a suitable *exponentially twisted* distribution.

We also review importance sampling techniques for rare events involving heavy-tailed random variables. This has proved to be a challenging problem in rare-event simulation and except for the simplest of cases, the important problems remain unsolved.

In addition, we review a growing literature on application of rare-event simulation techniques in financial engineering settings. These focus on efficiently estimating value-at-risk in a portfolio of investments and the probability of large losses due to credit risk in a portfolio of loans.

The following example[2] is useful in demonstrating the problem of rare-event simulation and the essential idea of importance sampling for beginners.

1.1 An illustrative example

Consider the problem of determining the probability that eighty or more heads are observed in one hundred independent tosses of a fair coin.

Although this is easily determined analytically by noting that the number of heads is binomially distributed (the probability equals 5.58×10^{-10}), this example is useful in demonstrating the problem of rare-event simulation and in giving a flavor of some solution methodologies. Through simulation, this probability may be estimated by conducting repeated experiments or trials of one hundred independent fair coin tosses using a random number generator. An experiment is said to be a success and its output is set to one if eighty or more heads are observed. Otherwise the output is set to zero. Due to the law of large numbers, an average of the outputs over a large number of independent trials gives a consistent estimate of the probability. Note that on average 1.8×10^9 trials are needed to observe one success. It is reasonable to expect that a few orders of magnitude higher number of trials are needed before the simulation estimate becomes somewhat reliable (to get a 95% confidence level of width ±5% of the probability value about 2.75×10^{12} trials are needed). This huge computational effort needed to generate a large number of trials to reliably estimate small probabilities via 'naive' simulation is the basic problem of rare-event simulation.

Importance sampling involves changing the probability dynamics of the system so that each trial gives a success with a high probability. Then, instead of setting the output to one every time a success is observed, the output is unbiased by setting it equal to the likelihood ratio of the trial or the ratio of the original probability of observing this trial with the new probability of observing the trial. The output is again set to zero if the trial does not result in a success. In the coin tossing example, suppose under the new measure the trials remain independent and the probability of heads is set to $p > 1/2$. Suppose that in a trial m heads are observed for $m \geqslant 80$. The output is then set to the likelihood ratio which equals

$$\frac{(1/2)^m (1/2)^{100-m}}{p^m (1-p)^{100-m}}. \tag{1}$$

It can be shown (see Section 2) that the average of many outputs again gives an unbiased estimator of the probability. The key issue in importance sampling is to select the new probability dynamics (e.g., p) so that the resultant output is smooth, i.e., its variance is small so that a small number of trials are needed

[2] This example and some of the discussion appeared in Juneja (2003).

to get a reliable estimate. Finding such a probability can be a difficult task requiring sophisticated analysis. A wrong selection may even lead to increase in variance compared to naive simulation.

In the coin tossing example, this variance reduction may be attained by keeping p large so that success of a trial becomes more frequent. However, if p is very close to one, the likelihood ratio on trials can have a large amount of variability. To see this, consider the extreme case when $p \approx 1$. In this case, in a trial where the number of heads equals 100, the likelihood ratio is $\approx 0.5^{100}$ whereas when the number of heads equals 80, the likelihood ratio is $\approx 0.5^{100}/(1-p)^{20}$, i.e., orders of magnitude higher. Hence, the variance of the resulting estimate is large. An in-depth analysis of this problem in Section 4 (in a general setting) shows that $p = 0.8$ gives an estimator of the probability with an enormous amount of variance reduction compared to the naive simulation estimator. Whereas trials of order 10^{12} are required under naive simulation to reliably estimate this probability, only a few thousand trials under importance sampling with $p = 0.8$ give the same reliability. More precisely, for $p = 0.8$, it can be easily numerically computed that only 7,932 trials are needed to get a 95% confidence level of width ±5% of the probability value, while interestingly, for $p = 0.99$, 3.69×10^{22} trials are needed for this accuracy.

Under the zero-variance probability measure, the output from each experiment is constant and equals the probability of interest (this is discussed further in Sections 2 and 3). Interestingly, in this example, the zero-variance measure has the property that the probability of heads after n tosses is a function of m, the number of heads observed in n tosses. Let $p_{n,m}$ denote this probability. Let $\mathrm{P}(n, m)$ denote the probability of observing at least m heads in n tosses under the original probability measure. Note that $\mathrm{P}(100, 80)$ denotes our original problem. Then, it can be seen that (see Section 3.2)

$$p_{n,m} = \frac{1}{2}\frac{\mathrm{P}(100-n-1, 80-m-1)}{\mathrm{P}(100-n, 80-m)}.$$

Numerically, it can be seen that $p_{50,40} = 0.806$, $p_{50,35} = 0.902$ and $p_{50,45} = 0.712$, suggesting that $p = 0.8$ mentioned earlier is close to the probabilities corresponding to the zero variance measure.

The structure of this chapter is as follows: In Section 2 we introduce the rare-event simulation framework and importance sampling in the abstract setting. We also discuss the zero-variance estimator and common measures of effectiveness of more implementable estimators. This discussion is specialized to a Markovian framework in Section 3. In this section we also discuss examples showing how common diverse applications fit this framework. In Section 4 we discuss effective importance sampling techniques for some rare events associated with multidimensional random walks. Adaptive importance sampling methods are discussed in Section 5. In Section 6 we discuss some recent developments in queueing systems. Heavy-tailed simulation is described in Section 7. In Section 8 we give examples of specific rare-event simulation problems in the financial engineering area and discuss the approaches that have been used. Sections 7 and 8 may be read independently of the rest of the paper as long as one has the basic background that is described in Section 2.

2 Rare-event simulation and importance sampling

2.1 Naive simulation

Consider a sample space Ω with a probability measure P. Our interest is in estimating the probability $\mathrm{P}(\mathcal{E})$ of a rare event $\mathcal{E} \subset \Omega$. Let $I(\mathcal{E})$ denote the indicator function of the event $\mathcal{E}$, i.e., it equals 1 along outcomes belonging to $\mathcal{E}$ and equals zero otherwise. Let γ denote the probability $\mathrm{P}(\mathcal{E})$. This may be estimated via naive simulation by generating independent samples $(I_1(\mathcal{E}), I_2(\mathcal{E}), \ldots, I_n(\mathcal{E}))$ of $I(\mathcal{E})$ via simulation and taking the average

$$\frac{1}{n}\sum_{i=1}^{n} I_i(\mathcal{E})$$

as an estimator of γ. Let $\hat{\gamma}_n(\mathrm{P})$ denote this estimator. The law of large numbers ensures that $\hat{\gamma}_n(\mathrm{P}) \to \gamma$ almost surely (a.s.) as $n \to \infty$.

However, as we argued in the introduction, since γ is small, most samples of $I(\mathcal{E})$ would be zero, while rarely a sample equaling one would be observed. Thus, n would have to be quite large to estimate γ reliably. The central limit theorem proves useful in developing a confidence interval (CI) for the estimate and may be used to determine the n necessary for accurate estimation. To this end, let $\sigma^2_{\mathrm{P}}(X)$ denote the variance of any random variable X simulated under the probability P. Then, for large n, an approximate $(1-\alpha)100\%$ CI for γ is given by

$$\hat{\gamma}_n(\mathrm{P}) \pm z_{\alpha/2}\frac{\sigma_{\mathrm{P}}(I(\mathcal{E}))}{\sqrt{n}},$$

where z_x is the number satisfying the relation $\mathrm{P}(\mathrm{N}(0,1) \geqslant z_x) = x$. Here, $\mathrm{N}(0,1)$ denotes a normally distributed random variable with mean zero and variance one (note that $\sigma^2_{\mathrm{P}}(I(\mathcal{E})) = \gamma(1-\gamma)$, and since $\hat{\gamma}_n(\mathrm{P}) \to \gamma$ a.s., $\sigma^2_{\mathrm{P}}(I(\mathcal{E}))$ may be estimated by $\hat{\gamma}_n(\mathrm{P})(1-\hat{\gamma}_n(\mathrm{P}))$ to give an approximate $(1-\alpha)100\%$ CI for γ).

Thus, n may be chosen so that the width of the CI, i.e., $2z_{\alpha/2}\sqrt{\gamma(1-\gamma)/n}$ is sufficiently small. More appropriately, n should be chosen so that the width of the CI relative to the quantity γ being estimated is small. For example, a confidence interval width of order 10^{-6} is not small in terms of giving an accurate estimate of γ if γ is of order 10^{-8} or less. On the other hand, it provides an excellent estimate if γ is of order 10^{-4} or more.

Thus, n is chosen so that $2z_{\alpha/2}\sqrt{(1-\gamma)/(\gamma n)}$ is sufficiently small, say within 5% (again, in practice, γ is replaced by its estimator $\hat{\gamma}_n(\mathrm{P})$, to approximately select the correct n). This implies that as $\gamma \to 0$, $n \to \infty$ to obtain a reasonable level of relative accuracy. In particular, if γ decreases at an exponential rate with respect to some system parameter b (e.g., $\gamma \approx \exp(-\theta b)$, $\theta > 0$; this may be the case for queues with light tailed service distribution where the probability of exceeding a threshold b in a busy cycle decreases at an exponential rate with b) then the computational effort n increases at an exponential rate with b to maintain a fixed level of relative accuracy. Thus, naive simulation becomes an infeasible proposition for sufficiently rare events.

2.2 *Importance sampling*

Now we discuss how importance sampling may be useful in reducing the variance of the simulation estimate and hence reducing the computational effort required to achieve a fixed degree of relative accuracy. Consider another distribution P^* with the property that $\mathrm{P}^*(A) > 0$ whenever $\mathrm{P}(A) > 0$ for $A \subset \mathcal{E}$. Then,

$$\begin{aligned}
\mathrm{P}(\mathcal{E}) &= \mathrm{E}_{\mathrm{P}}\big(I(\mathcal{E})\big) \\
&= \int I(\mathcal{E})\,\mathrm{dP} = \int I(\mathcal{E})\frac{\mathrm{dP}}{\mathrm{dP}^*}\,\mathrm{dP}^* \\
&= \int I(\mathcal{E})L\,\mathrm{dP}^* = \mathrm{E}_{\mathrm{P}^*}\big(LI(\mathcal{E})\big),
\end{aligned} \tag{2}$$

where the random variable $L = \frac{\mathrm{dP}}{\mathrm{dP}^*}$ denotes the Radon–Nikodym derivative (see, e.g., Royden, 1984) of the probability measure P with respect to P^* and is referred to as the likelihood ratio. When the state space Ω is finite or countable, $L(\omega) = \mathrm{P}(\omega)/\mathrm{P}^*(\omega)$ for each $\omega \in \Omega$ such that $\mathrm{P}^*(\omega) > 0$ and (2) equals $\sum_{\omega\in\mathcal{E}} L(\omega)\mathrm{P}^*(\omega)$ (see Section 3 for examples illustrating the form of the likelihood ratio in simple Markovian settings). This suggests the following alternative importance sampling simulation procedure for estimating γ: Generate n independent samples $(I_1(\mathcal{E}), L_1), (I_2(\mathcal{E}), L_2), \ldots, (I_n(\mathcal{E}), L_n)$ of $(I(\mathcal{E}), L)$ using P^*. Then

$$\hat{\gamma}_n\big(\mathrm{P}^*\big) = \frac{1}{n}\sum_{i=1}^{n} I_i(\mathcal{E})L_i \tag{3}$$

provides an unbiased estimator of γ.

Consider the estimator of γ in (3). Again the central limit theorem may be used to construct confidence intervals for γ. The relative width of the confidence interval is proportional to $\sigma_{\mathrm{P}^*}(LI(\mathcal{E}))/(\gamma\sqrt{n})$. The ratio of the standard deviation of an estimate to its mean is defined as the relative error. Thus, the larger the relative error of $LI(\mathcal{E})$ under P^*, the larger the sample size needed to achieve a fixed relative width of the confidence interval. In particular, the aim of importance sampling is to find a P^* that minimizes this relative error, or equivalently, the variance of the output $LI(\mathcal{E})$.

In practice, the simulation effort required to generate a sample under importance sampling is typically higher compared to naive simulation, thus the ratio of the variances does not tell the complete story. Therefore, the comparison of two estimators should be based not on the variances of each estimator, but on the product of the variance and the expected computational effort required to generate samples to form the estimator (see, e.g., Glynn and Whitt, 1992). Fortunately, in many cases the variance reduction achieved through importance sampling is so high that even if there is some increase in effort to generate a single sample, the total computational effort compared to naive simulation is still orders of magnitude less for achieving the same accuracy (see, e.g., Chang et al., 1994; Heidelberger, 1995).

Also note that in practice, the variance of the estimator is also estimated from the generated output and hence needs to be stable. Thus, the desirable P* also has a well behaved fourth moment of the estimator (see, e.g., Sadowsky, 1996; Juneja and Shahabuddin, 2002, for further discussion on this).

2.3 Zero-variance measure

Note that an estimator has zero variance if every independent sample generated always equals a constant. In such a case in every simulation run we observe $I(\mathcal{E}) = 1$ and $L = \gamma$. Thus, for $A \subset \mathcal{E}$,

$$\mathrm{P}^*(A) = \frac{\mathrm{P}(A)}{\gamma} \tag{4}$$

and $\mathrm{P}^*(A) = 0$ for $A \subset \mathcal{E}^{\mathrm{c}}$ (for any set H, H^{c} denotes its complement). The zero-variance measure is typically unimplementable as it involves knowledge of γ, the quantity that we are hoping to estimate through simulation. Nonetheless, this measure proves a useful guide in selecting a good implementable importance sampling distribution in many cases. In particular, it suggests that under a good change of measure, the most likely paths to the rare set should be given larger probability compared to the less likely ones and that the relative proportions of the probabilities assigned to the paths to the rare set should be similar to the corresponding proportions under the original measure.

Also note that the zero-variance measure is simply the conditional measure under the original probability conditioned on the occurrence of $\mathcal{E}$, i.e., (4) is equivalent to the fact that

$$\mathrm{P}^*(A) = \frac{\mathrm{P}(A \cap \mathcal{E})}{\mathrm{P}(\mathcal{E})} = \mathrm{P}(A|\mathcal{E})$$

for all events $A \in \Omega$.

2.4 Characterizing good importance sampling distributions

Intuitively, one expects that a change of measure that emphasizes the most likely paths to the rare event (assigns high probability to them) is a good one, as then the indicator function $I(\mathcal{E})$ is one with significant probability and the likelihood ratio is small along these paths as its denominator is assigned a large value. However, even a P* that has such intuitively desirable properties may lead to large and even infinite variance in practice, because on a small set in $\mathcal{E}$ the likelihood ratio may take large values, leading to a blow-up in the second moment and the variance of the estimator (see Glasserman and Kou, 1995; Glasserman and Wang, 1997; Andradottir et al., 1995; Juneja and Shahabuddin, 2001; Randhawa and Juneja, 2004). Thus, it is imperative to closely study the characteristics of good importance sampling distributions. We now discuss the different criteria for evaluating good importance sampling distributions and develop some guidelines for such selections. For this purpose we need a more concrete framework to discuss rare-event simulation.

Consider a sequence of rare events $(\mathcal{E}_b\colon b \geqslant 1)$ and associated probabilities $\gamma_b = \mathrm{P}(\mathcal{E}_b)$ indexed by a rarity parameter b such that $\gamma_b \to 0$ as $b \to \infty$. For example, in a stable single server queue setting, if $\mathcal{E}_b$ denotes the event that the queue length hits level b in a busy cycle, then we may consider the sequence $\gamma_b = \mathrm{P}(\mathcal{E}_b)$ as $b \to \infty$ (in the reliability set-up this discussion may be modified by replacing b with ε, the maximum of failure rates, and considering the sequence of probabilities γ_ε as $\varepsilon \to 0$).

Now consider a sequence of random variables $(Z_b\colon b \geqslant 1)$ such that each Z_b is an unbiased estimator of γ_b under the probability P* (this probability measure may depend upon b). The sequence of estimators $(Z_b\colon b \geqslant 1)$ is said to possess the *bounded relative error property* if

$$\limsup_{b\to\infty} \frac{\sigma_{\mathrm{P}^*}(Z_b)}{\gamma_b} < \infty.$$

It is easy to see that if the sequence of estimators possesses the bounded relative error property, then the number of samples, n, needed to guarantee a fixed relative accuracy remains bounded no matter how small the probability is, i.e., the computational effort is bounded in n for all b.

Example 1. Suppose we need to find $\gamma_b = \mathrm{P}(\mathcal{E}_b)$ for large b through importance sampling as discussed earlier. Let $Z_b = L(b)I(\mathcal{E}_b)$ denote the importance sampling estimator of γ_b under P*, where $L(b)$ denotes the associated likelihood ratio (see (2)). Further suppose that under P*:

(1) $\mathrm{P}^*(\mathcal{E}_b) \geqslant \beta > 0$ for all b.
(2) For each b, the likelihood ratio is constant over sample paths belonging to $\mathcal{E}_b$. Let k_b denote its constant value.

Then, it is easy to see that the estimators $(Z_b\colon b \geqslant 1)$ have bounded relative error. To see this, note that $\gamma_b = \mathrm{E}_{\mathrm{P}^*}(L(b)I(\mathcal{E}_b)) = k_b \mathrm{P}^*(\mathcal{E}_b)$ and $\mathrm{E}_{\mathrm{P}^*}(L(b)^2 I(\mathcal{E}_b)) = k_b^2 \mathrm{P}^*(\mathcal{E}_b)$. Recall that

$$\sigma^2_{\mathrm{P}^*}(Z_b) = \mathrm{E}_{\mathrm{P}^*}\bigl(L(b)^2 I(\mathcal{E}_b)\bigr) - \mathrm{E}_{\mathrm{P}^*}\bigl(L(b)I(\mathcal{E}_b)\bigr)^2.$$

Then

$$\frac{\sigma_{\mathrm{P}^*}(Z_b)}{\gamma_b} \leqslant \frac{\sqrt{\mathrm{E}_{\mathrm{P}^*}(L(b)^2 I(\mathcal{E}_b))}}{\gamma_b} \leqslant \frac{1}{\sqrt{\beta}}.$$

The two conditions in Example 1 provide useful insights in finding a good importance sampling distribution, although typically it is difficult to find an implementable P* that has constant likelihood ratios along sample paths to the rare set (Example 8 discusses one such case). Often one finds a distribution such that the likelihood ratios are *almost constant* (see, e.g., Siegmund, 1976; Sadowsky, 1991; Sadowsky and Szpankowski, 1995; Juneja, 2001, and the discussion in Section 4). In such and more general cases, it may be difficult to find a P* that has bounded relative error and we often settle for estimators that are efficient on a ‘logarithmic scale’. These are referred to in the literature as *asymptotically optimal* or *asymptotically efficient*. Notable exceptions where P* with bounded relative error are known include rare-event probabilities associated with certain reliability systems (see, e.g., Shahabuddin, 1994) and level crossing probabilities (see, e.g., Asmussen and Rubinstein, 1995). To understand the notion of asymptotic optimality, note that since $\sigma^2_{\mathrm{P}^*}(Z_b) \geqslant 0$ and $\gamma_b = \mathrm{E}_{\mathrm{P}^*}(Z_b)$, it follows that

$$\mathrm{E}_{\mathrm{P}^*}\left(Z_b^2\right) \geqslant \gamma_b^2,$$

and hence $\log(\mathrm{E}_{\mathrm{P}^*}(Z_b^2)) \geqslant 2\log(\gamma_b)$. Since $\log(\gamma_b) < 0$, it follows that

$$\frac{\log(\mathrm{E}_{\mathrm{P}^*}(Z_b^2))}{\log(\gamma_b)} \leqslant 2$$

for all b and for all P*. The sequence of estimators are said to be asymptotically optimal if the above relation holds as an equality in the limit as $b \to \infty$. For example, suppose that $\gamma_b = \mathrm{P}_1(b)\exp(-cb)$ and $\mathrm{E}_{\mathrm{P}^*}(Z_b^2) = \mathrm{P}_2(b)\exp(-2cb)$ where $c > 0$, and $\mathrm{P}_1(\cdot)$ and $\mathrm{P}_2(\cdot)$ are any two polynomial functions of b (of course, $\mathrm{P}_2(b) \geqslant \mathrm{P}_1(b)^2$). The measure P* may be asymptotically optimal, although we may not have bounded relative error.

2.4.1 Uniformly bounded likelihood ratios

In many settings, one can identify a change of measure where the associated likelihood ratio is uniformly bounded along paths to the rare set $\mathcal{E}$ (the subscript b is dropped as we again focus on a single set) by a small constant $k < 1$, i.e.,

$$LI(\mathcal{E}) \leqslant kI(\mathcal{E}).$$

This turns out to be a desirable trait. Note that $\mathrm{E}_{\mathrm{P}^*}(L^2 I(\mathcal{E})) = \mathrm{E}_{\mathrm{P}}(LI(\mathcal{E}))$. Thus,

$$\frac{\sigma^2_{\mathrm{P}^*}(L(I(\mathcal{E}))}{\sigma^2_{\mathrm{P}}(I(\mathcal{E}))} = \frac{\mathrm{E}_{\mathrm{P}}(L(I(\mathcal{E})) - \gamma^2}{\gamma - \gamma^2} \leqslant \frac{k\gamma - \gamma^2}{\gamma - \gamma^2} \leqslant k. \tag{5}$$

Thus, guaranteed variance reduction by at least a factor of k is achieved. Often, a parameterized family of importance sampling distributions can be identified so that the likelihood ratio associated with each distribution in this family is uniformly bounded along paths to the rare set by a constant that may depend on the distribution. Then, a good importance sampling distribution from this family may be selected as the one with the minimum uniform bound. For instance, in the example considered in Section 1.1, it can be seen that the likelihood ratio in (1) is upper bounded by

$$\frac{(1/2)^{100}}{p^{80}(1-p)^{20}}$$

for each $p \geqslant 1/2$ when the experiment is a success, i.e., the number of heads n is greater than or equal to 80 (also see Section 4). Note that this bound is minimized for $p = 0.8$.

In some cases, we may be able to partition the rare event of interest $\mathcal{E}$ into disjoint sets $\mathcal{E}_1, \ldots, \mathcal{E}_J$ such that there exist probability measures $(\mathrm{P}_j^*\colon\ j \leqslant J)$ such that the likelihood ratio $L^{(j)}$ corresponding to each probability measure P_j^* satisfies the relation

$$L^{(j)} \leqslant k_j$$

for a constant $k_j \ll 1$ on the set $\mathcal{E}_j$ (although, the likelihood ratio may be unbounded on other sets). One option then may be to estimate each $\mathrm{P}(\mathcal{E}_j)$ separately using the appropriate change of measure. Sadowsky and Bucklew (1990) propose that a convex combination of these measures may work in estimating $\mathrm{P}(\mathcal{E})$. To see this, let $(p_j\colon\ j \leqslant J)$ denote positive numbers that sum to one, and consider the measure

$$\mathrm{P}^*(\cdot) = \sum_{j \leqslant J} p_j \mathrm{P}_j^*(\cdot).$$

It is easy to see that the likelihood ratio of P w.r.t. P^*, then equals

$$\frac{1}{\sum_{j \leqslant J} p_j / L^{(j)}} \leqslant \max_{j \leqslant J} \frac{k_j}{p_j},$$

so that if the right-hand side is smaller than 1 (which is the case, e.g., if p_j is proportional to k_j and $\sum_{j \leqslant J} k_j < 1$) guaranteed variance reduction may be achieved.

In some cases, under the proposed change of measure, the uniform upper bound on the likelihood ratio is achieved on a substantial part of the rare set and through analysis it is shown that the remaining set has very small probability, so that even large likelihood ratios on this set contribute little to the variance of the estimator (see, e.g., Juneja and Shahabuddin, 2002). This remaining set may be asymptotically negligible so that outputs from it may be ignored (see, e.g., Boots and Shahabuddin, 2001) introducing an asymptotically negligible bias.

3 Rare-event simulation in a Markovian framework

We now specialize our discussion to certain rare events associated with discrete time Markov processes. This framework captures many commonly studied rare events in the literature including those discussed in Sections 4–7.

Consider a Markov process $(S_i\colon\ i \geqslant 0)$ where each S_i takes values in space $\mathcal{S}$ (e.g., $\mathcal{S} = \Re^d$). Often, in rare-event simulation we want to determine the small probability of an event $\mathcal{E}$ determined by the Markov process observed up to a stopping time T, i.e., $(S_0, S_1, \ldots, S_T)$. A random variable (r.v.) T is a stopping time w.r.t. the stochastic process $(S_i\colon\ i \geqslant 0)$ if for any nonnegative integer n, whether $\{T = n\}$ occurs or not can be completely determined by observing $(S_0, S_1, S_2, \ldots, S_n)$. In many cases we may be interested in the probability of a more specialized event $\mathcal{E} = \{S_T \in \mathcal{R}\}$, where $\mathcal{R} \subset \mathcal{S}$ and T denotes the hitting time to a 'terminal' set $\mathcal{T}$, $\mathcal{R} \subset \mathcal{T}$, i.e., $T = \inf\{n\colon\ S_n \in \mathcal{T}\}$. In many cases, the rare-event probability of interest may be reduced to $\mathrm{P}(S_T \in \mathcal{R})$ through state-space augmentation; the latter representation has the advantage that the zero-variance estimator is Markov for this probability. Also, as we discuss in Examples 5 and 6, in a common application, the stopping time under consideration is infinite with large probability and our interest is in estimating $\mathrm{P}(T < \infty)$.

Example 2. The coin tossing example discussed in the introduction fits this framework by setting $T = 100$ and letting $(X_i\colon\ i \geqslant 1)$ be a sequence of i.i.d. random variables where each X_i equals one with probability 0.5 and zero with probability 0.5. Here, $\mathcal{E} = \{\sum_{i=1}^{100} X_i \geqslant 80\}$. Alternatively, let S_n denote the vector $(\sum_{i=1}^{n} X_i, n)$. Let $\mathcal{T}$ denote the event $\{(x, 100)\colon\ x \geqslant 0\}$, $T = \inf\{n\colon\ S_n \in \mathcal{T}\}$ and let $\mathcal{R} = \{(x, 100)\colon\ x \geqslant 80\}$. Then the probability of interest equals $\mathrm{P}(S_T \in \mathcal{R})$.

Note that a similar representation may be obtained more generally for the case where $(X_i\colon\ i \geqslant 1)$ is a sequence of generally distributed i.i.d. random variables, and our interest is in estimating the probability $\mathrm{P}(S_n/n \in \mathcal{R})$ for $\mathcal{R}$ that does not include $\mathrm{E}X_i$ in its closure.

Example 3. The problem of estimating the small probability that the queue length in a stable $M/M/1$ queue hits a large threshold b in a busy cycle (a busy cycle is the stochastic process between the two consecutive times that an arrival to the system finds it empty), fits this framework as follows: Let λ denote the arrival rate to the queue and let μ denote the service rate. Let $p = \lambda/(\lambda + \mu)$. Let S_i denote the queue length after the ith state change (due to an arrival or a departure). Clearly $(S_n\colon\ n \geqslant 0)$ is a Markov process. To denote that the busy cycle starts with one customer we set $S_0 = 1$. If $S_i > 0$, then $S_{i+1} = S_i + 1$ with probability p and $S_{i+1} = S_i - 1$ with probability $1 - p$. Let $T = \inf\{n\colon\ S_n = b \text{ or } S_n = 0\}$. Then $\mathcal{R} = \{b\}$ and the probability of interest equals $\mathrm{P}(S_T \in \mathcal{R})$.

Example 4. The problem of estimating the small probability that the queue length in a stable $GI/GI/1$ queue hits a large threshold b in a busy cycle is important from an applications viewpoint. For instance, Chang et al. (1994) and Heidelberger (1995) discuss how techniques for efficient estimation of this probability may be used to efficiently estimate the steady state probability of buffer overflow in finite-buffer single queues. This probability also fits in our framework, although we need to keep in mind that the queue length process observed at state change instants is no longer Markov and additional variables are needed to ensure the Markov property. Here, we assume that the arrivals and the departures do not occur in batches of two or more. Let $(Q_i\colon i \geqslant 0)$ denote the queue-length process observed *just before* the time of state change (due to arrivals or departures). Let J_i equal 1 if the ith state change is due to an arrival. Let it equal 0, if it is due to a departure. Let R_i denote the remaining service time of the customer in service if $J_i = 1$ and $Q_i > 0$. Let it denote the remaining interarrival time if $J_i = 0$. Let it equal zero if $J_i = 1$ and $Q_i = 0$. Then, setting $S_i = (Q_i, J_i, R_i)$, it is easy to see that $(S_i\colon i \geqslant 0)$ is a Markov process. Let $T = \inf\{n\colon (Q_i, J_i) = (b, 1) \text{ or } (Q_i, J_i) = (1, 0)\}$. Then $\mathcal{R} = \{(b, 1, x)\colon x \geqslant 0\}$ and the probability of interest equals $\mathrm{P}(S_T \in \mathcal{R})$.

Example 5. Another problem of importance concerning small probabilities in a $GI/GI/1$ queue setting with first-come-first-serve scheduling rule involves estimation of the probability of large delays in the queue in steady state. Suppose that the zeroth customer arrives to an empty queue and that $(A_0, A_1, A_2, \ldots)$ denotes a sequence of i.i.d. nonnegative r.v.'s where A_n denotes the interarrival time between customer n and $n + 1$. Similarly, let $(B_0, B_1, \ldots)$ denote the i.i.d. sequence of service times in the queue so that the service of customer n is denoted by B_n. Let W_n denote the waiting time of customer n in the queue. Then $W_0 = 0$. The well-known Lindley recursion follows:

$$W_{n+1} = \max(W_n + B_n - A_n, 0)$$

for $n \geqslant 0$ (see, e.g., Asmussen, 2003). We assume that $\mathrm{E}(B_n) < \mathrm{E}(A_n)$, so that the queue is stable and the steady state waiting time distribution exists. Let $Y_n = B_n - A_n$. Then, since $W_0 = 0$, it follows that

$$W_{n+1} = \max(0, Y_n, Y_n + Y_{n-1}, \ldots, Y_n + Y_{n-1} + \cdots + Y_0).$$

Since the sequence $(Y_i\colon i \geqslant 0)$ is i.i.d., the right-hand side has the same distribution as

$$\max(0, Y_0, Y_0 + Y_1, \ldots, Y_0 + Y_1 + \cdots + Y_n).$$

In particular, the steady-state delay probability $\mathrm{P}(W_\infty > u)$ equals $\mathrm{P}(\exists n\colon \sum_{i=0}^n Y_i > u)$. Let $S_n = \sum_{i=0}^n Y_i$ denote the associated random walk with a negative drift. Let $T = \inf\{n\colon S_n > u\}$ so that T is a stopping time w.r.t.

(S_i: $i \geqslant 0$). Then $\mathrm{P}(W_\infty > u)$ equals $\mathrm{P}(T < \infty)$. The latter probability is referred to as the level-crossing probability of a random walk. Again, we need to generate $(S_0, S_1, \dots, S_T)$ to determine whether the event $\{T < \infty\}$ occurs or not. However, we now have an additional complexity that $\mathrm{P}(T = \infty) > 0$ and hence generating $(S_0, S_1, \dots, S_T)$ may no longer be feasible. Importance sampling resolves this by simulating under a suitable change of measure P^* under which the random walk has a positive drift so that $\mathrm{P}^*(T = \infty) = 0$ (see Siegmund, 1976). This is also discussed in Section 4 in a multidimensional setting when the X_i's have a light-tailed distribution.

Example 6. The problem of estimating ruin probabilities in the insurance sector also fits this framework as follows: Suppose that an insurance company accumulates premiums at a deterministic rate p. Further suppose that the claim interarrival times are an i.i.d. sequence of r.v.'s $(A_1, A_2, \dots)$. Let $N(t) = \sup\{n\colon \sum_{i=1}^n A_i \leqslant t\}$ denote the number of claims that have arrived by time t. Also, assume that the claim sizes are again another i.i.d. sequence of r.v.'s $(B_1, B_2, \dots)$ independent of the interarrival times (these may be modeled using light or heavy-tailed distributions). Let the initial reserves of the company be denoted by u. In such a model, the wealth of the company at time t is denoted by

$$W(t) = u + pt - \sum_{i=1}^{N(t)} B_i.$$

The probability of eventual ruin therefore equals $\mathrm{P}(\inf_t W(t) \leqslant 0)$. Note that a ruin can occur only at the times of claim arrivals. The wealth at the time of arrival of claim n equals

$$W\left(\sum_{i=1}^n A_i\right) = u + p\sum_{i=1}^n A_i - \sum_{i=1}^n B_i.$$

Let $Y_i = B_i - pA_i$ and $S_n = \sum_{i=1}^n Y_i$. The probability of eventual ruin then equals $\mathrm{P}(\max_n S_n > u)$ or equivalently $\mathrm{P}(T < \infty)$, where $T = \inf\{n\colon S_n > u\}$. Hence, the discussion at the end of Example 5 applies here as well.

Example 7 (Highly reliable Markovian systems). These reliability systems have components that fail and repair in a Markovian manner, i.e., they have exponentially distributed failure and repair times. High reliability is achieved due to the highly reliable nature of the individual components comprising the system. Complex system interdependencies may be easily modeled in the Markov framework. These interdependencies may include failure propagation, i.e., failure of one component with certain probability leads to failure of other components. They may also include other features such as different modes of component failure, repair and operational dependencies, component switch-over times, etc. See, e.g., Goyal and Lavenberg (1987) and Goyal et al. (1992) for further discussion on such modeling complexities.

A mathematical model for such a system may be built as follows: Suppose that the system has d distinct component-types. Each component type i has m_i identical components for functional and spare requirements. Let λ_i and μ_i denote the failure and repair rate, respectively, for each of these components. The fact that each component is highly reliable is modeled by letting $\lambda_i = \Theta(\varepsilon^{r_i})$[3] for $r_i \geqslant 1$, and letting $\mu_i = \Theta(1)$. The system is then analyzed as $\varepsilon \to 0$.

Let $(Y(t)\colon t \geqslant 0)$ be a continuous time Markov chain (CTMC) of this system, where $Y(t) = (Y_1(t), Y_2(t), \ldots, Y_d(t), R(t))$. Here, each $Y_i(t)$ denotes the number of failed components of type i at time t. The vector $R(t)$ contains all configurational information required to make $(Y(t)\colon t \geqslant 0)$ a Markov process. For example, it may contain information regarding the order in which the repairs occur, the failure mode of each component, etc. Let $\mathcal{A}$ denote the state when all components are 'up' (let it also denote the set containing this state). Let $\mathcal{R}$ denote the set of states deemed as failed states. This may be a rare set for small values of ε. The probability that the system starting from state $\mathcal{A}$, hits the set $\mathcal{R}$ before returning to $\mathcal{A}$ is important for these highly reliable systems as this is critical to efficient estimation of performance measures such as system unavailability and mean time to failure. Let $(S_i\colon i \geqslant 0)$ denote the discrete time Markov chain (DTMC) embedded in $(Y(t)\colon t \geqslant 0)$. For estimating this probability, the DTMC may be simulated instead of the CTMC as both give identical results. Set $S_0 = \mathcal{A}$. Then, the process $(S_1, \ldots, S_T)$ may be observed where $T = \inf\{n \geqslant 1\colon S_n \in \mathcal{T}\}$, where $\mathcal{T} = \mathcal{A} \cup \mathcal{R}$. The set $\mathcal{E}$ equals $\{S_T \in \mathcal{R}\}$.

In this chapter we do not pursue highly reliable systems further. Instead we refer the reader to Heidelberger (1995) and Nakayama et al. (2001) for surveys on this topic.

3.1 Importance sampling in a Markovian framework

Let P_n denote the probability P restricted to the events associated with $(S_0, S_1, \ldots, S_n)$ for $n = 1, 2, \ldots$. Then

$$\gamma := \mathrm{P}(\mathcal{E}) = \sum_n \mathrm{P}_n(\mathcal{E}_n),$$

where $\mathcal{E}_n = \mathcal{E} \cap \{T = n\}$. Consider another distribution P^* and let P^*_n denote its restriction to the events associated with $(S_0, S_1, \ldots, S_n)$ for $n = 1, 2, \ldots$. Suppose that for each n, $\mathrm{P}^*_n(A_n) > 0$ whenever $\mathrm{P}_n(A_n) > 0$ for $A_n \subset \mathcal{E}_n$.

[3] A nonnegative function $f(\varepsilon)$ is said to be $\mathrm{O}(\varepsilon^r)$ for $r \geqslant 0$ if there exists a positive constant K such that $f(\varepsilon) \leqslant K\varepsilon^r$ for all ε sufficiently small. It is said to be $\Theta(\varepsilon^r)$ for $r \geqslant 0$ if there exist positive constants K_1 and K_2 $(K_1 < K_2)$, such that $K_1\varepsilon^r \leqslant f(\varepsilon) \leqslant K_2\varepsilon^r$ for all ε sufficiently small.

Then, proceeding as in (2),

$$\mathrm{P}(\mathcal{E}) = \sum_n \int_{\mathcal{E}_n} L_n \, \mathrm{d}P_n^*,$$

where $L_n = \frac{\mathrm{dP}_n}{\mathrm{dP}_n^*}$. For example, if the sequence $(S_0, S_1, \ldots, S_n)$ has a density function $f_n(\cdot)$ for each n under P ($f_n^*(\cdot)$ under P^*) such that $f_n^*(x_0, x_1, \ldots, x_n) > 0$ whenever $f_n(x_0, x_1, \ldots, x_n) > 0$, then

$$L_n(S_0, S_1, \ldots, S_n) = \frac{f_n(S_0, S_1, \ldots, S_n)}{f_n^*(S_0, S_1, \ldots, S_n)} \tag{6}$$

for each n a.s.

Thus, $\gamma = \mathrm{E}_{\mathrm{P}^*}(L_T I(\mathcal{E}))$ where $\mathrm{E}_{\mathrm{P}^*}$ is an expectation operator under the probability P^*. To further clarify the discussion, we illustrate the form of the likelihood ratio for Examples 3 and 4.

Example 8. In Example 3, suppose the queue is simulated under a probability P^* under which it is again an $M/M/1$ queue with arrival rate λ^* and service rate μ^*. Let $p^* = \lambda^*/(\lambda^* + \mu^*)$. Consider a sample path $(S_0, S_1, \ldots, S_T)$ that belongs to $\mathcal{E}$, i.e., $\{S_T \in \mathcal{R}\}$. Let N_A denote the number of arrivals and N_S denote the number of service completions up to time T along this sample path. Thus, $N_A = b + N_S - 1$ where b denotes the buffer size. The likelihood ratio L_T along this path therefore equals

$$\left(\frac{p}{p^*}\right)^{N_A} \left(\frac{1-p}{1-p^*}\right)^{N_S}.$$

In the case $\lambda < \mu$, it can be seen that $\lambda^* = \mu$ and $\mu^* = \lambda$ achieves the two conditions discussed in Example 1 (with $k_b = (\lambda/\mu)^{b-1}$) and hence the associated importance sampling distribution has the bounded relative error property.

Example 9. In Example 4, let $f(\cdot)$ and $g(\cdot)$ denote the p.d.f. of the interarrival times and the service times, respectively under the probability P. Let P^* be another probability under which the queue remains a $GI/GI/1$ queue with the new p.d.f.'s for interarrival and service times denoted by $f^*(\cdot)$ and $g^*(\cdot)$, respectively. Consider a sample path $(S_0, S_1, \ldots, S_T)$ that belongs to $\mathcal{E}$, i.e., $\{Q_T = b\}$. Let N_A denote the number of arrivals and N_B denote the number of service initiations up to time T along this sample path. Let $(A_1, A_2, \ldots, A_{N_A})$ denote the N_A interarrival times generated and let $(B_1, B_2, \ldots, B_{N_B})$ denote the N_B service times generated along this sample path. The likelihood ratio L_T along this path therefore equals

$$\prod_{i=1}^{N_A} \frac{f(A_i)}{f^*(A_i)} \prod_{i=1}^{N_B} \frac{g(B_i)}{g^*(B_i)}.$$

Thus, from the simulation viewpoint the computation of the likelihood ratio in Markovian settings is straightforward and may be done iteratively as follows: Before generation of a sample path of $(S_0, S_1, \ldots, S_T)$ under the new probability, the likelihood ratio may be initialized to 1. Then, it may be updated at each transition by multiplying it with the ratio of the original probability density function of the newly generated sample(s) at that transition and the new probability density function of this sample(s). The probability density function may be replaced by the probability values when discrete random variables are involved.

3.2 *Zero-variance measure in Markovian settings*

For probabilities such as $\mathrm{P}(S_T \in \mathcal{R})$, the zero-variance measure has a Markovian representation. For $\mathcal{E} = \{S_T \in \mathcal{R}\}$, let $\mathrm{P}_x(\mathcal{E})$ denote the probability of this event, conditioned on $S_0 = x$. Recall that $T = \inf\{n\colon S_n \in \mathcal{T}\}$. For simplicity suppose that the state space $\mathcal{S}$ of the Markov chain is finite (the following discussion is easily extended to more general state spaces) and let $P = (p_{xy}\colon x, y \in \mathcal{S})$ denote the associated transition matrix. In this setting,

$$\mathrm{P}_x(\mathcal{E}) = \sum_{y \in \mathcal{R}} p_{xy} + \sum_{y \in \mathcal{S} - \mathcal{T}} p_{xy} \mathrm{P}_y(\mathcal{E}).$$

Thus, $p^*_{xy} = p_{xy}/\mathrm{P}_x(\mathcal{E})$ for $y \in \mathcal{R}$ and $p^*_{xy} = p_{xy}\mathrm{P}_y(\mathcal{E})/\mathrm{P}_x(\mathcal{E})$ for $y \in \mathcal{S} - \mathcal{T}$ is a valid transition probability. It is easy to check that in this case

$$L_T = \frac{p_{S_0,S_1} p_{S_1,S_2} \cdots p_{S_{T-1},S_T}}{p^*_{S_0,S_1} p^*_{S_1,S_2} \cdots p^*_{S_{T-1},S_T}}$$

equals $\mathrm{P}_{S_0}(\mathcal{E})$ a.s., i.e., the associated P^* is the zero-variance measure. The problem again is that determining p^*_{xy} requires knowledge of $\mathrm{P}_x(\mathcal{E})$ for all $x \in \mathcal{S}$.

Consider the probability $\mathrm{P}(S_n/n \geqslant a)$, where $S_n = \sum_{i \leqslant n} X_i$, the $(X_i\colon i \geqslant 0)$ are i.i.d. r.v.'s taking values in $\Re$, and $a > \mathrm{E}X_i$. From the above discussion and using the associated augmented Markov chain discussed at the end of Example 2, it can be seen that the zero-variance measure conditioned on the event that $S_m = s_m < na$, $m < n$, has transition probabilities

$$p^*_{m,s_m}(y) = \mathrm{P}(X_{m+1} = y) \frac{\mathrm{P}(S_n \geqslant na | S_{m+1} = s_m + y)}{\mathrm{P}(S_n \geqslant na | S_m = s_m)}.$$

More generally,

$$\begin{aligned} &\mathrm{P}^*(X_{m+1} \in \mathrm{d}y | S_m = s_m) \\ &\quad = \mathrm{P}(X_{m+1} \in \mathrm{d}y) \frac{\mathrm{P}(S_{n-m-1} \geqslant na - s_m - y)}{\mathrm{P}(S_{n-m} \geqslant na - s_m)}. \end{aligned} \tag{7}$$

Such an explicit representation of the zero-variance measure proves useful in adaptive algorithms where one adaptively learns the zero-variance measure (see Section 5). This representation is also useful in developing simpler implementable importance sampling distributions that are in an asymptotic sense close to this measure (see Section 3.3).

3.3 Exponentially twisted distributions

Again consider the probability $\mathrm{P}(S_n/n \geqslant a)$. Let $\Psi(\cdot)$ denote the log-moment generating function of X_i, i.e., $\Psi(\theta) = \log \mathrm{E}(\exp(\theta X_i))$. Let $\Theta = \{\theta : \Psi(\theta) < \infty\}$. Suppose that Θ^{o} (for any set H, H^{o} denotes its interior) contains the origin, so that X_i has a light-tailed distribution. For $\theta \in \Theta^{\mathrm{o}}$, consider the probability P_θ under which the $(X_i\colon\ i \geqslant 1)$ are i.i.d. and

$$\mathrm{P}_\theta(X_i \in \mathrm{d}y) = \exp\bigl(\theta y - \Psi(\theta)\bigr)\,\mathrm{P}(X_i \in \mathrm{d}y).$$

This is referred to as the probability obtained by exponentially twisting the original probability by θ. We now show that the distribution of X_{m+1} conditioned on $S_m = s_m$ under the zero-variance measure for the probability $\mathrm{P}(S_n/n \geqslant a)$ (shown in (7)) converges asymptotically (as $n \to \infty$) to a suitable exponentially twisted distribution independent of s_m, thus motivating the use of such distributions for importance sampling of constituent r.v.'s in random walks in complex stochastic processes.

Suppose that $\theta_a \in \Theta^{\mathrm{o}}$ solves the equation $\Psi'(\theta) = a$. In that case, when the distribution of X_i is nonlattice, the following exact asymptotic is well known (see Bahadur and Rao, 1960; Dembo and Zeitouni, 1998):

$$\mathrm{P}\biggl(\frac{S_n}{n} \geqslant a + \frac{k}{n} + \mathrm{o}\biggl(\frac{1}{n}\biggr)\biggr) \sim \frac{c}{\sqrt{n}} \exp\bigl[-n\bigl(\theta_a a - \Psi(\theta_a)\bigr) - \theta_a k\bigr], \tag{8}$$

where $c = 1/(\sqrt{2\pi\Psi''(\theta_a)}\theta_a)$ ($a_n \sim b_n$ means that $a_n/b_n \to 1$ as $n \to \infty$) and k is a constant. Usually, the exact asymptotic is developed for $\mathrm{P}(S_n/n \geqslant a)$. The minor generalization in (8) is discussed, e.g., in Borkar et al. (2004). This exact asymptotic may be inaccurate if n is not large enough especially for certain sufficiently 'nonnormal' distributions of X_i. In such cases, simulation using importance sampling may be a desirable option to get accurate estimates.

Using (8) in (7) as $n \to \infty$, for a fixed m, it can be easily seen that

$$\lim_{n\to\infty} \mathrm{P}^*(X_{m+1} \in \mathrm{d}y | S_m = s_m) = \mathrm{P}(X_{m+1} \in \mathrm{d}y) \exp\bigl(\theta_a y - \Psi(\theta_a)\bigr),$$

i.e., asymptotically the zero-variance measure converges to P_{θ_a}. This suggests that P_{θ_a} may be a good importance sampling distribution to estimate $\mathrm{P}(S_n/n \geqslant a)$ for large n. We discuss this further in Section 4. Also, it is easily seen through differentiation that the mean of X_i under P_θ equals $\Psi'(\theta)$. In particular, under P_{θ_a}, the mean of X_i equals a, so that $\{S_n/n \geqslant a\}$ is no longer a rare event.

4 Large deviations of multidimensional random walks

In this section we focus on efficient estimation techniques for two rare-event probabilities associated with multidimensional random walks, namely: (1) the probability that the random walk observed after a large time period n, lies in a rare set; (2) the probability that the random walk ever hits a rare set. We provide a heuristic justification for the large deviations asymptotic in the two cases and identify the asymptotically optimal changes of measures. We note that the ideas discussed earlier greatly simplify the process of identifying a good change of measure. These include restricting the search for the change of measure to those obtained by exponentially twisting the original measure, selecting those that have constant (or almost constant) likelihood ratios along paths to the rare set, or selecting those whose likelihood ratios along such paths have the smallest uniform bound.

4.1 Random walk in a rare set

Consider the probability $\mathrm{P}(S_n/n \in \mathcal{R})$, where $S_n = \sum_{i=1}^{n} X_i$, the X_i's are i.i.d. and each X_i is a random column vector taking values in $\Re^d$. Thus, $X_i = (X_{i1}, \dots, X_{id})^{\mathrm{T}}$ where the superscript "T" denotes the transpose operation. The set $\mathcal{R} \subset \Re^d$ and its closure does not include $\mathrm{E}X_i$. The essential ideas for this discussion are taken from Sadowsky and Bucklew (1990) (also see Sadowsky, 1996) where this problem is studied in a more general framework. We refer the reader to these references for rigorous analysis, while the discussion here is limited to illustrating the key intuitive ideas in a simple setting.

For simplicity suppose that the log moment generating function,

$$\Psi(\theta) = \log \mathrm{E}\bigl(\exp\bigl(\theta^{\mathrm{T}} X\bigr)\bigr),$$

exists for each column vector $\theta \in \Re^d$. This is true, e.g., when X_i is bounded or has a multivariate Gaussian distribution. Further suppose that X_i is nondegenerate, i.e., it is not a.s. constant in any dimension. Define the associated rate function

$$J(\alpha) = \sup_{\theta}\bigl(\theta^{\mathrm{T}}\alpha - \Psi(\theta)\bigr)$$

for $\alpha \in \Re^d$. Note that for each θ, $\theta^{\mathrm{T}}\alpha - \Psi(\theta)$ is a convex function of α, hence, $J(\cdot)$ being a supremum of convex functions, is again convex. It can be shown that it is strictly convex in the interior $\mathcal{J}^{\mathrm{o}}$, where

$$\mathcal{J} = \bigl\{\alpha\colon J(\alpha) < \infty\bigr\}.$$

From large deviations theory (see, e.g., Dembo and Zeitouni, 1998), we see that

$$\mathrm{P}\left(\frac{S_n}{n} \approx a\right) \approx \exp\bigl(-nJ(a)\bigr). \tag{9}$$

Here, $S_n/n \approx a$ may be taken to be the event that S_n/n lies in a small ball of radius ε centered at a. The relation (9) becomes an equality when an appropriate $\mathrm{O}(\varepsilon)$ term is added to $J(a)$ in the exponent in the right-hand side. It is instructive to heuristically see this result. Note that

$$\mathrm{P}\left(\frac{S_n}{n} \approx a\right) = \int_{x \approx na} \mathrm{d}F_n(x),$$

where $F_n(\cdot)$ denotes the distribution function (d.f.) of S_n (obtained by convolution of the d.f. of X_i n times). Let $F_\theta(\cdot)$ denote the d.f. obtained by exponentially twisting $F(\cdot)$ by θ, i.e.,

$$\mathrm{d}F_\theta(x) = \exp\bigl(\theta^{\mathrm{T}} x - \Psi(\theta)\bigr)\, \mathrm{d}F(x).$$

It follows (heuristically speaking) that

$$\mathrm{P}\left(\frac{S_n}{n} \approx a\right) \approx \exp\bigl[-n\bigl(\theta^{\mathrm{T}} a - \Psi(\theta)\bigr)\bigr]\, \mathrm{P}_\theta\left(\frac{S_n}{n} \approx a\right), \tag{10}$$

where P_θ denotes the probability induced by $F_\theta(\cdot)$. Since the left-hand side is independent of θ, for large n it is plausible that the θ which maximizes $\mathrm{P}_\theta(S_n/n \approx a)$, also maximizes $\theta^{\mathrm{T}} a - \Psi(\theta)$. Clearly, for large n, $\mathrm{P}_\theta(S_n/n \approx a)$ is maximized by $\tilde{\theta}_a$ such that $\mathrm{E}_{\tilde{\theta}_a} X_i = a$, so that by the law of large numbers this probability tends to 1 as $n \to \infty$ (E_θ denotes the expectation under the measure P_θ). Indeed

$$\theta_a = \arg\max_\theta \bigl(\theta^{\mathrm{T}} a - \Psi(\theta)\bigr),$$

uniquely satisfies the relation $\mathrm{E}_{\theta_a} X_i = a$. To see this note that θ_a is the solution to $\nabla\Psi(\theta) = a$ (it can be shown that such a θ_a uniquely exists for each $a \in \mathcal{J}^{\mathrm{o}}$). Also via differentiation, it is easily checked that for each θ,

$$\mathrm{E}_\theta X_i = \nabla\Psi(\theta).$$

In particular, $J(a) = \theta_a^{\mathrm{T}} a - \Psi(\theta_a)$ and (9) follows from (10).

For any set H, let $\overline{H}$ denote its closure. Define the rate function of the set $\mathcal{R}$,

$$J(\mathcal{R}) = \inf_{\alpha \in \mathcal{R}} J(\alpha).$$

For $\mathcal{R}$ that is sufficiently 'nice' so that $\overline{\mathcal{R}} = \overline{\mathcal{R}^{\mathrm{o}}}$ (e.g., in two dimensions $\mathcal{R}$ does not contain any isolated points or lines) and $\mathcal{R} \cap \mathcal{J}^{\mathrm{o}} \neq \emptyset$ so that there exist open intervals in $\mathcal{R}$ that can be reached with positive probability, the following large deviations relation holds

$$\lim_{n \to \infty} \frac{1}{n} \log \mathrm{P}\left(\frac{S_n}{n} \in \mathcal{R}\right) = -J(\mathcal{R}). \tag{11}$$

Note that there exists a point a^* on the boundary of $\overline{\mathcal{R}}$ such that $J(a^*) = J(\mathcal{R})$. Such an a^* is referred to as a minimum rate point. Intuitively, (11) may be seen quite easily when $\mathcal{R}$ is compact. Loosely speaking, the lower bound follows since, $\mathrm{P}(S_n/n \in \mathcal{R}) \geqslant \mathrm{P}(S_n/n \approx a^*)$ (where, in this special case, $S_n/n \approx a$ may be interpreted as the event that S_n/n lies in the intersection of $\mathcal{R}$ and a small ball of radius ε centered at a). Now if one thinks of $\mathcal{R}$ as covered by a finite number $m(\varepsilon)$ balls of radius ε centered at $(a^*, a_2, \dots, a_{m(\varepsilon)})$, then

$$\mathrm{P}\biggl(\frac{S_n}{n} \in \mathcal{R}\biggr) \leqslant \mathrm{P}\biggl(\frac{S_n}{n} \approx a^*\biggr) + \sum_{i=2}^{m(\varepsilon)} \mathrm{P}\biggl(\frac{S_n}{n} \approx a_i\biggr)$$
$$\overset{(\approx)}{\leqslant} m(\varepsilon) \exp\bigl(-nJ\bigl(a^*\bigr)\bigr) = m(\varepsilon) \exp\bigl(-nJ(\mathcal{R})\bigr)$$

and thus (11) may be expected.

Recall that from zero-variance estimation considerations, the new change of measure should assign high probability to the neighborhood of a^*. This is achieved by selecting $F_{\theta_{a^*}}(\cdot)$ as the IS distribution (since $\mathrm{E}_{\theta_{a^*}}(X_i) = a^*$). However, this may cause problems if the corresponding likelihood ratio

$$L_n = \exp\bigl(-n\bigl(\theta_{a^*}^{\mathrm{T}} x - \Psi(\theta_{a^*})\bigr)\bigr)$$

becomes large for some $x \in \mathcal{R}$, i.e., some points are assigned insufficient probability under $F_{\theta_{a^*}}(\cdot)$.

If all $x \in \mathcal{R}$ have the property that

$$\theta_{a^*}^{\mathrm{T}} x \geqslant \theta_{a^*}^{\mathrm{T}} a^*, \tag{12}$$

then the likelihood ratio for all $x \in \mathcal{R}$ is uniformly bounded by

$$\exp\bigl(-n\bigl(\theta_{a^*}^{\mathrm{T}} a^* - \Psi(\theta_{a^*})\bigr)\bigr) = \exp\bigl(-nJ(\mathcal{R})\bigr).$$

Hence $\mathrm{P}(S_n/n \in \mathcal{R}) = \mathrm{E}_{\theta_{a^*}}(L_n I(\mathcal{R})) \leqslant \exp(-nJ(\mathcal{R}))$ and $\mathrm{E}_{\theta_{a^*}}(L_n^2 I(\mathcal{R})) \leqslant \exp(-2nJ(\mathcal{R}))$ so that asymptotic optimality of $F_{\theta_{a^*}}(\cdot)$ follows.

The relation (12) motivates the definition of a *dominating point* (see Ney, 1983; Sadowsky and Bucklew, 1990). A minimum rate point a^* is a dominating point of the set $\mathcal{R}$ if

$$\mathcal{R} \subset \mathcal{H}\bigl(a^*\bigr) = \bigl\{x\colon\ \theta_{a^*}^{\mathrm{T}} x \geqslant \theta_{a^*}^{\mathrm{T}} a^*\bigr\}.$$

Recall that

$$J(a) = \theta_a^{\mathrm{T}} a - \Psi(\theta_a)$$

for $a \in \mathcal{J}^{\mathrm{o}}$. Thus, differentiating with respect to a component-wise and noting that $\theta_{a^*} = \nabla\Psi(\theta_{a^*})$ it follows that $\nabla J(a^*) = \theta_{a^*}$. Hence $\nabla J(a^*)$ is orthogonal to the plane $\theta_{a^*}^{\mathrm{T}} x = \theta_{a^*}^{\mathrm{T}} a^*$. In particular, this plane is tangential to the level set $\{x\colon J(x) = J(a^*)\}$. Clearly, if $\mathcal{R}$ is a convex set, we have $\mathcal{R} \subset \mathcal{H}(a^*)$. Of course, as Figure 1 indicates, this is by no means necessary. Figure 2 illustrates the case where $\mathcal{R}$ is not a subset of $\mathcal{H}(a^*)$. Even, in this case, $F_{\theta_{a^*}}(\cdot)$ may be asymptotically optimal if the region in $\mathcal{R}$ where the likelihood ratio is large has sufficiently small probability. Fortunately, in this more general setting, in

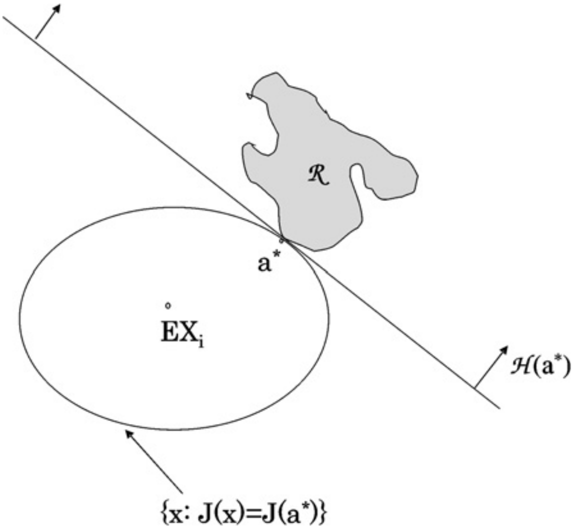


Fig. 1. Set with a dominating point a^*.

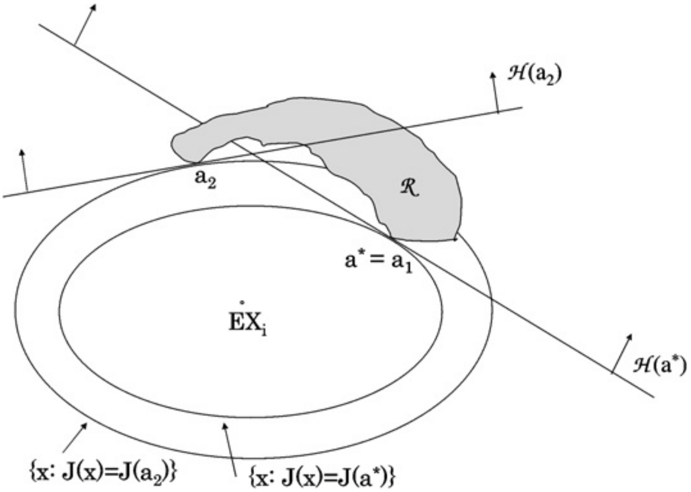


Fig. 2. Set with a minimum rate point a^* which is not a dominating point. Two points (a^*, a_2) are required to cover $\mathcal{R}$ with $\mathcal{H}(a^*)$ and $\mathcal{H}(a_2)$. Note that $J(a_2) > J(a^*)$ so that a_2 is not a minimum rate point.

Sadowsky and Bucklew (1990), sufficient conditions for asymptotic optimality are proposed that cover far more general sets $\mathcal{R}$. These conditions require existence of points $(a_1, \ldots, a_m) \subset \mathcal{J}^{\mathrm{o}} \cap \overline{\mathcal{R}}$ such that $\overline{\mathcal{R}} \subset \bigcup_{i=1}^{m} \mathcal{H}(a_i)$. Then for any positive numbers $(p_i\colon i \leqslant m)$ such that $\sum_{i \leqslant m} p_i = 1$, the distribution $F^*(\cdot) = \sum_{i \leqslant m} p_i F_{\theta_{a_i}}(\cdot)$ asymptotically optimally estimates $\mathrm{P}(S_n/n \in \mathcal{R})$.

Note that from an implementation viewpoint, generating S_n from the distribution F^* corresponds to generating a r.v. k from the discrete distribution $(p_1, \dots, p_m)$ and then generating $(X_1, \dots, X_n)$ using the distribution $F_{\theta_{a_k}}$ to independently generate each of the X_i's.

The fact that F^* is indeed a good importance sampling distribution is easy to see as the corresponding likelihood ratio (F w.r.t. F^*) equals

$$\frac{1}{\sum_{i\leqslant m} p_i \exp[n(\theta_{a_i}^{\mathrm{T}} x) + \Psi(\theta_{a_i})]} \leqslant \frac{\exp[-n(\theta_{a_i}^{\mathrm{T}} x) - \Psi(\theta_{a_i})]}{p_i} \leqslant \frac{\exp[-nJ(a_i)]}{p_i},$$

where the upper bound holds for any choice of i. This in turn is upper bounded by

$$\frac{\exp[-nJ(a^*)]}{\min_i p_i}.$$

For large n, this is a uniform upper bound assuring guaranteed variance reduction. It follows that

$$\lim_{n\to\infty} \frac{1}{n} \log \mathrm{E}_{\mathrm{P}^*} L^2 I(\mathcal{R}) \leqslant -2J(\mathcal{R})$$

assuring asymptotic optimality of P^*.

4.2 Probability of hitting a rare set

Let $T_\delta = \inf\{n\colon \delta S_n \in \mathcal{R}\}$. We now discuss efficient estimation techniques for the probability $\mathrm{P}(T_\delta < \infty)$ as $\delta \downarrow 0$. This problem generalizes the level crossing probability in the one-dimensional setting discussed by Siegmund (1976) and Asmussen (1989). Lehtonen and Nyrhinen (1992a, 1992b) considered the level crossing problem for Markov-additive processes. (Recall that Examples 5 and 6 also consider this.) Collamore (2002) considered the problem for Markov-additive processes in general state spaces. Again, we illustrate some of the key ideas for using importance sampling for this probability in the simple framework of S_n being a sum of i.i.d. random variables taking values in $\Re^d$, when $\overline{\mathcal{R}^o} = \overline{\mathcal{R}}$.

Note that the central tendency of the random walk S_n is along the ray $\lambda \mathrm{E} X_i$ for $\lambda \geqslant 0$. We further assume that $\mathrm{E}X_i$ does not equal zero and that $\mathcal{R}$ is disjoint with this ray, in the sense that

$$\mathcal{R} \cap \{\lambda x\colon \lambda > 0, x \approx \mathrm{E}X_i\} = \emptyset,$$

where $x \approx \mathrm{E}X_i$ means that x lies in a ball of radius $\varepsilon > 0$ centered at $\mathrm{E}X_i$, for some ε. Thus, $\mathrm{P}(T_\delta < \infty)$ is a rare event as $\delta \downarrow 0$. Figure 3 graphically illustrates this problem.

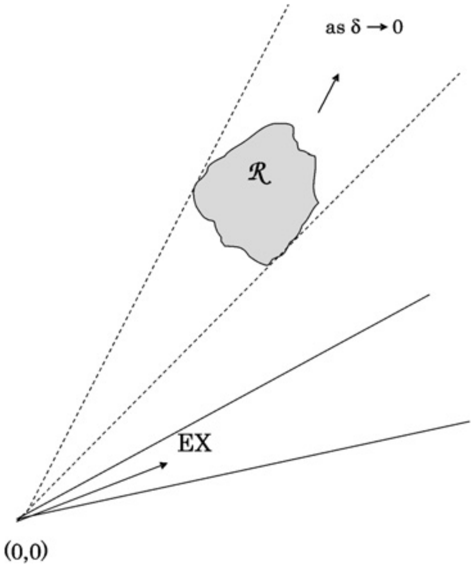


Fig. 3. Estimating the probability that the random walk hits the rare set.

First we heuristically arrive at the large deviations approximation for $\mathrm{P}(T_\delta < \infty)$ (see Collamore, 1996, for a rigorous analysis). Let

$$T_\delta(a) = \inf\{n\colon\ \delta S_n \approx a\},$$

where again $\delta S_n \approx a$ may be taken to be the event that δS_n lies in a small ball of radius ε centered at a.

Again, under importance sampling suppose that each X_i is generated using the twisted distribution F_θ. Then the likelihood ratio along $\{T_\delta(a) < \infty\}$ up till time $T_\delta(a)$ equals (approximately)

$$\exp\left[-\theta^{\mathrm{T}}\frac{a}{\delta} + T_\delta(a)\Psi(\theta)\right].$$

Suppose that θ is restricted to the set $\{\theta\colon\ \Psi(\theta) = 0\}$. This ensures that the likelihood ratio is *almost* constant. Thus, for such a θ we may write

$$\mathrm{P}\big(T_\delta(a) < \infty\big) \approx \exp\left[-\theta^{\mathrm{T}}\frac{a}{\delta}\right]\mathrm{P}_\theta\big(T_\delta(a) < \infty\big).$$

Again, the left-hand side is independent of θ so that $\tilde{\theta}$ that maximizes $\mathrm{P}_\theta(T_\delta(a) < \infty)$ as $\delta \to 0$ should also maximize $\theta^{\mathrm{T}}a$ subject to $\Psi(\theta) = 0$. Intuitively, one expects such a $\tilde{\theta}$ to have the property that the ray $\lambda \mathrm{E}_{\tilde{\theta}}(X_i)$ for $\lambda \geqslant 0$ intersects a, so that the central tendency of the random walk under $F_{\tilde{\theta}}$ is towards a. This may also be seen from the first-order conditions for the relaxed concave programming problem: Maximize $\theta^{\mathrm{T}}a$ subject to $\Psi(\theta) \leqslant 0$.

(It can be seen that the solution to the relaxed problem θ_a also satisfies the original constraint $\Psi(\theta) = 0$.) These amount to the existence of a scalar $\lambda > 0$ such that

$$\nabla\Psi(\theta_a) = \lambda a$$

(see, e.g., Luenberger, 1984).

We now heuristically argue that $\mathrm{P}_{\theta_a}(T_\delta(a) < \infty) \to 1$ as $\delta \to 0$. Under P_{θ_a}, from the central limit theorem,

$$S_n \approx n\mathrm{E}_{\theta_a} X_i + \sqrt{n}\,\mathrm{N}(0, C),$$

where $\mathrm{E}_{\theta_a} X_i = \lambda a$ denotes its drift and C denotes the covariance matrix of the components of X_i. In particular,

$$\delta S_{\lfloor \frac{1}{\lambda\delta} \rfloor} \approx a + \sqrt{\frac{\delta}{\lambda}}\,\mathrm{N}(0, C)$$

and this converges to a as $\delta \to 0$ suggesting that $\mathrm{P}_{\theta_a}(T_\delta(a) < \infty) \to 1$.

Thus heuristically,

$$\mathrm{P}\bigl(T_\delta(a) < \infty\bigr) \approx \exp\left[-\theta_a^{\mathrm{T}} \frac{a}{\delta}\right]$$

and

$$\mathrm{P}\bigl(T_\delta(\mathcal{R}) < \infty\bigr) \approx \exp\left[-\frac{H(\mathcal{R})}{\delta}\right],$$

where

$$H(\mathcal{R}) = \inf_{a\in\mathcal{R}} \theta_a^{\mathrm{T}} a = \inf_{a\in\mathcal{R}} \sup_{(\theta:\Psi(\theta)=0)} \theta^{\mathrm{T}} a.$$

Specifically, the following result may be derived

$$\lim_{\delta\to 0} \delta \log \mathrm{P}\bigl(T_\delta(\mathcal{R}) < \infty\bigr) = -H(\mathcal{R}) \tag{13}$$

(see Collamore, 1996). Suppose that there exists an $a^* \in \overline{\mathcal{R}}$ such that $H(\mathcal{R}) = \theta_{a^*}^{\mathrm{T}} a^*$. It is easy to see that such an a^* must be an *exposed point*, i.e., the ray $\{va^*\colon 0 \leqslant v < 1\}$ does not touch any point of $\mathcal{R}$. Furthermore, suppose that

$$\mathcal{R} \subset \mathcal{H}\bigl(a^*\bigr) \triangleq \bigl\{x\colon \theta_{a^*}^{\mathrm{T}} x \geqslant \theta_{a^*}^{\mathrm{T}} a^*\bigr\}.$$

Then, the likelihood ratio of F w.r.t. $F_{\theta_{a^*}}$ up till time $T_\delta(\mathcal{R})$ equals

$$\exp\bigl(-\theta_{a^*}^{\mathrm{T}} S_{T_\delta(\mathcal{R})}\bigr) \leqslant \exp\left(-\theta_{a^*}^{\mathrm{T}} \frac{a^*}{\delta}\right).$$

Thus, we observe guaranteed variance reduction while simulating under $F_{\theta_{a^*}}$ (note that $\mathrm{P}_{\theta_{a^*}}(T_\delta(\mathcal{R}) < \infty) \to 1$ as $\delta \to 0$). In addition, it follows that

$$\lim_{\delta\to 0} \delta \log \mathrm{E} L^2_{T_\delta(\mathcal{R})} I\bigl(T_\delta(\mathcal{R}) < \infty\bigr) \leqslant -2H(\mathcal{R}).$$

The above holds as an equality in light of (13), proving that $F_{\theta_{a^*}}$ ensures asymptotic optimality.

Again, as in the previous subsection, suppose that $\mathcal{R}$ is not a subset of $\mathcal{H}(a^*)$, and there exist points $(a_1, \ldots, a_m) \subset \overline{\mathcal{R}}$ $(a^* = a_1)$ such that $\overline{\mathcal{R}} \subset \bigcup_{i=1}^{m} \mathcal{H}(a_i)$. Then, for any positive numbers $(p_i\colon\ i \leqslant m)$ such that $\sum_{i \leqslant m} p_i = 1$, the distribution $F^*(\cdot) = \sum_{i \leqslant m} p_i F_{\theta_{a_i}}(\cdot)$ asymptotically optimally estimates $\mathrm{P}(T_\delta(\mathcal{R}) < \infty)$.

5 Adaptive importance sampling techniques

In this section we restrict our basic Markov process $(S_i\colon\ i \geqslant 0)$ to a finite state space $\mathcal{S}$. We associate a one-step transition reward $g(x, y) \geqslant 0$ with each transition $(x, y) \in \mathcal{S}^2$ and generalize our performance measure to that of estimating the expected cumulative reward until termination (when a terminal set of states $\mathcal{T}$ is hit) starting from any state $x \in \mathcal{S} - \mathcal{T}$, i.e., estimating

$$J^*(x) = \mathrm{E}_x\left[\sum_{k=0}^{T-1} g(S_k, S_{k+1})\right], \tag{14}$$

where the subscript x denotes that $S_0 = x$, and $T = \inf\{n\colon\ S_n \in \mathcal{T}\}$. Set $J^*(x) = 0$ for $x \in \mathcal{T}$. Note that if $g(x, y) = I(y \in \mathcal{R})$ with $\mathcal{R} \subseteq \mathcal{T}$, then $J^*(x)$ equals the probability $\mathrm{P}_x(S_T \in \mathcal{R})$.

We refer to the expected cumulative reward from any state as the value function evaluated at that state (this conforms with the terminology used in Markov decision process theory where the framework considered is particularly common; see, e.g., Bertsekas and Tsitsiklis, 1996). Note that by exploiting the regenerative structure of the Markov chain, the problem of estimating steady state measures can also be reduced to that of estimating cumulative reward until regeneration starting from the regenerative state (see, e.g., Fishman, 2001). Similarly, the problem of estimating the expected total discounted reward can be modeled as a cumulative reward until absorption problem after simple modifications (see, e.g., Bertsekas and Tsitsiklis, 1996; Ahamed et al., 2006).

For estimating $(J^*(x)\colon\ x \in \mathcal{T})$, the expression for the zero-variance change of measure is also well known, but involves knowing a priori these value functions (see Booth, 1985; Kollman et al., 1999; Desai and Glynn, 2001). Three substantially different adaptive importance sampling techniques have been proposed in the literature that iteratively attempt to learn this zero-variance change of measure and the associated value functions. These are: (i) the Adaptive Monte Carlo (AMC) method proposed in Kollman et al. (1999) (our terminology is adapted from Ahamed et al., 2006), (ii) the Cross Entropy (CE) method proposed in De Boer et al. (2000) and De Boer (2001) (also see Rubinstein, 1997, 1999) and (iii) the Adaptive Stochastic Approximation (ASA) based method proposed in Ahamed et al. (2006). We briefly review these methods. We refer the reader to Ahamed et al. (2006) for a comparison of the three methods on a small Jackson network example (this example is known to be difficult to efficiently simulate via static importance sampling).

Borkar et al. (2004) consider the problem of simulation-based estimation of performance measures for a Markov chain conditioned on a rare event. The conditional law depends on the solution of a multiplicative Poisson equation. They propose an adaptive two-time scale stochastic approximation based scheme for learning this solution. This solution is also important in estimating rare-event probabilities associated with queues and random walks involving Markov additive processes as in many such settings the static optimal importance sampling change of measure is known and is determined by solving an appropriate multiplicative Poisson equation (see, e.g., Chang et al., 1994; Beck et al., 1999). We also include a brief review of their scheme in this section.

5.1 The zero-variance measure

Let $P = (p_{xy}\colon x, y \in \mathcal{S})$ denote the transition matrix of the Markov chain and let $\mathbf{P}$ denote the probability measure induced by P and an appropriate initial distribution that will be clear from the context. We assume that $\mathcal{T}$ is reachable from all *interior* states $\mathcal{I} \overset{\Delta}{=} \mathcal{S} - \mathcal{T}$, i.e., there exists a path of positive probability connecting every state in $\mathcal{I}$ to $\mathcal{T}$. Thus T is an a.s. finite stopping time for all initial values of S_0. Consider another probability measure $\mathbf{P}'$ with a transition matrix $P' = (p'_{xy}\colon x, y \in \mathcal{S})$, such that for all $x \in \mathcal{I}$, $y \in \mathcal{S}$, $p'_{xy} = 0$ implies $p_{xy} = 0$. Let E' denote the corresponding expectation operator. Then $J^*(x)$ may be re-expressed as

$$J^*(x) = \mathrm{E}'_x\left[\left(\sum_{n=0}^{T-1} g(S_n, S_{n+1})\right) L(S_0, S_1, \ldots, S_T)\right], \tag{15}$$

where

$$L(S_0, S_1, \ldots, S_T) = \prod_{n=0}^{T-1} \frac{p_{S_n,S_{n+1}}}{p'_{S_n,S_{n+1}}}.$$

Noting that $\mathrm{E}'_x[g(S_n, S_{n+1})L(S_0, S_1, \ldots, S_{n+1})I(T > n)]$ equals

$$\mathrm{E}'_x\big[g(S_n, S_{n+1})L(S_0, S_1, \ldots, S_T)I(T > n)\big],$$

it may be easily seen that $J^*(x)$ equals

$$\mathrm{E}'_x\left[\left(\sum_{n=0}^{T-1} g(S_n, S_{n+1})L(S_0, S_1, \ldots, S_{n+1})\right)\right].$$

In this framework as well, the static zero-variance change of measure $\mathbf{P}^*$ (with corresponding transition matrix P^*) exists and the process $(S_i\colon i \geqslant 0)$ remains a Markov chain under this change of measure. Specifically, consider the transition probabilities

$$p^*_{xy} = \frac{p_{xy}(g(x, y) + J^*(y))}{\sum_{y \in S} p_{xy}(g(x, y) + J^*(y))} = \frac{p_{xy}(g(x, y) + J^*(y))}{J^*(x)}$$

for $x \in \mathcal{I}$ and $y \in \mathcal{S}$.

Then it can be shown that $K = \sum_{n=0}^{T-1} g(S_n, S_{n+1}) L(S_0, S_1, \ldots, S_{n+1})$ equals $J^*(S_0)$ a.s., where

$$L(S_0, S_1, \ldots, S_{n+1}) = \prod_{m=0}^{n} \frac{p_{S_m, S_{m+1}}}{p^*_{S_m, S_{m+1}}} = \prod_{m=0}^{n} \frac{J^*(S_m)}{g(S_m, S_{m+1}) + J^*(S_{m+1})}$$

(see Booth, 1985; Kollman et al., 1999; Desai and Glynn, 2001). We show this via induction. First consider $T = 1$. Then

$$K = g(S_0, S_1) \frac{J^*(S_0)}{g(S_0, S_1) + J^*(S_1)}.$$

Since $J^*(S_1) = J^*(S_T) = 0$, the result follows. Now suppose that the result is correct for all paths of length less than or equal to n. Suppose that $T = n + 1$. Then, K equals

$$g(S_0, S_1) \frac{J^*(S_0)}{g(S_0, S_1) + J^*(S_1)} + \frac{J^*(S_0)}{g(S_0, S_1) + J^*(S_1)} \left(\sum_{m=1}^{T-1} g(S_m, S_{m+1}) \right) \times \prod_{j=1}^{m} \frac{J^*(S_j)}{g(S_j, S_{j+1}) + J^*(S_{j+1})}.$$

By the induction hypothesis, $\sum_{m=1}^{T-1} g(S_m, S_{m+1}) \prod_{j=1}^{m} \frac{J^*(S_j)}{g(S_j, S_{j+1}) + J^*(S_{j+1})}$ equals $J^*(S_1)$ and the result follows.

Adaptive importance sampling techniques described in the following subsections attempt to learn this change of measure via simulation using an iterative scheme that updates the change of measure (while also updating the value function) so that eventually it converges to the zero-variance change of measure.

5.2 The adaptive Monte Carlo method

We describe here the basic AMC algorithm and refer the reader to Kollman et al. (1999) and Desai and Glynn (2001) for detailed analysis and further enhancements.

The AMC algorithm proceeds iteratively as follows: Initially make a reasonable guess $J^{(0)} > 0$ for J^*, where $J^{(0)} = (J^{(0)}(x) \colon x \in \mathcal{I})$ and

$J^* = (J^*(x)\colon x \in \mathcal{I})$. Suppose that $J^{(n)} = (J^{(n)}(x)\colon x \in \mathcal{I})$ denotes the best guess of the solution J^* at an iteration n (since $J^*(x) = 0$ for $x \in \mathcal{T}$, we also have $J^{(n)}(x) = 0$ for such x for all n). This $J^{(n)}$ is used to construct a new importance sampling change of measure that will then drive the sampling in the next iteration. The transition probabilities $P^{(n)} = (p_{xy}^{(n)}\colon x \in \mathcal{I}, y \in \mathcal{S})$ associated with $J^{(n)}$ are given as

$$p_{xy}^{(n)} = \frac{p_{xy}(g(x,y) + J^{(n)}(y))}{\sum_{y \in \mathcal{S}} p_{xy}(g(x,y) + J^{(n)}(y))}. \tag{16}$$

Then for each state $x \in \mathcal{S}$, the Markov chain is simulated until time T using the transition matrix $P^{(n)}$ and the simulation output is adjusted by using the appropriate likelihood ratio. The average of many, say r, such independent samples gives a new estimate $J^{(n+1)}(x)$. This is repeated independently for all $x \in \mathcal{I}$ and the resultant estimates of $(J^{(n+1)}(x)\colon x \in \mathcal{I})$ determine the transition matrix $P^{(n+1)}$ used in the next iteration. Since at any iteration, i.i.d. samples are generated, an approximate confidence interval can be constructed in the usual way (see, e.g., Fishman, 2001) and this may be used in a stopping rule.

Kollman et al. (1999) prove the remarkable result that if r in the algorithm is chosen to be sufficiently large, then there exists a $\theta > 0$ such that

$$\exp(\theta n)\|J^{(n)} - J^*\| \to 0,$$

a.s. for some norm in $\Re^{|\mathcal{I}|}$.

The proof involves showing the two broad steps:

- For any $\varepsilon > 0$, $\mathbf{P}(\|J^{(n)} - J^*\| < \varepsilon$ infinitely often) equals 1.
- Given that $\|J^{(0)} - J^*\| < \varepsilon$ there exists a $0 \leqslant c < 1$ and a positive constant ν such that the conditional probability

$$\mathbf{P}\big(\|J^{(n)} - J^*\| < c^n\|J^{(0)} - J^*\|, \forall n \big| \|J^{(0)} - J^*\| < \varepsilon\big) \geqslant \nu,$$

which makes the result easier to fathom.

5.3 The cross-entropy method

The Cross Entropy (CE) method was originally proposed in Rubinstein (1997) and Rubinstein (1999). See De Boer et al. (2005) for a tutorial. The essential idea is to select an importance sampling distribution from a specified set of probability distributions that minimizes the Kullback–Leibler distance from the zero-variance change of measure. To illustrate this idea, again consider the problem of estimating the rare-event probability $\mathbf{P}(\mathcal{E})$ for $\mathcal{E} \subset \Omega$. To simplify the description suppose that Ω consists of a finite or countable number of elements (the discussion carries through more generally in a straightforward manner). Recall that $\mathbf{P}^*$ such that

$$\mathbf{P}^*(\omega) = \frac{I(\mathcal{E})}{\mathbf{P}(\mathcal{E})}\mathbf{P}(\omega) \tag{17}$$

is a zero-variance estimator for $\mathbf{P}(\mathcal{E})$.

The CE method considers a class of distributions $(\mathbf{P}_\nu \colon \nu \in \mathcal{N})$ where $\mathbf{P}$ is absolutely continuous w.r.t. $\mathbf{P}_\nu$ on the set $\mathcal{E}$ for all ν. This class is chosen so that it is easy to generate samples of $I(\mathcal{E})$ under distributions in this class. Among this class, the CE method suggests that a distribution that minimizes the Kullback–Leibler distance from the zero variance change of measure be selected. The Kullback–Leibler distance of distribution $\mathbf{P}_1$ from distribution $\mathbf{P}_2$ equals

$$\sum_{\omega\in\Omega} \log\left[\frac{\mathbf{P}_2(\omega)}{\mathbf{P}_1(\omega)}\right]\mathbf{P}_2(\omega)$$

(note that this equals zero iff $\mathbf{P}_1 = \mathbf{P}_2$ a.s.). Thus, we search for a $\mathbf{P}_\nu$ that minimizes

$$\sum_{\omega\in\Omega} \log\left[\frac{\mathbf{P}^*(\omega)}{\mathbf{P}_\nu(\omega)}\right]\mathbf{P}^*(\omega),$$

where $\mathbf{P}^*$ corresponds to the zero-variance change of measure. From (17) and the fact that $\sum_{\omega\in\Omega} \log[\mathbf{P}^*(\omega)]\mathbf{P}^*(\omega)$ is a constant, this can be seen to be equivalent to finding

$$\arg\max_{\nu\in\mathcal{N}} \sum_{\omega\in\mathcal{E}} \log\bigl[\mathbf{P}_\nu(\omega)\bigr]\mathbf{P}(\omega). \tag{18}$$

Let $\widetilde{\mathbf{P}}$ be another distribution such that $\mathbf{P}$ is absolutely continuous w.r.t. it. Let $\widetilde{L}(\omega) = \frac{\mathbf{P}(\omega)}{\widetilde{\mathbf{P}}(\omega)}$. Then solving (18) is equivalent to finding

$$\begin{aligned}
&\arg\max_{\nu\in\mathcal{N}} \sum_{\omega\in\mathcal{E}} \log\bigl[\mathbf{P}_\nu(\omega)\bigr]\widetilde{L}(\omega)\widetilde{\mathbf{P}}(\omega)\\
&\quad = \arg\max_{\nu\in\mathcal{N}} \widetilde{\mathrm{E}} \log(\mathbf{P}_\nu)\widetilde{L}I(\mathcal{E}).
\end{aligned} \tag{19}$$

Rubinstein (1997, 1999) (also see Rubinstein and Kroese, 2004) propose to approximately solve this iteratively by replacing the expectation by the observed sample average as follows: Select an initial $\nu_0 \in \mathcal{N}$ in iteration 0. Suppose that $\nu_n \in \mathcal{N}$ is selected at iteration n. Generate i.i.d. samples $(\omega_1, \ldots, \omega_m)$ using $\mathbf{P}_{\nu_n}$, let $L_\nu(\omega) = \frac{\mathbf{P}(\omega)}{\mathbf{P}_\nu(\omega)}$ and select ν_{n+1} as the

$$\arg\max_{\nu\in\mathcal{N}} \frac{1}{m} \sum_{i=1}^{m} \log\bigl(\mathbf{P}_\nu(\omega_i)\bigr)L_{\nu_n}(\omega_i)I(\omega_i \in \mathcal{E}). \tag{20}$$

The advantage in this approach is that often it is easy to explicitly identify $\mathbf{P}_{\nu_n}$. Often the rare event considered corresponds to an event $\{f(\mathbf{X}) > x\}$, where $\mathbf{X}$ is a random vector, and $f(\cdot)$ is a function such that the event $\{f(\mathbf{X}) > x\}$ becomes rarer as $x \to \infty$. In such settings Rubinstein (1999) also proposes that the level x be set to a small value initially so that the event $\{f(\mathbf{X}) > x\}$ is not rare under the original probability. The iterations start with the original measure. Iteratively, as the probability measure is updated, this level may also be adaptively increased to its correct value.

In De Boer et al. (2000) and De Boer (2001) a more specialized Markov chain than the framework described in the beginning of this section is considered. They consider $\mathcal{T} = \mathcal{A} \cup \mathcal{R}$ ($\mathcal{A}$ and $\mathcal{R}$ are disjoint) and $g(x, y) = I(y \in \mathcal{R})$, so that $J^*(x)$ equals the probability that starting from state x, $\mathcal{R}$ is visited before $\mathcal{A}$. The set $\mathcal{A}$ corresponds to an attractor set, i.e., a set visited frequently by the Markov chain, and $\mathcal{R}$ corresponds to a rare set. Specifically, they consider a stable Jackson queueing network with a common buffer shared by all queues. The set $\mathcal{A}$ corresponds to the single state where all the queues are empty and $\mathcal{R}$ corresponds to the set of states where the buffer is full. The probability of interest is the probability that starting from a single arrival to an empty network, the buffer becomes full before the network re-empties (let $\mathcal{E}$ denote this event). Such probabilities are important in determining the steady state loss probabilities in networks with common finite buffer (see Parekh and Walrand, 1989; Heidelberger, 1995).

In this setting, under the CE algorithm, De Boer et al. (2000) and De Boer (2001) consider the search space that includes all probability measures under which the stochastic process remains a Markov chain so that $\mathbf{P}$ is absolutely continuous w.r.t. them. The resultant CE algorithm is iterative.

Initial transition probabilities are selected so that the rare event is no longer rare under these probabilities. Suppose that at iteration n the transition probabilities of the importance sampling distribution are $\mathrm{P}^{(n)} = (p_{xy}^{(n)}\colon x \in \mathcal{I}, y \in \mathcal{S})$. Using these transition probabilities a large number of paths are generated that originate from the attractor set of states and terminate when either the attractor or the rare set is hit. Let k denote the number of paths generated. Let $I_i(\mathcal{E})$ denote the indicator function of path i that takes value one if the rare set is hit and zero otherwise. The new $p_{xy}^{(n+1)}$ corresponding to the optimal solution to (20) is shown in De Boer (2001) to equal the ratio

$$\frac{\sum_{i=1}^{k} L_i N_{xy}(i) I_i(\mathcal{E})}{\sum_{i=1}^{k} L_i N_x(i) I_i(\mathcal{E})}, \tag{21}$$

where $N_{xy}(i)$ denotes the number of transitions from state x to state y and $N_x(i)$ denotes the total number of transitions from state x along the generated path i, L_i denotes the likelihood ratio of the path i, i.e., the ratio of the original probability of the path (corresponding to transition matrix P) and the new probability of the path (corresponding to transition matrix $P^{(n)}$). It is easy to see that as $k \to \infty$, the probabilities converge to the transition probabilities of the zero-variance change of measure (interestingly, this is not true if k is fixed and n increases to infinity).

The problem with the algorithm above is that when the state space is large, for many transitions (x, y), $N_{xy}(i)$ may be zero for all $i \leqslant k$. For such cases, the references above propose a number of modifications that exploit the fact that queues in Jackson networks behave like reflected random walks. Thus, consider the set of states where a subset of queues is nonempty in a network. For all these states, the probabilistic jump structure is independent of the state. This allows for clever state aggregation techniques proposed in the references above for updating the transition probabilities in each iteration of the CE method.

5.4 The adaptive stochastic approximation based algorithm

We now discuss the adaptive stochastic approximation algorithm proposed in Ahamed et al. (2006). It involves generating a trajectory via simulation where at each transition along the generated trajectory the estimate of the value function of the state visited is updated, and along with this at every transition the change of measure used to generate the trajectory is also updated. It is shown that as the number of transitions increases to infinity, the estimate of the value function converges to the true value and the transition probabilities of the Markov chain converge to the zero-variance change of measure.

Now we describe the algorithm precisely. Let $(a_n(x)\colon n \geqslant 0, x \in \mathcal{I})$ denote a sequence of nonnegative step-sizes that satisfy the conditions $\sum_{n=1}^{\infty} a_n(x) = \infty$ and $\sum_{n=1}^{\infty} a_n^2(x) < \infty$, a.s. for each $x \in \mathcal{I}$. Each $a_n(x)$ may depend upon the history of the algorithm until iteration n. This algorithm involves generating a path via simulation as follows:

- Select an arbitrary state $s_0 \in \mathcal{I}$. A reasonable positive initial guess $(J^{(0)}(x)\colon x \in \mathcal{I})$ for $(J^*(x)\colon x \in \mathcal{I})$ is made. Similarly, the initial transition probabilities $(p_{xy}^{(0)}\colon x \in \mathcal{I}, y \in \mathcal{S})$ are selected (e.g., these may equal the original transition probabilities). These probabilities are used to generate the next state s_1 in the simulation.
- At transition n, state s_{n+1} is generated using $(p_{xy}^{(n)}\colon x \in \mathcal{I}, y \in \mathcal{S})$. The updated values $(J^{(n+1)}(x)\colon x \in \mathcal{I})$ and $(p_{xy}^{(n+1)}\colon x \in \mathcal{I}, y \in \mathcal{S})$ are determined as follows:

$$\begin{aligned} &J^{(n+1)}(s_n) \\ &\quad = \bigl(1 - a_n(s_n)\bigr)J^{(n)}(s_n) \\ &\qquad + a_n(s_n)\bigl(g(s_n, s_{n+1}) + J^{(n)}(s_{n+1})\bigr)\left(\frac{p_{s_n s_{n+1}}}{p_{s_n s_{n+1}}^{(n)}}\right) \end{aligned} \tag{22}$$

and $J^{(n+1)}(x) = J^{(n)}(x)$ for $x \neq s_n$. Also, let

$$\tilde{p}_{s_n s_{n+1}}^{(n+1)} = p_{s_n s_{n+1}}\left(\frac{g(s_n, s_{n+1}) + J^{(n+1)}(s_{n+1})}{J^{(n+1)}(s_n)}\right). \tag{23}$$

This is normalized by setting $p^{(n+1)}_{s_n y} = (\tilde{p}^{(n+1)}_{s_n y})/(\sum_{z\in S} \tilde{p}^{(n+1)}_{s_n z})$ for all y (here $\tilde{p}^{(n+1)}_{s_n z} = p^{(n)}_{s_n z}$ for all $z \neq s_{n+1}$). Again for $x \neq s_n$, $p^{(n+1)}_{xy} = p^{(n)}_{xy}$ for all y.

- If $s_{n+1} \in \mathcal{T}$, the simulation is resumed by selecting s_{n+2} in $\mathcal{I}$ according to a probability distribution μ with the property that the Markov chain with transition probabilities that are the same as the original for all transitions from states in $\mathcal{I}$ and transition probabilities that are identically given by μ for transitions out of $\mathcal{T}$, is irreducible. In that case $(J^{(n+2)}(x)\colon\ x \in \mathcal{I})$ and $(p^{(n+2)}_{xy}\colon\ x \in \mathcal{I}, y \in \mathcal{S})$ are set to $(J^{(n+1)}(x)\colon\ x \in \mathcal{I})$ and $(p^{(n+1)}_{xy}\colon\ x \in \mathcal{I}, y \in \mathcal{S})$.

Ahamed et al. (2006) show that the algorithm above has the standard Robbins–Monro stochastic approximation form

$$J^{n+1} = (1 - a_n)J^n + a_n\big(HJ^n + w_n\big),$$

where each $J^n \in \Re^{|\mathcal{I}|}$, H is a mapping $\Re^{|\mathcal{I}|} \to \Re^{|\mathcal{I}|}$, w_n take values in $\Re^{|\mathcal{I}|}$ and are zero mean random vectors (see Kushner and Yin, 1997), and a_n are the step sizes. Under mild regularity conditions the mapping H can be seen to be a contraction under a suitable norm with a unique fixed point $J^* = (J^*(x)\colon\ x \in \mathcal{I})$ for any set of transition probabilities used to generate transitions (as long as the requirement of absolute continuity is met). Further, it can be shown that the second moment of w_n conditioned on the history of the algorithm up till time $n - 1$ is well behaved as required for convergence of the Robbins–Monro algorithm. From this it becomes easy to show that $J^{(n)} \to J^*$ and $P^{(n)} \to \mathrm{P}^*$. If, each step size a_n is set equal to a constant $a > 0$, then it is further shown that $\limsup_{n\to\infty} \mathrm{E}[\|J^{(n)} - J^*\|^2] = \mathrm{O}(a)$ and $\limsup_{n\to\infty} \mathrm{E}[\|P^{(n)} - \mathrm{P}^*\|^2] = \mathrm{O}(a)$.

Ahamed et al. (2006) report that empirically on representative examples, the ASA algorithm performs better than the AMC and the CE algorithm in estimating rare event probabilities when the state spaces involved become large or even moderately large. They further empirically show that for large state spaces, it may perform better than numerical procedures such as value iteration in developing estimates within a reasonable degree of accuracy.

5.5 Multiplicative Poisson equation and conditional measures

Many asymptotically optimal static importance sampling techniques often involve solving a complex set of equations to determine a provably effective static importance sampling distribution (see, e.g., Heidelberger, 1995). This could become particularly difficult when the underlying variables are Markov chains or more general Markov additive processes (see, e.g., Chang et al., 1994; Beck et al., 1999). We illustrate this through an example. Again, consider an irreducible Markov chain $(S_i\colon\ i \geqslant 0)$ on a large finite state space $\mathcal{S}$ with transition probabilities $(p(x, y)\colon\ x, y \in \mathcal{S})$. Let $g\colon\mathcal{S} \to R$. Let $\mathrm{E}[\cdot]$ denote the expectation under the associated invariant measure (i.e., steady-state measure) and let $\alpha > \mathrm{E}[g(S_n)]$. Now consider the problem of estimating the probability

$$\mathrm{P}_x\left(\frac{1}{n}\sum_{i=0}^{n-1} g(S_i) \geqslant \alpha\right) \tag{24}$$

for large values of n, where the subscript x denotes the condition $S_0 = x$ (note that this generalizes the i.i.d. case considered in Section 3.1). For such a probability, the static asymptotically optimal importance sampling measure (as $n \to \infty$) is well known (see, e.g., Bucklew, 1990). Under it, the transition probabilities are given by $(p^{\zeta^*}(x, y)\colon\ x, y \in \mathcal{S})$ where

$$p^{\zeta}(x, y) = \frac{\mathrm{e}^{\zeta g(x)} p(x, y) V_{\zeta}(y)}{\rho_{\zeta} V_{\zeta}(x)},$$

where $(V_{\zeta}(x)\colon\ x \in S)$ (resp., ρ_{ζ}) are the Perron–Frobenius eigenvector (resp., eigenvalue) of the positive operator

$$f(\cdot) \to \mathrm{e}^{\zeta g(\cdot)} \sum_{y} p(\cdot, y) f(y), \tag{25}$$

i.e., they solve the multiplicative Poisson equation

$$V_{\zeta}(x) = \frac{\mathrm{e}^{\zeta g(x)}}{\rho_{\zeta}} \sum_{y} p(x, y) V_{\zeta}(y), \quad x \in \mathcal{S}, \tag{26}$$

for $\zeta > 0$ and

$$\zeta^* \triangleq \arg\max_{\zeta \geqslant 0}\bigl(\zeta\alpha - \log(\rho_{\zeta})\bigr).$$

It can further be shown that $\log(\rho_{\zeta})$ is convex and that

$$\zeta^* = \arg\max\bigl(\zeta\alpha - \log(\rho_{\zeta})\bigr) \tag{27}$$

so that $\alpha = \frac{\rho'_{\zeta^*}}{\rho_{\zeta^*}}$ (the superscript "$'$" denotes the derivative). Furthermore, let $\mathrm{E}^{\zeta}[\cdot]$ denote the expectation under the invariant measure associated with transition probabilities $(p^{\zeta}_{xy}\colon\ x, y \in \mathcal{S})$. Then $\frac{\rho'_{\zeta}}{\rho_{\zeta}} = \mathrm{E}^{\zeta}[g(S_n)]$ (see, e.g., Bucklew, 1990).

Kontoyiannis and Meyn (2003) develop exact asymptotics for the probability (24) that again requires the knowledge of $(V_{\zeta^*}(x)\colon\ x \in \mathcal{S})$ and ρ_{ζ^*}. Borkar et al. (2004) use these exact asymptotics and observe the following asymptotic conditional law

$$\lim_{n\to\infty} \mathrm{P}\left(S_m = s_m \middle| S_k = s_k, 0 \leqslant k < m, \frac{1}{n}\sum_{i=0}^{n-1} g(S_i) \geqslant \alpha\right)$$
$$\to p^{\zeta^*}(s_{m-1}, s_m).$$

(The discussion in Section 3.3 can be generalized to include the probability in (24). The left-hand side above can be associated with the zero-variance measure that is shown to be asymptotically similar to the exponentially twisted distribution in the right-hand side.) Thus, the knowledge of these transition probabilities is useful in evaluating performance measures conditioned on occurrence of certain rare events, e.g., expected behavior of one queue conditioned on abnormal behavior of another queue in a Markovian finite state space network.

As mentioned earlier, in queueing set-ups when the constituent input or service processes are modeled as Markov additive processes (see, e.g., Chang et al., 1994; Beck et al., 1999) related Perron–Frobenius eigenvectors and eigenvalues need to be determined. Thus, the adaptive methodology discussed below becomes useful in such settings as well.

Evaluating $(V_{\zeta^*}(x)\colon\ x \in \mathcal{S})$ and ρ_{ζ^*} is especially difficult when the state space $\mathcal{S}$ is large. A deterministic iterative method to do this may involve first fixing ζ and evaluating ρ_ζ and ρ'_ζ (by re-evaluating the eigenvalue at a perturbed value of ζ). Borkar and Meyn (2002) develop deterministic numerical iterative schemes to solve this (however, such numerical schemes may not be computationally viable when large state spaces are involved). Once ρ_ζ and ρ'_ζ have been ascertained, ζ may be varied by adding to it a suitable step-size times $\alpha - \frac{\rho'_\zeta}{\rho_\zeta}$, the gradient of (27).

Borkar et al. (2004) develop an adaptive scheme (outlined below) that emulates this using stochastic observations. The adaptive scheme is based on a single simulation run of $(S_i\colon\ i \geqslant 0)$. Fix a distinguished state $s_0 \in \mathcal{S}$. Let $\{a(n)\}$, $\{b(n)\}$ be positive scalar sequences satisfying

$$\sum_n a(n) = \sum_n b(n) = \infty,$$
$$\sum_n \left(a(n)^2 + b(n)^2\right) < \infty, \qquad \frac{b(n)}{a(n)} \to 0. \tag{28}$$

These serve as step-sizes or 'learning parameters' for the iterative scheme. At each iteration n of the algorithm,

(1) simulate a transition from $S_n = s_n$ to $S_{n+1} = s_{n+1}$ (say) according to the current 'guess' of the transition probability $p^{\zeta^*}(s_n, \cdot)$ given by

$$p_n(s_n, y) \triangleq \frac{\mathrm{e}^{\zeta_n g(s_n)}}{V_n(s_n)V_n(s_0)} p(s_n, y)V_n(y), \tag{29}$$

normalized suitably to render it a probability vector, and

(2) update current guesses for $(V_{\zeta^*}(s_n), \zeta^*)$, denoted by $(V_n(s_n), \zeta_n)$, according to the iterative scheme that sets $V_{n+1}(s_n)$ to equal

$$V_n(s_n) + a(n)\left(\frac{\mathrm{e}^{\zeta_n g(s_n)}}{V_n(s_0)} V_n(s_{n+1})\left(\frac{p(s_n, s_{n+1})}{p_n(s_n, s_{n+1})}\right) - V_n(s_n)\right), \tag{30}$$

and

$$\zeta_{n+1} = \zeta_n + b(n)\big(\alpha - g(s_{n+1})\big). \tag{31}$$

Note that the two iterations proceed on different time-scales as $b(n) = \mathrm{o}(a(n))$ so that from the viewpoint of (30), ζ_n is more-or-less a constant function of n, while from the viewpoint of (31), (30) has reached the equilibrium associated with ζ_n.

The iteration (30) is motivated by the following considerations: To solve the equation

$$V_\zeta(x) = \frac{\mathrm{e}^{\zeta g(x)}}{\rho_\zeta} \sum_y p(x, y) V_\zeta(y), \quad x \in \mathcal{S}, \tag{32}$$

the following 'value iteration' scheme has been justified in Borkar and Meyn (2002) and Borkar (2002)

$$V^{n+1}(x) = \frac{\mathrm{e}^{\zeta g(x)}}{V^n(s_0)} \sum_y p(x, y) V^n(y), \quad x \in \mathcal{S}.$$

The conditional average on the right-hand side may be replaced by an actual evaluation at a simulated transition, i.e., by

$$\frac{\mathrm{e}^{\zeta g(x)}}{V^n(s_0)} V^n(y)$$

when $S_n = x$ and $S_{n+1} = y$. However, since this transition is conducted under the probability $p_n(x, y)$, the sample is unbiased by multiplying it by the likelihood ratio

$$\frac{p(x, y)}{p_n(x, y)}.$$

Then the averaging property of stochastic approximation is used to get (30) with $\zeta_n \equiv \zeta$. The iteration (31) corresponds to the stochastic gradient scheme applied to solve (27). In Borkar et al. (2004) convergence conditions of this algorithm are also discussed.

5.6 *Brief review of adaptive schemes in other contexts*

In a recent work Dupuis and Wang (2004) show that an adaptive importance sampling scheme can be devised to asymptotically optimally estimate

$\mathrm{P}(S_n/n \in \mathcal{R})$ for general sets $\mathcal{R}$ (recall that static importance sampling techniques for this probability were discussed in Section 3.1). Under this scheme each X_i is generated using a probability distribution that depends on the previously generated sum $\sum_{j=1}^{i-1} X_i$ (although, they do not learn the associated zero-variance measure).

Stochastic-approximation based adaptive approaches to importance sampling in the specific context of option pricing have been developed in Vázquez-Abad and Dufresne (1998), Su and Fu (2000, 2002). These take a 'stochastic-gradient' based approach using an approximate gradient search. They search for an optimal importance sampling change of measure from among a class of change of measures that is easy to implement. However this class does not include the zero-variance change of measure.

We conclude this section by noting that development of adaptive importance sampling techniques is an exciting, evolving area of research as many problems that were difficult to efficiently solve under naive Monte Carlo simulation or under static importance sampling can be efficiently solved under adaptive importance sampling techniques. The existing work on adaptive techniques has focused primarily on learning the zero-variance change of measure in discrete time discrete state Markov chains. Further research is needed to generalize this to continuous state spaces (see Bolia et al., 2004, for initial attempts in this direction for pricing American options).

6 Queueing systems

Heidelberger (1995) provides a survey of the literature on queueing systems. In this section we briefly mention some recent developments and review the earlier ones to motivate these. As discussed in Section 5, the recent research in adaptive importance sampling techniques shows great promise for estimation of rare-event probabilities associated with queues and queueing networks modeled as finite state Markov chains with relatively small state space. However, as the following discussion indicates, this problem remains open for a large variety of rare events associated with queueing networks when large state spaces or non-Markovian distributions are involved.

6.1 Single queues

Szechtman and Glynn (2002) develop large deviations asymptotics and asymptotically optimal importance sampling techniques for the probability of large queue lengths at a fixed time t for $GI/GI/\infty$ systems in heavy traffic, i.e., when the arrival rates are large. Kroese and Nicola (1999) and Juneja (2001) develop asymptotically optimal importance sampling techniques for estimation of buffer overflow probabilities for queues subject to server breakdowns.

6.2 Queueing networks

The key issue in rare-event estimation is illustrated by viewing rare events associated with suitably scaled queueing networks. For example, consider a Jackson network having K queues. Let $Q(t) = (Q_1(t), \dots, Q_K(t))$ for $t \geqslant 0$ denote the vector of the queue-length process. Consider the scaled process $\widetilde{Q}_n(t) = (1/n)Q(nt)$. Note that in a stable Jackson network $\widetilde{Q}_n(t)$ converges to zero for any t as $n \to \infty$. Significant large deviations literature has focused on identifying the most likely paths along which the process $(\widetilde{Q}_n(t)\colon t \geqslant 0)$ hits a set not containing the origin in the nonnegative orthant $\Re_+^K$ (see, e.g., Ignatiouk-Robert, 2000; Ignatyuk et al., 1994). In particular, it is shown that the most likely paths to such rare events are piece-wise linear. Avram et al. (2001) show this in the setting of semimartingale reflected Brownian motion, which typically provides a good approximation for heavily loaded queueing networks.

This is illustrated via a simple two queue tandem Jackson network example, with arrival rate to the first queue equal to λ and service rates at the first and second queues equal to μ_1 and μ_2, respectively, such that $\lambda < \mu_1 < \mu_2$. From the analysis in Ignatyuk et al. (1994) specialized to this network it can be inferred that, for the scaled network, the most likely path to reach the state (x_1, x_2), $x_1 \geqslant 0$, $x_2 > 0$, from the origin involves two piece-wise linear paths. Along the first path, the arrival rate to queue 1 equals μ_1, the service rate at queue 1 equals λ and the service rate at queue 2 remains the same at μ_2. Queue 1 builds up along this path until it reaches the level

$$\frac{\mu_2 - \mu_1}{\mu_2 - \lambda} x_2 + x_1.$$

Thereafter, along the second path, the arrival rate to queue 1 equals μ_1, the service rate at queue 1 equals μ_2 and the service rate at queue 2 equals λ, so that now queue 1 empties as queue 2 builds up until the state (x_1, x_2) is reached. This can also be inferred from the fact that starting from an empty network, the most likely paths to the rare event associated with queue lengths in Jackson networks correspond to the most likely path (in reverse direction) followed by the reversed Jackson network starting from the rare set until it empties; see Frater et al. (1991), Anantharam et al. (1990) and Heidelberger (1995). Figure 4 shows these paths for $x_1 = 0$. (Figure 4 also shows the case when $\mu_1 > \mu_2$, where there is a single path leading to $(0, x_2)$ along which the arrival rate equals μ_2, the service rate at the second queue equals λ, and the service rate at queue 1 remains unchanged.) This suggests that a change of measure should emphasize these two piecewise linear paths to efficiently estimate the buffer overflow probability in the second queue. More generally, this suggests that in general Jackson and other networks, a change of measure that is appropriately piecewise constant ought to be used. However, no such successful implementation has so far been designed. The problem is that in importance sampling it is not sufficient that the new measure emphasizes the

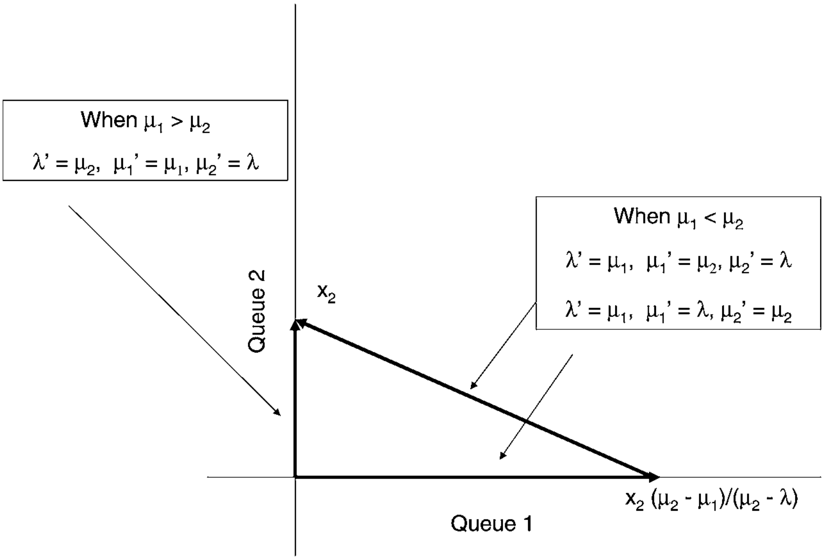

When $\mu_1 > \mu_2$
$\lambda' = \mu_2$, $\mu_1' = \mu_1$, $\mu_2' = \lambda$
When $\mu_1 < \mu_2$
$\lambda' = \mu_1$, $\mu_1' = \mu_2$, $\mu_2' = \lambda$
$\lambda' = \mu_1$, $\mu_1' = \lambda$, $\mu_2' = \mu_2$
Queue 2
x_2
$x_2 (\mu_2 - \mu_1)/(\mu_2 - \lambda)$
Queue 1

Fig. 4. The most likely paths along which the second queue builds up to level x_2 in a scaled two queue tandem Jackson network when $\mu_1 < \mu_2$ and when $\mu_1 > \mu_2$.

most likely paths to the rare event. It must not significantly reduce the original probability for *any* path to the rare event (to avoid build-up of the square of the likelihood ratio that can potentially blow up the second moment of the estimate).

The existing literature has focused primarily on two types of rare events. The first type focuses on estimating the probability that starting from an empty Jackson network, the total network population exceeds a given threshold before the network re-empties. For such a probability, when there exists a unique bottleneck queue, the most likely path to the rare event is linear and corresponds to this queue building up (this queue is unstable while the others remain stable along the most likely paths). Based on heuristic arguments using insights from large deviations theory Parekh and Walrand (1989) present a nonlinear program whose solution identifies these most likely paths (they consider generalized Jackson networks) and the associated importance sampling change of measure to simulate the queueing network. For Jackson networks Frater et al. (1991) obtain an explicit solution to this nonlinear program leading to an explicit form for this change of measure. However, for the simple two-node tandem Jackson network Glasserman and Kou (1995) show that this change of measure does not give good simulation results when the service rates at the two queues are close to each other. In a two-node Jackson network Randhawa and Juneja (2004) show that if feedback is allowed, then for certain traffic parameters the suggested change of measure leads to an estimator with infinite variance. They also discuss how the sensitivity of the second moment under importance sampling may be dampened by combining importance sampling with temporal difference control variates. Frater (1993) illustrate the difficulties associated with the case where the service rates in queues of the networks are close to each other.

The second type of rare-event probability in queueing network settings was considered by Chang et al. (1994) (also see Beck et al., 1999; L'Ecuyer and Champoux, 2001; Kroese and Nicola, 2002). They consider a special class of queueing networks referred to as in-tree networks. These in-tree networks are feed-forward networks (with no feedback). They consist of a 'target' queue at the root and other 'feeder' queues which are like leafs of a tree feeding the target queue at the root. They focus on estimating the probability that the buffer at the target queue overflows during its busy period (a busy period of the target queue is initiated when an arrival to it finds it empty, and it ends when subsequently the target queue re-empties). The remaining network is assumed to have the steady-state distribution at the instants of busy period initiation. The problem of efficient estimation of this probability is closely related to the problem of estimating the steady-state loss probability, i.e., the fraction of customers lost due to buffer overflow at the target queue in the steady-state (see, e.g., Chang et al., 1994; Heidelberger, 1995). They propose a change of measure that typically gives a large amount of variance reduction compared to naive simulation. Based on empirical observations they further suggest that the proposed change of measure is asymptotically optimal or near optimal when the queue-lengths at the feeder queues are assumed to be bounded at the initiation of a busy period of the target queue.

Juneja and Nicola (2005) consider this second type of rare-event probability in a more general setting that allows probabilistic routing with feedback. They generalize the change of measure proposed by Chang et al. (1994) to these settings (however, their analysis is restricted to Jackson networks). Here, they prove the asymptotic optimality of the proposed change of measure when the queue lengths at the feeder queues are assumed to be bounded at the initiation of a busy period of the target queue. Under the condition that the service rates at each feeder queue exceed a specified threshold, they prove that the proposed change of measure is asymptotically optimal, even when the feeder queue-lengths have steady state distributions at the instants of initiation of target queue busy periods. The condition on the feeder queue service rates ensures the large deviations path along which the target queue builds up has a single linear component. For example, in the simple two-queue tandem Jackson network example discussed earlier, the second queue builds up along a single linear path when $\mu_1 > \mu_2$.

7 Heavy-tailed simulations

A recent area of research is investigating rare-event simulation techniques when the random variables in the stochastic system are heavy-tailed. For the purposes of this paper we may define heavy-tailed to mean that the moment generating function of the distribution is infinite for any positive value of the argument. One important consequence of this is that the framework of exponential twisting that is used widely in the light-tailed area can no longer be used here. Also, as explained later, the manner in which rare events occur is very different in the light-tailed and the heavy-tailed settings.

Work in this area began with the problem of simulating tail probabilities of a sum of n i.i.d., nonnegative, heavy-tailed random variables, where n is either fixed or random. In the latter case n is denoted by N which is assumed to be independent of the i.i.d. random sequence. Estimation of tail probabilities of some simple functions (instead of just sums) of a fixed number of random variables that appear in the financial engineering setting and PERT (Project Evaluation and Review Technique) setting have also been developed recently. In this section we focus on fixed and geometric sums, and deal with the financial engineering setting in Section 8; for the PERT setting the reader is referred to Juneja et al. (2005). Recently, advances have been made in efficiently estimating level crossing probabilities of random walks. Recall that application of these probabilities include queueing and insurance settings (as discussed in Examples 5 and 6). We briefly review these developments later in this section.

The case of geometric N has applications in estimating the probability of large steady-state delays in the $M/\mathrm{GI}/1$ queue with heavy-tailed service times, or equivalently the probability of ultimate ruin in the insurance risk process with heavy-tailed claims (see Example 6). Consider Example 5, where the A_i's are exponentially distributed with rate λ and the B_i's are heavy-tailed. Let $\rho = \lambda \mathrm{E}(B_1)$. Let $(H_1, H_2, \ldots)$ denote the 'ladder-heights' of the random walk described in Example 5. The ladder-height H_i is the difference between the ith maximum and $(i-1)$st maximum of the random walk on a sample path where it achieves at least i new maxima; a maximum is achieved at j, if $S_i < S_j$ for all $i < j$ (see, e.g., Asmussen, 2003). These ladder heights are i.i.d., and when the A_i's are exponentially distributed, they have the distribution of the integrated-tail of B_i; note that if F_B denotes the distribution function of B_i then the distribution function of its integrated tail is given by $F_{B,I}(x) = \frac{1}{\mathrm{E}(B_1)} \int_0^x (1 - F_B(y))\,\mathrm{d}y$. Using the Pollaczeck–Khinchine formula, $\mathrm{P}(W_\infty > u)$ may be represented as $\mathrm{P}(\sum_{i=1}^N H_i > u)$, where N is independent of the H_i's, and geometrically distributed with parameter ρ (see, e.g., Asmussen and Binswanger, 1997). Typically H_i is heavy-tailed if B_i is.

The definition of heavy-tailed random variables as given above is almost equivalent to random variables belonging to the *subexponential family*. A nonnegative random variable X is said to be subexponential iff

$$\lim_{u\to\infty} \frac{\mathrm{P}(S_n > u)}{\mathrm{P}(X_1 > u)} = n$$

for all n (Chistyakov, 1964; Sigman, 1999), where $X_1, X_2, \ldots, X_n$ are i.i.d. copies of the random variable X and $S_n = \sum_{i=1}^n X_i$. This definition can be seen to be equivalent to the requirement that

$$\lim_{u\to\infty} \mathrm{P}\Big(\max_{i\leqslant n} X_i > u \big| S_n > u\Big) = 1.$$

This provides the main intuition that is used to investigate stochastic models with heavy tails, i.e., the most likely way for a sum of heavy-tailed random variables to become large is by one of the random variables becoming large. This is different from the light-tailed case, where all the random variables in the sum contribute to the sum becoming large. Common examples of subexponential random variables are the Pareto that has a tail that decays at a polynomial rate (e.g., $1/x^\alpha$ for $\alpha > 0$), the log-normal whose tail decays at the rate $\mathrm{e}^{-(\ln x)^2/(2\sigma^2)}$ (σ^2 is the variance of the associated normal random variable), and the heavy-tailed Weibull whose tail decays at the rate $\mathrm{e}^{-\lambda x^\alpha}$ for $0 < \alpha < 1$ and $\lambda > 0$. A comprehensive reference for subexponential distributions is Embrechts et al. (1997).

Consider the estimation of $\mathrm{P}(S_n > u)$ when $X_1, X_2, \ldots, X_n$ are nonnegative, subexponential random variables, and u is large. The first algorithm for their fast simulation was given by Asmussen and Binswanger (1997). It made direct use of the intuition mentioned above. Let $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$ denote the order statistics of $X_1, X_2, \ldots, X_n$. Since the most likely way u is exceeded is by one of the X_i's becoming large, it is $X_{(n)}$ that is the main cause of the variance of $I(S_n > u)$. Asmussen and Binswanger (1997) propose a sampling technique that involves generating samples of $X_1, X_2, \ldots, X_n$, then discarding sample of $X_{(n)}$, and using the conditional probability $\mathrm{P}(S_n > u | X_{(1)}, X_{(2)}, \ldots, X_{(n-1)})$ as an estimator of $\mathrm{P}(S_n > u)$. Asmussen and Binswanger (1997) show that this estimator is asymptotically optimal when the X_i's belong to the class of subexponential distributions that have regularly varying tails, e.g., Pareto-type tails (see Embrechts et al., 1997, for a precise definition) both for fixed n and geometric N. Asmussen and Kroese (2004) gives a related algorithm that has bounded relative error (instead of just asymptotic optimality) for all regularly varying distributions with Pareto-type tails.

Asmussen et al. (2000) gave an importance sampling algorithm for the fixed n and geometric N case. It involves doing importance sampling with another distribution with p.d.f. h, that has the following properties: the distribution K with density $k(x) = f^2(x)/(ch(x))$ where $c = \int f^2/h$ is subexponential and has the property $\ln \overline{K}(x) \leqslant \ln(\overline{F}(x))^2$ (here f denotes the original p.d.f. of X_i, $\overline{F}$ and $\overline{K}$ denote the tail d.f. associated with f and k, respectively). The main motivation for this condition is that terms similar to $k(X_i)$ occur in the square of the second moment of the importance sampling estimator. The condition is accomplished by selecting a p.d.f. h that has a tail that is much heavier than f. For example, for the case of Pareto, Weibull and lognormal f, the choice

$$h(x) = \frac{1}{(x+\mathrm{e})\ln(x+\mathrm{e})^2}, \quad x > 0,$$

works. For the geometric N case, an additional condition $c\rho < 1$ is needed for asymptotic optimality of h (recall that ρ is the parameter of the geometric distribution). Asmussen et al. (2000) also propose modifications to h to achieve this condition (similar modifications were independently proposed in Juneja et al., 1999; Juneja and Shahabuddin, 2002).

Juneja and Shahabuddin (2002) propose an importance sampling distribution obtained by *hazard rate twisting* the original distribution. Define the hazard rate function of X as $\Lambda(x) = -\ln \overline{F}(x)$ where $F(x)$ is the d.f. of X. The hazard rate twisted distribution with twisting parameter θ is given by

$$\mathrm{d}F_\theta(x) = \frac{\mathrm{e}^{\theta\Lambda(x)}\,\mathrm{d}F(x)}{\int_0^\infty \mathrm{e}^{\theta\Lambda(s)}\,\mathrm{d}F(s)} \tag{33}$$

for $0 < \theta < 1$. This is similar to exponential twisting, except that the twisting rate is $\theta\Lambda(x)$ instead of θx. Juneja and Shahabuddin (2002) show that the distributions associated with $\theta \equiv \theta_u = 1 - b/\Lambda(u)$ where b is any positive constant, asymptotically optimally estimate $\mathrm{P}(S_n > u)$, for a large class of subexponential distributions. The proof is easily seen if we assume that $\Lambda(\cdot)$ is concave (this is true for the common subexponential distributions like Weibull and Pareto). Assuming that X has a p.d.f.,

$$\begin{aligned}\int_0^\infty \mathrm{e}^{\theta\Lambda(s)}\,\mathrm{d}F(s) &= \int_0^\infty \mathrm{e}^{\theta\Lambda(s)} f(s)\,\mathrm{d}s\\ &= \int_0^\infty \mathrm{e}^{\theta\Lambda(s)}\big(\Lambda'(s)\mathrm{e}^{-\Lambda(s)}\big)\,\mathrm{d}s\\ &= \frac{1}{1-\theta}.\end{aligned}$$

Hence the likelihood ratio of the original distribution w.r.t. the hazard rate twisted distribution is given by

$$\prod_{i=1}^n \left(\frac{1}{1-\theta}\mathrm{e}^{-\Lambda(X_i)\theta}\right).$$

Since Λ is concave,

$$\sum_{i=1}^n \Lambda(X_i) \geqslant \Lambda\left(\sum_{i=1}^n X_i\right) = \Lambda(S_n).$$

Hence one can upper bound the second moment

$$\begin{aligned}&\mathrm{E}_\theta\left(I(S_n > u)\left(\frac{1}{1-\theta}\right)^{2n} \exp\left\{-\sum_{i=1}^n 2\theta\Lambda(X_i)\right\}\right)\\ &\quad\leqslant \left(\frac{1}{1-\theta}\right)^{2n} \mathrm{e}^{-2\theta\Lambda(u)}.\end{aligned}$$

(Here E_θ denotes the expectation operator under the hazard rate twisted distribution.) It is easily verified that selecting $\theta = 1 - b/\Lambda(u)$ gives asymptotic optimality. For fixed n, the choice $b = n$ minimizes the above bound on the second moment (this choice also minimizes the cross entropy as mentioned in Asmussen et al., 2005). Delayed hazard rate twisting and weighted delayed hazard rate twisting are modifications of hazard rate twisting, that, for suitably chosen parameters, asymptotically optimally estimate $\mathrm{P}(S_N > u)$ when N has a geometrically decaying tail. In cases where the hazard function is not available in closed form (e.g. log-normal), 'asymptotic' hazard rate twisting may also be effective (Juneja et al., 2005). In this case the $\Lambda(x)$ in (33) is replaced by $\widetilde{\Lambda}(x)$ where $\lim_{x\to\infty} \widetilde{\Lambda}(x)/\Lambda(x) = 1$ and $\widetilde{\Lambda}(x)$ is available in a simple closed form.

Kroese and Rubinstein (2004) and Huang and Shahabuddin (2004) consider the problem of estimating small tail probabilities of general functions of a finite number of heavy-tailed random variables. Kroese and Rubinstein (2004) propose approaches based on various parameterizations of the input distributions (that determine allowable changes of measure for the importance sampling), one of them being hazard rate twisting. The specific parameters for doing the importance sampling for a given problem are obtained by adaptively minimizing the cross-entropy. No asymptotic optimality results are shown. Huang and Shahabuddin (2004) propose applying hazard rate twisting by an amount θ to all the input distributions, and give an analytical way of determining a θ that yields asymptotic optimality in the simulation.

Huang and Shahabuddin (2003) and Kroese and Rubinstein (2004) make the observation that transforming a heavy-tailed random variable via its hazard function converts it to an exponential random variable with rate one. Hence one of the approaches suggested in Kroese and Rubinstein (2004) is to first use this transformation on each of the input random variables, and then apply exponential twisting to each of these resulting light-tailed random variables. This approach and the one mentioned earlier are equivalent, since it is shown in Huang and Shahabuddin (2003) that applying an exponential change of measure to the transformed input random variables by amount θ, $0 < \theta < 1$, is equivalent to hazard rate twisting the original random variable by amount θ.

We now consider two basic probabilities associated with random walks and queues: The level crossing probability and the level crossing probability during the busy cycle. Specifically, let S_0 be the initial state and for $n \geqslant 1$, let $S_n = S_0 + \sum_{i=1}^{n} X_i$ where the X_i's are random variables with negative mean and with right tails decaying at a subexponential rate. Let $\tau = \inf\{n \geqslant 1\colon S_n < 0\}$. Consider the problems of estimating $\mathrm{P}_0(\max_n S_n > u)$ and $\mathrm{P}_0(\max_{n\leqslant\tau} S_n > u)$ for large u, where the subscript x in P_x denotes that $S_0 = x$.

Boots and Shahabuddin (2000) propose a hazard rate twisting based algorithm for efficiently estimating $\mathrm{P}_0(\max_n S_n > u)$. They propose that the random walk be truncated after a specified number of transitions and they develop an asymptotically valid upper bound on the bias resulting from this truncation. Though this approach is asymptotically optimal for Weibull-type tails, it did not work well for Pareto-type tails. Bassamboo et al. (2005b) show that no distribution among the class of importance sampling distributions under which X_i's remain i.i.d., can asymptotically optimally estimate $\mathrm{P}_0(\max_{n\leqslant\tau} S_n > u)$ when the X_i's have Pareto-type polynomially decaying tails. They also develop explicit upper bounds on the improvement possible under such importance sampling distributions. This motivates the development of *state-dependent* importance sampling changes of measure that we now briefly discuss.

Again consider the problem of estimating $\mathrm{P}_0(\max_n S_n > u)$. From the discussion in Section 3.2 it can be seen that conditioned on $(X_1, X_2, \ldots, X_{n-1})$ and that the rare event has not occurred, the distribution of X_n under the zero-variance measure is given by

$$\mathrm{d}F_s^*(x) = \mathrm{d}F(x)\frac{\mathrm{P}_{s+x}(\max_n S_n > u)}{\mathrm{P}_s(\max_n S_n > u)},$$

when $\sum_{i=1}^{n-1} X_i = s$ (here F denotes the d.f. of X_1, and $\mathrm{P}_s(\max_n S_n > u) = 1$ for $s \geqslant u$). This is not implementable as $\mathrm{P}_s(\max_n S_n > u)$ is not known for $s < u$; if these were known, there would be no need for simulation. The idea then is to use asymptotic approximations to these probabilities as surrogates for them. Note that the same discussion also applies to $\mathrm{P}_0(\max_{n\leqslant\tau} S_n > u)$. Bassamboo et al. (2005b) develop asymptotic approximations for the latter probability in discrete settings. They then use the corresponding approximate zero variance measure and empirically demonstrate its asymptotic optimality in certain settings. Shahabuddin (2005) uses the approximate zero variance measure associated with the well-known asymptotic

$$\mathrm{P}_s\Bigl(\max_n S_n > u\Bigr) \sim \frac{1}{|\mathrm{E}(X)|}\int_{u-s}^{\infty} \overline{F}(t)\,\mathrm{d}t$$

derived in Pakes (1975) that holds when X is subexponentially distributed (under additional mild restrictions; see, e.g., Asmussen, 2003) to estimate the level crossing probability. However, they achieve limited success in this.

In a recent presentation, Blanchet and Glynn (2005) display a more refined approximation of the level crossing probability and claim that the change of measure corresponding to it leads to provably asymptotically optimal estimation of the level crossing probability.

8 Financial engineering applications

We first present some examples of rare-event simulation problems that arise in financial engineering.

Example 10 (Light-tailed value-at-risk). We first give a brief overview of the standard setting that has been given in Glasserman et al. (2000). Consider a portfolio consisting of several instruments (e.g., shares, options, bonds, etc.). The value of each instrument depends on one or more of m risk factors (e.g. stock price, price of gold, foreign exchange rate, etc.). Let $S(t) =$

$(S_1(t), \ldots, S_m(t))$ denote the values of the risk factors at time t and let $V(S(t), t)$ denote the value of the portfolio at time t (the values of several instruments, e.g., options, may depend directly on the time). Let t denote the current time, and let $\Delta S = [S(t + \Delta t) - S(t)]^{\mathrm{T}}$ (the notation A^{T} stands for the transpose of the matrix A) be the random change in risk factors over the future interval $(t, t + \Delta t)$. Hence the loss over the interval Δt is given by $\mathcal{L} = V(S(t), t) - V(S(t) + \Delta S, t + \Delta t)$. Note that the only random quantity in the expression for the loss is ΔS. The risk problem is to estimate $\mathrm{P}(\mathcal{L} > x)$ for a given x, and the value-at-risk problem is to estimate x such that $\mathrm{P}(\mathcal{L} > x) = p$ for a given p, $0 < p < 1$. Usually p is of the order 0.01 and Δt is either 1 day or 14 days. Techniques that are efficient for estimating $\mathrm{P}(\mathcal{L} > x)$ for a given x, can be adapted to estimate the value-at-risk. Hence the focus in most papers in this area is on efficient estimation of $\mathrm{P}(\mathcal{L} > x)$ for a given x.

A quadratic approximation to $\mathcal{L}$ is an approximation of the form

$$\mathcal{L} \approx a_0 + a^{\mathrm{T}} \Delta S + (\Delta S)^{\mathrm{T}} A \Delta S \equiv a_0 + Q, \tag{34}$$

where a_0 is a scalar, a is a vector and A is a matrix. The importance sampling approach given in Glasserman et al. (2000) involves determining an efficient change of measure on the ΔS for estimating $\mathrm{P}(Q + a_0 > x)$, and then using the same change of measure for estimating $\mathrm{P}(\mathcal{L} > x)$; since $\mathcal{L} \approx a_0 + Q$, it is likely that such an approach will be efficient for estimating the latter. The r.v. Q is more tractable and it is easier to come up with efficient changes of measure for estimating $\mathrm{P}(Q + a_0 > x)$ and proving their asymptotic optimality as $x \to \infty$. Glasserman et al. (2000) use the 'delta–gamma' approximation. This is simply the Taylor series expansion of the loss $\mathcal{L}$ with respect to ΔS and it uses the gradient and the Hessian of $\mathcal{L}$ with respect to ΔS to come up with Q. The gradient and Hessian may be computed analytically in cases where the portfolio consists of stocks and simple options.

Usually some probability model is assumed for the distribution of ΔS, and parameters of the model are estimated from historical data. A common assumption, that is also used in Glasserman et al. (2000), is that ΔS is distributed as $\mathrm{N}(0, \Sigma)$, i.e., it is multi-variate normal with mean zero and Σ is its covariance matrix. If we let C be such that $CC^{\mathrm{T}} = \Sigma$, then ΔS may be expressed as CZ where $Z \sim \mathrm{N}(0, I)$. Hence $Q = (Z^{\mathrm{T}} C^{\mathrm{T}} A C Z) + (a^{\mathrm{T}} C Z)$. For the case where Σ is positive definite, Glasserman et al. (2000) give a procedure to find such a C so that $C^{\mathrm{T}} A C$ is a diagonal matrix. In that case

$$Q = Z^{\mathrm{T}} \Lambda Z + b^{\mathrm{T}} Z = \sum_{i=1}^{m} \bigl(\lambda_i Z_i^2 + b_i Z_i\bigr), \tag{35}$$

where Λ is a diagonal matrix with λ_i's in the diagonal, and b is a vector with elements b_i. The problem is to find an efficient change of measure to estimate $\mathrm{P}(Q > y)$, for large $y := x + a_0$. Note that in this case Q is a sum of the independent random variables $X_i = (\lambda_i Z_i^2 + b_i Z_i)$.

Example 11 (Heavy-tailed value-at-risk). The multivariate normal is quite light-tailed and there is evidence from empirical finance that risk factors may have tails that are heavier than normal. Glasserman et al. (2002) consider the case where ΔS has a multivariate t distribution (i.e., the marginals have the univariate t distribution) with mean vector 0. The univariate t distribution with ν degrees of freedom has a tail that decays polynomially, i.e., similar to $x^{-\nu}$, as compared to $x^{-1}\exp(-x^2/2\sigma^2)$ which roughly describes the order of decay for the normal distribution. Glasserman et al. (2000) consider the version of the multivariate t as defined in Anderson (1984) and Tong (1990). This random variable may be expressed as

$$\frac{W}{\sqrt{\chi_\nu^2/\nu}},$$

where $W \sim \mathrm{N}(0, \Sigma)$ and χ_ν^2 is a chi-square random variable with ν degrees of freedom (see, e.g., Fang et al., 1987) that is independent of W. If we let $V = \chi_\nu^2/\nu$, then similar to (35), the diagonalized quadratic form becomes

$$Q = \sum_{i=1}^{m} \left(\frac{1}{V} \lambda_i Z_i^2 + \frac{1}{\sqrt{V}} b_i Z_i \right) \tag{36}$$

(as before, $Z \sim \mathrm{N}(0, I)$ and λ_i and b_i are constants). The problem is to determine an efficient change of measure for estimating $\mathrm{P}(Q > y)$ for large y, so that the same change of measure can be used for estimating the actual probability $\mathrm{P}(\mathcal{L} > y)$.

In this case the quadratic form is more complicated than the quadratic form for the normal case, due to two reasons.

- In this case, Q is heavy-tailed.
- We no longer have a sum of independent random variables; we now have dependence among the components in the sum through V. In this sense, this problem is more complex than the heavy-tailed problems considered in Asmussen and Binswanger (1997), Asmussen et al. (2000) and Juneja and Shahabuddin (2002), that dealt with sums of independent heavy-tailed random variables.

Example 12 (Credit risk). Consider a portfolio of loans that a lending institution makes to several obligors, say m. Obligors may default causing losses to the lending institution. There are several default models in the literature. In “static” default models, the interest is in the distribution of losses for the institution over a fixed horizon. More formally, corresponding to each obligor there is a default indicator Y_k, i.e., $Y_k = 1$ if the kth obligor defaults in the given time horizon, and it is zero otherwise. Let p_k be the probability of default of the kth obligor and c_k be the loss resulting from the default. The loss is then given by $\mathcal{L}_m = \sum_{k=1}^{m} c_k Y_k$. Efficient estimation of $\mathrm{P}(\mathcal{L}_m > x)$ when

m and x are large then becomes important. To study the asymptotics and rare-event simulation for this as well as more general performance measures, it is assumed that $x \equiv x_m = qm$ for fixed q, so that $\mathrm{P}(\mathcal{L}_m > x_m) \to 0$ as $m \to \infty$ (Glasserman and Li, 2005).

An important element that makes this problem different from the earlier random walk models is that in this case the Y_k's are dependent. One method to model this dependence is the normal copula model. (This methodology underlies essentially all models that descend from Merton's seminal firm-value work Merton (1974); also see Gupta et al. (1997).) In this case, with each Y_k a standard normal random variable X_k is associated. Let x_k be such that $\mathrm{P}(X_k > x_k) = p_k$, i.e., $x_k = \Phi^{-1}(1 - p_k)$ where Φ is the d.f. of standard normal distribution. Then, setting $Y_k = I(X_k > x_k)$, we get $\mathrm{P}(Y_k = 1) = p_k$ as required. Dependence can be introduced among the Y_k's by introducing dependence among the X_k's. This is done by assuming that each X_k depends on some "systemic risk factors" $Z_1, \ldots, Z_d$ that are standard normal and independent of one another, and an "idiosyncratic" risk factor ε_k that is also standard normal and independent of the Z_i's. Then each X_k is expressed as

$$X_k = \sum_{i=1}^{d} a_{ki} Z_i + b_k \varepsilon_k.$$

The a_{ki}'s are constants and represent the "factor-loadings", i.e., the effect of factor i on obligor k. The b_k is a constant that is set to $\sqrt{1 - \sum_{i=1}^{d} a_{ki}^2}$ so that X_k is standard normal.

8.1 Approaches for importance sampling

There are two basic approaches that have been used for determining asymptotically optimal changes of measures for problems of the type mentioned above. The first approach makes use of the light-tailed simulation framework of exponential twisting. As in Section 4, this is done with the aim of getting a uniform bound (see Section 2.4.1) on the likelihood ratio. For light-tailed problems like the one in Example 10, the framework can be applied directly. Heavy-tailed problems like the ones in Example 11, are transformed into light-tailed problems and then the framework is applied to them. All this is discussed in Sections 8.2–8.4. A general reference for this approach applied to several value-at-risk problems is Glasserman (2004); in the further discussion we attempt to bring out the essentials.

The second approach uses conditioning. Note that in Example 11, if we condition on V or B, then Q is reduced to the one in Example 10 (that is light-tailed) for which exponential twisting can be effectively used. Similarly in Example 12 in the normal copula model, conditioned on Z, the loss function is a sum of independent random variables for which the exponential twisting approach is well known. The question then arises as to what change of measure to use on the conditioning random variable, if any. This is discussed in Sections 8.5 and 8.6.

8.2 A light-tailed simulation framework

Consider the problem of estimating $\mathrm{P}(Y > y)$ where $Y = h(X_1, \dots, X_m)$, h is some function from $\Re^m$ to $\Re$, and $X_1, \dots, X_m$ are independent random variables, not necessarily i.i.d. For simplicity in presentation we assume that each X_i has a p.d.f. $f_i(x)$ and that the function h is sufficiently smooth so that Y also has a p.d.f. Let $F_i(x)$ be the d.f. of X_i, let $\overline{F_i}(x) = 1 - F_i(x)$, and define the hazard function as $\Lambda_i(x) = -\ln \overline{F_i}(x)$. Recall that for any two functions, say $g_1(x)$ and $g_2(x)$, $g_1(x) \sim g_2(x)$ means that $\lim_{x\to\infty} g_1(x)/g_2(x)$ exists and equals 1.

If we let $\tilde{f}_i(x)$ be a new p.d.f. for X_i, with the same support as X_i, then the importance sampling equation (2) specializes to

$$\mathrm{P}(Y > y) = \mathrm{E}\bigl(I(Y > y)\bigr) = \widetilde{\mathrm{E}}\bigl(I(Y > y)L(X_1, \dots, X_m)\bigr), \tag{37}$$

where

$$L(x_1, \dots, x_m) = \prod_{i=1}^{m} \frac{f_i(x_i)}{\tilde{f}_i(x_i)},$$

and $\widetilde{\mathrm{E}}(\cdot)$ denotes the expectation operator associated with the p.d.f.'s $\tilde{f}_i$. Once again, the attempt is to find $\tilde{f}_i$'s so that the associated change of measure is asymptotically optimal.

As mentioned in Section 3.3, for light-tailed random variables one may use the change of measure obtained by exponentially twisting the original distributions. In our case, exponentially twisting $f_i(x)$ by amount θ, $\theta > 0$, gives the new density

$$f_{i,\theta}(x) = \frac{f_i(x)\mathrm{e}^{\theta x}}{M_{X_i}(\theta)},$$

where $M_{X_i}(\theta)$ denotes the moment generating function (m.g.f.) of the random variable X_i.

Consider the case when Y is light-tailed. In that case the attempt in the literature is to find $\tilde{f}_1, \dots, \tilde{f}_m$, that translate into exponential twisting of Y by amount θ. This means that the new likelihood ratio, $L(X_1, \dots, X_m)$, is of the form $M_Y(\theta)\mathrm{e}^{-\theta Y}$. For example, consider the simple case where $Y = \sum_{i=1}^m X_i$, and the X_i's are light-tailed random variables. Now consider doing exponential twisting by amount θ on X_i. Then one can easily see that

$$L(X_1, \dots, X_m) = \prod_{i=1}^{m} \bigl(M_{X_i}(\theta)\mathrm{e}^{-\theta X_i}\bigr) = M_Y(\theta)\mathrm{e}^{-\theta Y}.$$

Hence, in this specific case, the exponential twisting of X_i's by θ translates into exponential twisting of Y by θ.

If such an exponential twist on the X_i's can be found, then the second moment can be bounded as follows:

$$\begin{aligned}\widetilde{\mathrm{E}}\big(I(Y > y)L^2(X_1, \ldots, X_m)\big) &= \widetilde{\mathrm{E}}\big(I(Y > y)M_Y^2(\theta)\mathrm{e}^{-2\theta Y}\big) \\ &\leqslant M_Y^2(\theta)\mathrm{e}^{-2\theta y}. \end{aligned} \tag{38}$$

Then $\theta = \theta_y^*$ may be selected that minimizes $M_Y^2(\theta)\mathrm{e}^{-2\theta y}$ or equivalently that minimizes $\ln M_Y(\theta) - \theta y$. Huang and Shahabuddin (2003) generalize earlier specific results and show that under fairly general conditions, this procedure yields asymptotically optimal estimation.

Hence, the main challenge in this approach is to find a change of measure on the X_i's that translates into exponential twisting of Y. We now see how this is done for the examples mentioned in the beginning of this section.

8.3 Light-tailed value-at-risk

Consider the problem of estimating $\mathrm{P}(Q > y)$, where Q is given by (35). In this case $Q = \sum_{i=1}^m V_i$, where $V_i = \lambda_i Z_i^2 + b_i Z_i$. Hence, as shown in Section 8.2, exponentially twisting each V_i by θ, will translate into exponential twisting of Q by θ. The question then is: What is the change of measure on the Z_i's that would achieve exponential twisting of the V_i's. Glasserman et al. (2000) show that this is achieved if the mean and variance of Z_i are changed to $\mu_i(\theta)$ and $\sigma_i^2(\theta)$, respectively, where

$$\sigma_i^2(\theta) = \frac{1}{1 - 2\theta\lambda_i}, \qquad \mu_i(\theta) = \theta b_i \sigma_i^2(\theta)$$

(the Z_i's remain independent).

Glasserman et al. (2000) perform a further enhancement to the simulation efficiency by using stratification on Q. Note that by completing squares, each V_i may be expressed as the sum of a noncentral chi-square r.v. and a constant. Hence its m.g.f. is known in closed form and thus the m.g.f. of Q can easily be obtained in closed form. This can be inverted to get the distribution of Q. This enables stratification on Q that further brings down the variance of the importance sampling estimator $I(Q > y)M_Q(\theta_y^*)\exp(-\theta_y^* Q)$. Glasserman et al. (2000) give a simple algorithm for generating $(Z_1, \ldots, Z_m)$ conditional on Q lying in given stratas.

8.4 Heavy-tailed value-at-risk: transformations to light tails

Consider estimating $\mathrm{P}(Q > y)$ where Q is given by (36). As mentioned before, Q is heavy-tailed and thus direct application of exponential twisting cannot be attempted here. Glasserman et al. (2002) transform this problem into a light-tailed problem before using exponential twisting. In particular, they define

$$Q_y = V(Q-y) = \sum_{i=1}^{m}\bigl(\lambda_i Z_i^2 + b_i Z_i\sqrt{V}\bigr) - yV.$$

It is easy to check that Q_y is light-tailed for each y, since all its components are light-tailed. Also $\mathrm{P}(Q > y) = \mathrm{P}(Q_y > 0)$ and hence a heavy-tailed simulation problem is transformed into a light-tailed one!

An exponential change of measure by amount $\theta \geqslant 0$ on Q_y can be attempted through selecting appropriate changes of measure for the Z_i's and V. In this case, we have the following simple bound on the second moment:

$$\mathrm{E}\bigl(I(Q_y > 0)M_{Q_y}^2(\theta)\mathrm{e}^{-2\theta Q_y}\bigr) \leqslant M_{Q_y}^2(\theta).$$

Adapting the same approach as in Section 8.2, a θ_y^* is selected that minimizes this bound. Indeed, as proved in Glasserman et al. (2002), for the case where $\lambda_i > 0$ for $i = 1, \dots, m$, this selection gives bounded relative error. Glasserman et al. (2002) also give an explicit change of measure (in terms of θ) on V, and changes of measure (in terms of θ) on Z_i's conditional on V, that achieve exponential twisting of Q_y by amount θ.

Huang and Shahabuddin (2003) give another approach for transforming a heavy-tailed simulation problem into a light-tailed one. Note that the hazard function of any random variable whose p.d.f. is positive on $\Re^+$ (resp., $\Re$) is an increasing function on $\Re^+$ (resp., $\Re$). Let $\Lambda_Y(y)$ be the hazard function of Y, and let $\Lambda(y)$ be any monotonically increasing function such that $\Lambda(y) \sim \Lambda_Y(y)$. Then it is shown in Huang and Shahabuddin (2003) that $\Lambda(Y)$ is exponential-tailed with rate 1. Usually such a $\Lambda(y)$ may be determined through asymptotic results in heavy-tailed theory, or by clever application of the Laplace method for solving integrals. Then $\mathrm{P}(Y > y)$ may be re-expressed as $\mathrm{P}(\Lambda(Y) > \Lambda(y))$, and we again have a light-tailed simulation problem where y is replaced by its monotonic transformation $\Lambda(y)$ (note that $\Lambda(y) \to \infty$ as $y \to \infty$). In this case, since $\Lambda(Y)$ is usually not in the form of a sum of functions of the individual X_i's, it is difficult to find a change of measure on the X_i's that will achieve exponential twisting on the $\Lambda(Y)$. For the case where the changes in risk factors have the Laplace distribution, Huang and Shahabuddin (2003) find upper bounds on $\Lambda(Y)$ that are in this form, so that exponential twisting can easily be applied.

8.5 Conditional importance sampling and zero-variance distributions

As mentioned in Example 11, conditioned on V, Q has the same form as in Example 10, for which the asymptotically optimal change of measure is much simpler to determine. This motivates a conditioning approach for such problems.

Consider the more general problem of estimating $\mathrm{P}(Y_y > 0)$ where $Y_y = h_y(X_1, \dots, X_m)$ and h_y is some function from $\Re^m$ to $\Re$ that also depends on y.

For the class of problems considered in Section 8.2, $Y_y = Y - y$. The Q_y described in Section 8.4 is also an example of this. Assume that $\mathrm{P}(Y_y > 0) \to 0$ as $y \to \infty$. Let $V = \tilde{h}(X_1, \dots, X_m)$ be a 'conditioning' random variable, where $\tilde{h}$ is some other function of the input random variables (usually V is a function of just one of the input random variables). As mentioned in the previous paragraph, it is important to select V such that, given $V = v$, it is easy to determine changes of measure on the X_i's that translate into exponential twisting of the Y_y. This implies that for any v, given $V = v$, the Y_y should be light-tailed.

For conditional importance sampling, we again use insights from the zero-variance change of measure. Note that

$$\mathrm{P}(Y_y > 0) = \int \mathrm{P}(Y_y > 0 | V = v) f_V(v)\,\mathrm{d}v. \tag{39}$$

Hence if $\mathrm{P}(Y_y > 0 | V = v)$ were computable for each v, then the zero-variance change of measure on the V (for estimating $\mathrm{P}(Y_y > 0)$) would be

$$\frac{\mathrm{P}(Y_y > 0 | V = v) f_V(v)}{\int \mathrm{P}(Y_y > 0 | V = v) f_V(v)\,\mathrm{d}v}. \tag{40}$$

Recall that Y_y is a tractable approximation to the actual loss function. Usually, given $V = v$, Y_y is a sum of independent random variables and hence $\mathrm{P}(Y_y > 0 | V = v)$ may be determined by numerical transform inversion techniques. Once one is able to compute $\mathrm{P}(Y_y > 0 | V = v)$ for each v, then one can compute $\mathrm{P}(Y_y > 0)$ by numerical integration. One can then generate from the zero-variance change of measure on the V by first computing its cumulative distribution function (using numerical integration) and then using numerical inversion. All this, even though theoretically possible, is practically possible usually for the case of only discrete V. The asymptotic optimality proof is usually not possible for either continuous or discrete V. This approach has been proposed in Shahabuddin and Woo (2004) and applied to the estimation of the tail probability of the quadratic form in the value-at-risk problem, where the risk factors have the distribution of a finite mixture of multivariate normals. In this case, the conditioning random variable V is the random identifier of the multivariate normal that one samples from in each step. The multivariate mixture of normals has applications for the case where the asset prices obey the jump diffusion model (see Shahabuddin and Woo, 2004).

As another approach to this problem, Shahabuddin and Woo (2004) use the Markov inequality

$$\mathrm{P}(Y_y > 0 | V = v) \leqslant \mathrm{E}\big(\mathrm{e}^{Y_y \theta^*_{y,v}} | V = v\big),$$

where $\theta^*_{y,v}$ is obtained from minimizing the Markov bound $\mathrm{E}(\mathrm{e}^{Y_y \theta} | V = v)$ over all $\theta \geqslant 0$. Then $\mathrm{E}(\mathrm{e}^{Y_y \theta^*_{y,v}} | V = v)$ may be used as a close surrogate to $\mathrm{P}(Y_y > 0 | V = v)$ in (40). Usually, this inequality is somewhat tight for large y, and hence not much loss in performance may be expected due to this substitution. Also, $\mathrm{E}(\mathrm{e}^{Y_y\theta^*_{y,v}}|V = v)$, the conditional moment generating function, is usually computable in closed form. By using this surrogate, the approximate zero-variance change of measure for the V would be

$$\tilde{f}_V(v) = \frac{\mathrm{E}(\mathrm{e}^{Y_y\theta^*_{y,v}}|V = v)f_V(v)}{\int \mathrm{E}(\mathrm{e}^{Y_y\theta^*_{y,v}}|V = v)f_V(v)\,\mathrm{d}v}.$$

Once again, even though theoretically possible, there are difficulties with this approach both regarding the implementation and the asymptotic optimality proof. In addition to the numerical computational burden in the previous approach, we have the additional burden of determining $\theta^*_{y,v}$ for each v, and $\theta^*_{y,v}$ is rarely available in closed form. Hence this approach is again efficient only for the case where V is a discrete random variable taking a finite number of values. In Shahabuddin and Woo (2004) it has been applied to the mixture of normal problems mentioned in the previous paragraph. The asymptotic optimality for this case is also proved.

In order to make the implementation simpler for a continuous conditioning random variable V, Shahabuddin and Woo (2004) consider relaxing the bound on $\mathrm{P}(Y_y > 0|V = v)$, by using the same θ for all $V = v$, and then determining the best θ to use. In that case, the approximate zero-variance distribution is given by

$$\tilde{f}_V(v) = \frac{\mathrm{E}(\mathrm{e}^{Y_y\theta}|V = v)f_V(v)}{\int \mathrm{E}(\mathrm{e}^{Y_y\theta}|V = v)f_V(v)\,\mathrm{d}v}.$$

In this case, if V is such that

$$\mathrm{E}\bigl(\mathrm{e}^{Y_y\theta}|V = v\bigr) = g_1(\theta, y)\mathrm{e}^{g_2(\theta,y)v} \tag{41}$$

(for any functions g_1 and g_2) then

$$\tilde{f}_V(v) = \frac{\mathrm{e}^{g_2(\theta,y)v}f_V(v)}{\int \mathrm{e}^{g_2(\theta,y)v}f_V(v)\,\mathrm{d}v},$$

i.e., the approximate zero-variance change of measure is then an exponential twisting by amount $g_2(\theta, y)$. Once V is sampled from the new measure, then one needs to do a change of measure on the X_i's given $V = v$, so that one achieves exponential twisting of Y_y. If this can be done, then it is easy to check that the likelihood ratio is $M_{Y_y}(\theta)\mathrm{e}^{-Y_y\theta}$, and thus we are in a framework similar to that in Section 8.4. As in that section, the second moment $\mathrm{E}(I(Y_y > 0)M^2_{Y_y}(\theta)\mathrm{e}^{-2\theta Y_y})$ may then be upper bounded by $M^2_{Y_y}(\theta)$, and a θ^*_y may be selected that minimizes this bound.

It is easy to check that for Q_y in the multivariate t case in Section 8.4, selecting V as the chi-square random variable achieves the condition given in (41). However, the choice of the conditioning variable may not always be obvious.

For example, consider the case where the risk factors have the Laplace distribution with mean vector 0, as considered in Huang and Shahabuddin (2003). In this case, the tails of the marginal distributions decay according to $\frac{1}{\sqrt{x}}\mathrm{e}^{-cx}$, for some constant $c > 0$. Justifications of this type of tail behavior may be found in Heyde and Kou (2004). The multivariate Laplace random-variable with mean vector 0 may be expressed as

$$\sqrt{B}W,$$

where $W \sim \mathrm{N}(0, \Sigma)$ and B is an exponentially distributed random variable with rate 1 (see, e.g., Kotz et al., 2001). In this case the Q_y becomes

$$Q_y = \sum_{i=1}^{m}\left(\lambda_i Z_i^2 + \frac{1}{\sqrt{B}}b_i Z_i - \frac{y}{B}\right). \tag{42}$$

However, taking B as the conditioning random variable does not work. In fact, a conditioning random variable V that satisfies (41) in this case is $V = -1/B$. This is indeed surprising since the V does not even take positive values and we are doing exponential twisting on this random variable! Shahabuddin and Woo (2004) thus generalize the exponential twisting idea in Glasserman et al. (2002) to make it more widely applicable. It also improves the earlier method in Huang and Shahabuddin (2003) for the case where the changes in risk factors are Laplace distributed (that was based on hazard rate twisting).

8.6 Credit risk models

Consider the model described in Example 12 where the problem is to estimate $\mathrm{P}(\mathcal{L}_m > x_m)$ where $\mathcal{L}_m = \sum_{k=1}^{m} c_k Y_k$ and $x_m = qm$ for some constant q. For the case of independent obligors, where the Y_k's are independent Bernoulli's, the basic procedure is the same as described in Section 8.2. For the case when the Y_k's are dependent, with the dependence structure specified in Example 12, Glasserman and Li (2005) first attempt doing importance sampling conditional on the realization of the normal random variable Z, but leaving the distribution of Z unchanged. A similar approach is also followed by Merino and Nyefeler (2004). Note that in this case, the probability of default for obligor k now becomes a function of Z, i.e.,

$$p_k(Z) = \Phi\left(\frac{\sum_{i=1}^{d} a_{ki} Z_i + \Phi^{-1}(p_k)}{b_k}\right).$$

In the simulation procedure, first Z is sampled and $p_k(Z)$'s are computed, then the importance sampling procedure mentioned above is applied by treating the $p_k(Z)$'s as fixed. In particular, let $\psi_i^{(m)}(\theta, z)$ be the log moment generating function of $\mathcal{L}_m$ given $Z = z$, and let $\theta_m(z)$ be the $\theta \geqslant 0$ that maximizes $-\theta qm + \psi_i^{(m)}(\theta, z)$. Then after sampling Z, exponential twisting is performed on the $c_i Y_i$'s by the amount $\theta_m(Z)$. Note that $\theta_m(Z) > 0$ only when Z is such that $\sum_{i=1}^{m} c_k p_k(Z) = \mathrm{E}(\mathcal{L}_m|Z) < qm$; for the other case $\theta_m(Z) = 0$, and we do not do importance sampling.

Glasserman and Li (2005) show that when the dependence among the Y_i's is sufficiently low, conducting importance sampling conditional on the Z is enough, i.e., the distribution of Z's need not be changed under importance sampling. However, when the dependence is higher, one also has to change the distribution of Z so that it has greater probability of falling in regions where the default events are likely to occur. Again, one approach is to select the importance sampling distribution of Z that is close to the zero-variance distribution.

In an earlier paper Glasserman et al. (1999) consider the problem of estimating $\mathrm{E}(\mathrm{e}^{G(Z)})$ where Z is $\mathrm{N}(0, I)$, and G is some function from $\Re^m$ to $\Re$. An importance sampling method proposed in Glasserman et al. (1999) was to find the point that maximizes $\mathrm{e}^{G(z)}\phi(z)$ (assuming it is unique) where $\phi(z)$ is the p.d.f. of $\mathrm{N}(0, I)$. If the maximum occurs at μ, then the new measure that is used for Z is $\mathrm{N}(\mu, I)$.

Once again, the intuition behind this procedure in Glasserman et al. (1999) is obtained from the zero-variance distribution. Note that the zero-variance distribution of Z is one that is proportional to $\mathrm{e}^{G(z)}\phi(z)$. The heuristic in Glasserman et al. (1999) is based on the idea that if one aligns the mode of the new normal distribution (i.e., μ) and the mode of $\mathrm{e}^{G(z)}\phi(z)$, then the two may also roughly have the same shape, thus approximately achieving the proportionality property. One can also see this if one approximates $G(z)$ by its first order Taylor series expansion around μ. Note that if $G(z)$ is exactly linear with slope a, then the zero-variance distribution can be easily derived as $\mathrm{N}(a, I)$. Also, in this case, it is easy to see that a minimizes $\mathrm{e}^{G(z)}\phi(z)$.

In the credit risk case, $G(z) = \ln \mathrm{P}(\mathcal{L}_m > qm|Z = z)$. Since $\mathrm{P}(\mathcal{L}_m > qm|Z = z)$ is usually not computable, one uses the upper bound obtained from the Markov inequality, i.e.,

$$G(z) = \ln \mathrm{P}(\mathcal{L}_m > qm|Z = z) \leqslant -\theta qm + \psi^{(m)}(\theta, z)$$

for all $\theta \geqslant 0$. As before, if we let $\theta_m(z)$ be the $\theta \geqslant 0$ that maximizes $-\theta qm + \psi^{(m)}(\theta, z)$ for a given z, and define $F_m(z) := -\theta_m(z)qm + \psi^{(m)}(\theta_m(z), z)$, then $G(z) = \ln \mathrm{P}(\mathcal{L}_m > qm|Z = z) \leqslant F_m(z)$. One can then use $F_m(z)$ as a close surrogate to $G(z)$, in order to determine the importance sampling change of measure for Z. Glasserman and Li (2005) develop some new asymptotic regimes and prove asymptotic optimality of the above procedure as $m \to \infty$, again for the homogeneous ($p_k = p$ and $c_k = 1$) single factor case.

Algorithms and asymptotic optimality results for the multi-factor, nonhomogeneous case have been analyzed in Glasserman et al. (2005). Another approach, but without any asymptotic optimality proof has been presented in Morokoff (2004). Algorithms for the “t-copula model” (in contrast to the Gaussian copula model) and related models, have been studied in Bassamboo et al. (2005c) and Kang and Shahabuddin (2005). Bassamboo et al. (2005c) develop sharp asymptotics for the probability of large losses and importance sampling techniques that have bounded relative error in estimating this probability. This analysis is extended to another related and popular performance measure, namely *expected shortfall* or the expected excess loss given that a large loss occurs, in Bassamboo et al. (2005a) (also see Merino and Nyefeler, 2004).

Acknowledgement

This work was partially supported by the National Science Foundation (USA) Grant DMI 03-00044.

References

Ahamed, T.P.I., Borkar, V.S., Juneja, S. (2006). Adaptive importance sampling technique for Markov chains using stochastic approximation. *Operations Research*, in press.

Anantharam, V., Heidelberger, P., Tsoucas, P. (1990). Analysis of rare events in continuous time Markov chains via time reversal and fluid approximation. Report RC 16280, IBM, Yorktown Heights, NY.

Anderson, T. (1984). *An Introduction to Multivariate Statistical Analysis*, 2nd edition. Wiley, New York.

Andradottir, S., Heyman, D.P., Ott, T. (1995). On the choice of alternative measures in importance sampling with Markov chains. *Operations Research* 43 (3), 509–519.

Asmussen, S. (1985). Conjugate processes and the simulation of ruin problems. *Stochastic Process and Applications* 20, 213–229.

Asmussen, S. (1989). Risk theory in a Markovian environment. *Scandinavian Actuarial Journal* 1989, 69–100.

Asmussen, S. (2000). *Ruin Probabilities*. World Scientific, London.

Asmussen, S. (2003). *Applied Probability and Queues*, 2nd edition. Springer-Verlag, New York.

Asmussen, S., Binswanger, K. (1997). Simulation of ruin probabilities for subexponential claims. *ASTIN Bulletin* 27 (2), 297–318.

Asmussen, S., Kroese, D. (2004). Improved algorithms for rare-event simulation with heavy tails. Research report, Department of Mathematical Sciences, Aarhus University, Denmark.

Asmussen, S., Rubinstein, R.Y. (1995). Steady state rare-event simulation in queueing models and its complexity properties. In: Dshalalow, J.H. (Ed.), *Advances in Queueing: Theory, Methods and Open Problems*. CRC Press, New York, pp. 429–462.

Asmussen, S., Binswanger, K., Hojgaard, B. (2000). Rare-event simulation for heavy-tailed distributions. *Bernoulli* 6 (2), 303–322.

Asmussen, S., Kroese, D., Rubinstein, R. (2005). Heavy tails, importance sampling and cross entropy. *Stochastic Models* 21 (1), 57–76.

Avram, F., Dai, J., Hasenbein, J. (2001). Explicit solutions for variational problems in the quadrant. *Queueing Systems* 37, 261–291.

Bahadur, R., Rao, R.R. (1960). On deviations of the sample mean. *The Annals of Mathematical Statistics* 31, 1015–1027.

Bassamboo, A., Juneja, S., Zeevi, A. (2005a). Expected shortfall in credit porfolios with extremal dependence. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1850–1858.

Bassamboo, A., Juneja, S., Zeevi, A. (2005b). On the efficiency loss of state-dependent importance sampling in the presence of heavy tails. *Operations Research Letters*, in press.

Bassamboo, A., Juneja, S., Zeevi, A. (2005c). Portfolio credit risk with extremal dependence. Preprint.

Beck, B., Dabrowski, A., McDonald, D. (1999). A unified approach to fast teller queues and ATM. *Advances in Applied Probability* 31, 758–787.

Bertsekas, D., Tsitsiklis, J. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Athena, MA.

Blanchet, J., Glynn, P. (2005). Efficient simulation for the maximum of a random walk with heavy-tailed increments. Presentation at the 13th INFORMS Applied Probability Conference, July 6–8, 2005, Ottawa, Canada.

Bolia, N., Glasserman, P., Juneja, S. (2004). Function-approximation-based importance sampling for pricing American options. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 604–611.

Booth, T. (1985). Exponential convergence for Monte Carlo particle transport. *Transactions of the American Nuclear Society* 50, 267–268.

Boots, N., Shahabuddin, P. (2000). Simulating $GI/GI/1$ queues and insurance processes with subexponential distributions. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 656–665. Latest version: Research report, IEOR Department, Columbia University, New York.

Boots, N., Shahabuddin, P. (2001). Simulating ruin probabilities in insurance risk processes with subexponential claims. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 468–476.

Borkar, V. (2002). Q-learning for risk-sensitive control. *Mathematics of Operations Research* 27, 294–311.

Borkar, V.S., Meyn, S.P. (2002). Risk-sensitive optimal control for Markov decision processes with monotone cost. *Mathematics of Operations Research* 27, 192–209.

Borkar, V.S., Juneja, S., Kherani, A.A. (2004). Performance analysis conditioned on rare events: An adaptive simulation scheme. *Communications in Information* 3 (4), 259–278.

Bucklew, J.S. (1990). *Large Deviations Techniques in Decision, Simulation and Estimation*. Wiley, New York.

Chang, C.S., Heidelberger, P., Juneja, S., Shahabuddin, P. (1994). Effective bandwidth and fast simulation of ATM in tree networks. *Performance Evaluation* 20, 45–65.

Chistyakov, V.P. (1964). A theorem on sums of independent positive random variables and its applications to branching random processes. *Theory of Probabability and Applications* 9, 640–648.

Collamore, J.F. (1996). Hitting probabilities and large deviations. *The Annals of Probability* 24 (4), 2065–2078.

Collamore, J.F. (2002). Importance sampling techniques for the multidimensional ruin problem for general Markov additive sequences of random vectors. *The Annals of Applied Probability* 12 (1), 382–421.

De Boer, P. (2001). Analysis and efficient simulation of queueing models of telecommunication systems. PhD thesis, University of Twente.

De Boer, P., Nicola, V., Rubinstein, R. (2000). Adaptive importance sampling simulation of queueing networks. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 646–655.

De Boer, P.T., Kroese, D.P., Mannor, S., Rubinstein, R. (2005). A tutorial on the cross-entropy method. *Annals of Operations Research* 134 (1), 19–67.

Dembo, A., Zeitouni, O. (1998). *Large Deviations Techniques and Applications*. Springer-Verlag, New York, NY.

Desai, P., Glynn, P. (2001). A Markov chain perspective on adaptive Monte Carlo algorithms. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 379–384.

Dupuis, P., Wang, H. (2004). Importance sampling, large deviations, and differential games. Preprint.

Embrechts, P., Kluppelberg, C., Mikosch, T. (1997). *Modelling Extremal Events for Insurance and Finance*. Springer-Verlag, Berlin, Heidelberg.

Fang, K.T., Kotz, S., Ng, K. (1987). *Symmetric Multivariate and Related Distributions*. Chapman and Hall, London.

Fishman, G. (2001). *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer-Verlag, Berlin.

Frater, M. (1993). Fast simulation of buffer overflows in equally loaded networks. *Australian Telecommun. Res.* 27 (1), 13–18.

Frater, M., Lennon, T., Anderson, B. (1991). Optimally efficient estimation of the statistics of rare events in queueing networks. *IEEE Transactions on Automatic Control* 36, 1395–1405.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glasserman, P., Kou, S. (1995). Analysis of an importance sampling estimator for tandem queues. *ACM TOMACS* 5, 22–42.

Glasserman, P., Li, J. (2005). Importance sampling for portfolio credit risk. *Management Science* 50 (11), 1643–1656.

Glasserman, P., Wang, Y. (1997). Counterexamples in importance sampling for large deviations probabilities. *The Annals of Applied Probability* 7, 731–746.

Glasserman, P., Heidelberger, P., Shahabuddin, P. (1999). Asymptotically optimal importance sampling and stratification for pricing path-dependent options. *Mathematical Finance* 9, 117–152.

Glasserman, P., Heidelberger, P., Shahabuddin, P. (2000). Variance reduction techniques for estimating value-at-risk. *Management Science* 46, 1349–1364.

Glasserman, P., Heidelberger, P., Shahabuddin, P. (2002). Portfolio value-at-risk with heavy-tailed risk factors. *Mathematical Finance* 9, 117–152.

Glasserman, P., Kang, W., Shahabuddin, P. (2005). Fast simulation of multifactor portfolio credit risk. Working paper, IEOR Department, Columbia University.

Glynn, P., Iglehart, D. (1989). Importance sampling for stochastic simulations. *Management Science* 35, 1367–1392.

Glynn, P., Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Operations Research* 40, 505–520.

Goyal, A., Lavenberg, S. (1987). Modeling and analysis of computer system availability. *IBM Journal of Research and Development* 31 (6), 651–664.

Goyal, A., Shahabuddin, P., Heidelberger, P., Nicola, V., Glynn, P. (1992). A unified framework for simulating Markovian models of highly reliable systems. *IEEE Transactions on Computers* C-41, 36–51.

Grassberger, P. (2002). Go with the winners: A general Monte Carlo strategy. *Computer Physics Communications* 147 (1/2), 64–70.

Grassberger, P., Nadler, W. (2000). "Go with the winners" – Simulations. In: Hoffman, K.H., Schreiber, M. (Eds.), *Computational Statistical Physics: From Billards to Monte Carlo*. Springer-Verlag, Heidelberg, pp. 169–190.

Gupta, G., Finger, C., Bhatia, M. (1997). Credit metrics technical document. Technical report, J.P. Morgan and Co., New York.

Heidelberger, P. (1995). Fast simulation of rare events in queueing and reliability models. *ACM TOMACS* 5 (1), 43–85.

Heyde, C., Kou, S. (2004). On the controversy over tailweight of distributions. *Operations Research Letters* 32, 399–408.

Huang, Z., Shahabuddin, P. (2003). Rare-event, heavy-tailed simulations using hazard function transformations with applications to value-at-risk. In: Chick, S., Sanchez, P.J., Ferrin, D., Morrice, D. (Eds.), *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 276–284. Latest version: Working paper, IEOR Department, Columbia University.

Huang, Z., Shahabuddin, P. (2004). A unified approach for finite dimensional, rare-event Monte Carlo simulation. In: Ignalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1616–1624.

Ignatiouk-Robert, I. (2000). Large deviations of Jackson networks. *The Annals of Applied Probability* 10 (3), 962–1001.

Ignatyuk, I., Malyshev, V., Scherbakov, V. (1994). Boundary effects in large deviations problems. *Russian Mathematical Surveys* 49 (2), 41–99.

Juneja, S. (2001). Importance sampling and the cyclic approach. *Operations Research* 46 (4), 900–912.

Juneja, S. (2003). Efficient rare-event simulation using importance sampling: An introduction. In: Misra, J.C. (Ed.), *Computational Mathematics, Modelling and Algorithms*. Narosa Publishing House, New Delhi, pp. 357–396.

Juneja, S., Nicola, V. (2005). Efficient simulation of buffer overflow probabilities in Jackson networks with feedback. *ACM TOMACS* 15 (4), 281–315.

Juneja, S., Shahabuddin, P. (2001). Efficient simulation of Markov chains with small transition probabilities. *Management Science* 47 (4), 547–562.

Juneja, S., Shahabuddin, P. (2002). Simulating heavy-tailed processes using delayed hazard rate twisting. *ACM TOMACS* 12, 94–118.

Juneja, S., Shahabuddin, P., Chandra, A. (1999). Simulating heavy-tailed processes using delayed hazard-rate twisting. In: Farrington, P.A., Nembhard, H.B., Sturrock, D.T., Evans, G.W. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 420–427.

Juneja, S., Karandikar, R., Shahabuddin, P. (2005). Tail asymptotics and fast simulation of delay probabilities in stochastic PERT networks. *ACM TOMACS*, undergoing review.

Kang, W., Shahabuddin, P. (2005). Fast simulation for multifactor portfolio credit risk in the t-copula model. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1859–1868.

Kollman, C., Baggerly, K., Cox, D., Picard, R. (1999). Adaptive importance sampling on discrete Markov chains. *The Annals of Applied Probability* 9, 391–412.

Kontoyiannis, I., Meyn, S. (2003). Spectral theory and limit theorems for geometrically ergodic Markov processes. *The Annals of Applied Probability* 13, 304–362.

Kotz, S., Kozubowski, T., Podgorski, K. (2001). *The Laplace Distribution and Generalizations: A Revisit with Applications to Communications, Economics, Engineering, and Finance*. Birkhäuser, Boston.

Kroese, D., Nicola, V. (1999). Efficient estimation of overflow probabilities in queues with breakdowns. *Performance Evaluation* 36/37, 471–484.

Kroese, D., Nicola, V. (2002). Efficient simulation of a tandem Jackson network. *ACM TOMACS* 12, 119–141.

Kroese, D., Rubinstein, R. (2004). The transform likelihood ratio method for rare-event simulation with heavy tails. *Queueing Systems* 46, 317–351.

Kushner, H., Yin, G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York.

L'Ecuyer, P., Champoux, Y. (2001). Estimating small cell-loss ratios in atm switches via importance sampling. *ACM TOMACS* 11 (1), 76–105.

Lehtonen, T., Nyrhinen, H. (1992a). Simulating level-crossing probabilities by importance sampling. *Advances in Applied Probability* 24, 858–874.

Lehtonen, T., Nyrhinen, H. (1992b). On asymptotically efficient simulation of ruin probabilities in a Markovian environment. *Scandinavian Actuarial Journal* 1992, 60–75.

Luenberger, D. (1984). *Linear and Non-Linear Programming*, 2nd edition. Addison–Wesley, Reading, MA.

Merino, S., Nyefeler, M. (2004). Applying importance sampling for estimating coherent credit risk contributions. *Quantitative Finance* 4, 199–207.

Merton, R. (1974). On the pricing of corporate debt: The risk structure of interest rates. *Journal of Finance* 29, 449–470.

Morokoff, W. (2004). An importance sampling method for portfolios of credit risky assets. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1668–1676.

Nakayama, M., Nicola, V., Shahabuddin, P. (2001). Techniques for fast simulation of models of highly dependable systems. *IEEE Transactions on Reliability* 50, 246–264.

Ney, P. (1983). Dominating points and the asymptotics of large deviations for random walks on $\Re^d$. *The Annals of Probability* 11, 158–167.

Pakes, A. (1975). On the tails of waiting time distribution. *Journal of Applied Probability* 12, 555–564.

Parekh, S., Walrand, J. (1989). A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control* 34 (1), 54–66.

Randhawa, R.S., Juneja, S. (2004). Combining importance sampling and temporal difference control variates to simulate Markov chains. *ACM TOMACS* 14, 1–30.

Royden, H.L. (1984). *Real Analysis*. Prentice Hall, New York.

Rubinstein, R.Y. (1997). Optimization of computer simulation models with rare events. *European Journal of Operations Research* 99, 89–112.

Rubinstein, R.Y. (1999). Rare-event simulation via cross-entropy and importance sampling. In: *Second Workshop on Rare Event Simulation, RESIM'99*, *Enshede*, *The Netherlands*, pp. 1–17.

Rubinstein, R.Y., Kroese, D.P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte Carlo Simulation and Machine Learning*. Springer-Verlag, New York.

Sadowsky, J. (1991). Large deviations and efficient simulation of excessive backlogs in a $GI/G/m$ queue. *IEEE Transactions on Automatic Control* 36, 1383–1394.

Sadowsky, J. (1996). On Monte Carlo estimation of large deviations probabilities. *The Annals of Applied Probability* 6 (2), 399–422.

Sadowsky, J.S., Bucklew, J. (1990). On large deviation theory and asymptotically efficient Monte Carlo estimation. *IEEE Transactions on Information Theory* 36 (3), 579–588.

Sadowsky, J.S., Szpankowski, W. (1995). The probability of large queue lengths and waiting times in a heterogeneous multiserver queue part I: Tight limits. *Advances in Applied Probability* 27, 532–566.

Shahabuddin, P. (1994). Importance sampling for the simulation of highly reliable Markovian systems. *Management Science* 40, 333–352.

Shahabuddin, P. (2005). An approach to simulation of random walks with heavy-tailed increments. Technical report, Department of IEOR, Columbia University.

Shahabuddin, P., Woo, B. (2004). Conditional importance sampling with applications to value-at-risk simulations. Working paper, IEOR Department, Columbia University.

Shwartz, A., Weiss, A. (1995). *Large Deviations for Performance Analysis*. Chapman and Hall, New York.

Siegmund, D. (1976). Importance sampling in the Monte Carlo study of sequential tests. *The Annals of Statistics* 4, 673–684.

Sigman, K. (1999). A primer on heavy-tailed distributions. *Queueing Systems* 33, 261–275.

Su, Y., Fu, M. (2000). Importance sampling in derivatives security pricing. In: Joines, J.A., Barton, R.R., Jang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 587–596.

Su, Y., Fu, M.C. (2002). Optimal importance sampling in securities pricing. *Journal of Computational Finance* 5, 27–50.

Szechtman, R., Glynn, P. (2002). Rare-event simulation for infinite server queues. In: Yucesan, E., Chen, H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 416–423.

Tong, Y. (1990). *The Multivariate Normal Distribution*. Springer-Verlag, New York.

Vázquez-Abad, F., Dufresne, D. (1998). Accelerated simulation for pricing Asian options. In: Medeiros, D.J., Watson, E.F., Carson, M.S., Manivannan, J.S. (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1493–1500.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13012-1

Chapter 12

Quasi-Random Number Techniques

C. Lemieux

Department of Mathematics and Statistics, University of Calgary, Canada
E-mail: lemieux@math.ucalgary.ca

Abstract

Over the last decade, quasi-Monte Carlo methods have been used as an efficient estimation tool in various high-dimensional applications, particularly in the field of finance. These methods can be seen as a deterministic version of the Monte Carlo method for multidimensional integration, in which quasi-random numbers are used to construct highly-uniform point sets over which the integrand is sampled. This chapter discusses the use of these techniques in simulation.

1 Introduction

At the heart of any simulation study, a reliable source of (pseudo)random numbers – such as those discussed in Chapter 3 – is required to ensure that the statistical output analysis of interest is done correctly. It may thus appear dangerous to replace this pseudorandom source by a stream of *quasi-random* numbers, which, unlike pseudorandom numbers, are designed to produce a highly-uniform sampling in which correlations are allowed. Nevertheless, over the last decade there have been many success stories in which using these quasi-random numbers have produced estimates with a smaller error than their pseudorandom counterpart (Paskov and Traub, 1995; Spanier, 1995; Ninomiya and Tezuka, 1996; Acworth et al., 1997; Caflisch et al., 1997; Morokoff and Caflisch, 1997). These successful applications of quasi-random numbers (particularly those in the field of finance) have received a great deal of attention (Business Week, 1994; The Economist, 1995; New York Times, 1995). But maybe more importantly, they have generated considerable interest among many researchers, thereby resulting in numerous papers where improvements and theories explaining this success have been presented (see, e.g., Caflisch et al., 1997; Morokoff and Caflisch, 1997; Acworth et al., 1997; Owen, 1998a; Sloan and Woźniakowski, 1998; Hickernell, 1998c; Hickernell and Wang, 2001; Owen, 2002; Papageorgiou, 2003; Sobol' and Asostsky, 2003; Wang and Fang, 2003). The purpose of this chapter is to present the general tools and principles required to use quasi-random numbers in simulation. In the remainder of this introduction, we briefly outline the main ideas behind these methods.

To better understand why it makes sense to use quasi-random numbers for simulation, it is useful to formulate the goal of the simulation study in terms of an integral to be estimated. Indeed, typically simulation is used to estimate one or more quantities of the form

$$\mu = \int_{[0,1)^s} f(\mathbf{u})\,\mathrm{d}\mathbf{u}, \tag{1}$$

where f is a real-valued function and s is a positive integer. The function f can be interpreted as the mapping that transforms a set of s numbers between 0 and 1 into an observation of the output quantity of interest, and μ is the expectation of this quantity. In other words, s is the number of pseudorandom numbers that are required in each run of the simulation, and $\mathbf{u}$ is the vector that contains those uniform numbers. If an unbounded number of uniform numbers are required in each simulation run, then s can be considered as infinite. An example describing what this function f is for a simple queueing problem is given in Section 2.

In this context, using n independent simulation runs to estimate μ amounts to using the Monte Carlo (MC) method. In this case, μ is approximated by

$$\hat{\mu}_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{u}_i), \tag{2}$$

where $P_n = \{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ is a set of n independent points uniformly distributed over $[0, 1)^s$. (Of course, in practice these n points are obtained by using a pseudorandom number generator.)

From this point of view, it seems like better estimates of μ could be obtained by choosing these n sampling points more carefully. For example, it is well known that for small dimensions s (say, below 4 or 5), methods like Gaussian quadrature or Simpson's rule can achieve a smaller error than MC (Davis and Rabinowitz, 1984). The use of quasi-random numbers, in what has become known as *quasi-Monte Carlo (QMC) methods*, can also be shown to produce a smaller error than MC, at least asymptotically and for certain classes of functions (see, e.g., Niederreiter, 1992). These methods provide estimators of the same form as the MC estimator (2), but in which the random point set P_n is replaced by a deterministic, highly-uniform point set. They were proposed as an alternative to the MC method shortly after the latter was introduced at the end of the 1940's. One of these early proposals was from Richtmyer (1951), who suggested approximating multidimensional integrals by using the points produced by the sequence $\{n\boldsymbol{\alpha} \bmod 1, n = 1, 2, \dots\}$, where $\boldsymbol{\alpha}$ is an s-dimensional vector of irrational numbers which, together with 1, are linearly independent over the rationals, and the mod 1 operation is taken coordinate-wise. Other constructions that produce highly-uniform point sets were proposed later, including Korobov rules (Korobov, 1959), Halton sequences (Halton, 1960), Sobol' sequences (Sobol', 1967), Faure sequences (Faure, 1982), Niederreiter sequences (Niederreiter, 1987) and Niederreiter–Xing sequences (Niederreiter and Xing, 1996).

The uniformity of the point sets used by QMC methods is often measured by their "discrepancy", that is, by how far their induced empirical distribution departs from a truly uniform distribution. For this reason, another way to refer to quasi-random numbers is to talk about *low-discrepancy point sets or sequences*. In this chapter, we interchangeably use the terms low-discrepancy point sets, highly-uniform point sets, or quasi-random point sets. Typically, a sequence of numbers $\mathbf{u}_1, \mathbf{u}_2, \ldots$ in $[0,1)^s$ is considered to have low discrepancy if $D(P_n) \in \mathrm{O}(\log^s n/n)$, where $P_n = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ contains the first n points of the sequence, and $D(P_n)$ is the so-called *star discrepancy* of P_n, which is defined by

$$D(P_n) = \sup_{\mathbf{v}=(v_1,\ldots,v_s)\in[0,1)^s} \left| \prod_{j=1}^{s} v_j - \frac{1}{n} \left| P_n \cap \prod_{j=1}^{s} [0, v_j) \right| \right|.$$

That is, consider all rectangular boxes anchored at the origin (as determined by $\mathbf{v}$) and compute the difference between the volume of the box and the fraction of points in P_n that fall in the box. Then take the supremum of this difference over all boxes. For example, in one dimension the regular grid $\{0, 1/n, 2/n, \ldots, (n-1)/n\}$ has a discrepancy of $1/n$. For a dimension s larger than one however, the s-dimensional grid $\{(i_1/N, \ldots, i_s/N), 0 \leqslant i_j < N, 1 \leqslant j \leqslant s\}$ with $N = n^{1/s}$ has a discrepancy in $\mathrm{O}(1/N) = \mathrm{O}(n^{-1/s})$, so it is not a low-discrepancy point set. When $s > 1$, the points must be placed more carefully to get a low discrepancy, i.e., not by simply taking the Cartesian product of s one-dimensional low-discrepancy point sets. For example, one kind of construction called a *lattice rule* places the points on the vertices of a *lattice* that intersect the unit hypercube $[0,1)^s$. Figure 1 illustrates the difference between a lattice point set and a regular grid in two dimensions.

There is a large amount of literature on the concept of discrepancy (Niederreiter, 1992; Morokoff and Caflisch, 1994; Hickernell, 1998a; Matoušek, 1998), but we do not discuss it further here since its relevance for simulation is rather limited.

Because of the deterministic nature of the point sets P_n on which QMC methods are based, the error analysis for these methods is often done by trying to derive upper bounds on the deterministic error

$$E_n = |\hat{\mu}_{\mathrm{QMC}} - \mu|,$$

where

$$\hat{\mu}_{\mathrm{QMC}} = \frac{1}{n} \sum_{\mathbf{u}_i \in P_n} f(\mathbf{u}_i),$$

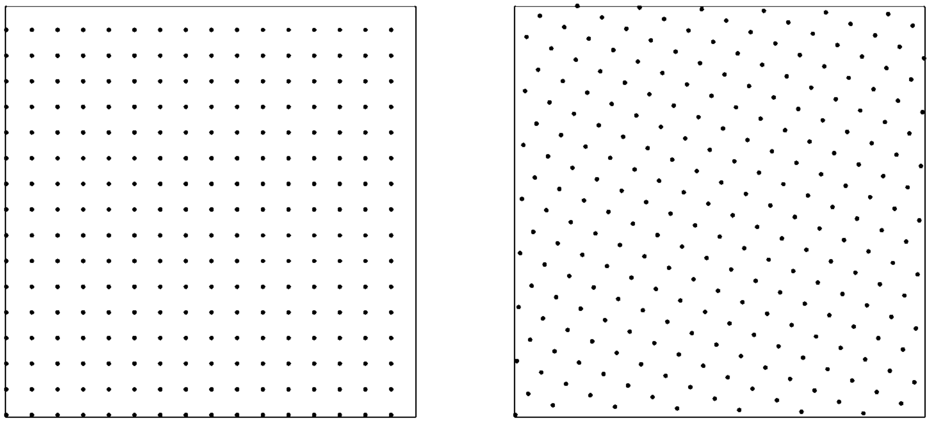


Fig. 1. *Left*: two-dimensional regular grid. *Right*: two-dimensional lattice point set.

and P_n is a highly-uniform point set. For example, one can use the *Koksma–Hlawka inequality*

$$E_n \leqslant D(P_n)V(f), \tag{3}$$

where $D(P_n)$ is the star discrepancy and $V(f)$ is the variation of f in the sense of Hardy and Krause (see Niederreiter, 1992, for more details). This inequality can be used to show that for any function with $V(f) < \infty$, the error of the approximation $\hat{\mu}_{\mathrm{QMC}}$ is in $\mathrm{O}(\log^s n/n)$, which for a fixed s is better than the probabilistic $\mathrm{O}(n^{-1/2})$ associated with MC.

What this means is that asymptotically and for a fairly limited class of functions, the error is smaller for approximations based on low-discrepancy point sets than it is for MC-based estimates. However, one of the problems with this type of analysis is that it does not explain the success of QMC methods on high-dimensional integrands (e.g., $s = 360$ in Paskov and Traub, 1995) since for s even as small as 10, n needs to be about 10^{39} in order to have $\log^s n/n \leqslant n^{-1/2}$. Also, inequalities such as (3) only provide bounds on the error and cannot be used to estimate E_n.

If we go back to the idea of using quasi-random numbers for simulation, it seems like it cannot work since in the simulation context, estimates of the error are important, unbiased estimators are typically preferred, and we often have to work with large values of s. Fortunately, these seemingly major hurdles can be removed if instead of using purely deterministic quasi-random numbers, one uses *randomized quasi-Monte Carlo* (RQMC) methods. The idea here is to randomize a highly-uniform point set P_n so that each of its points follows the uniform distribution over $[0,1)^s$, while preserving the high uniformity of P_n. Many randomization techniques that achieve this have been proposed and used in practice; see Owen (1998a), L'Ecuyer and Lemieux (2002) and the references therein. Since they are better suited for simulation, we will be mostly discussing RQMC methods in this chapter.

Once we have a randomized highly-uniform point set $\widetilde{P}_n$ with the properties mentioned above, each of its points are used to drive one run of the simulation. More precisely, the ith run is driven by the s coordinates of $\mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s})$. Since each $\mathbf{u}_i$ is uniformly distributed over $[0, 1)^s$, the estimator

$$\hat{\mu}_{\mathrm{RQMC}} = \frac{1}{n} \sum_{\mathbf{u}_i \in \widetilde{P}_n} f(\mathbf{u}_i) \tag{4}$$

is unbiased, and its variance can be estimated by creating m copies of $\hat{\mu}_{\mathrm{RQMC}}$ based on m i.i.d. copies of the randomized point set $\widetilde{P}_n$. More details on this procedure are given in Sections 2 and 6.

As we just outlined, with RQMC methods it is possible to estimate the variance of the estimator $\hat{\mu}_{\mathrm{RQMC}}$. One can therefore verify empirically whether the RQMC estimator has a smaller variance than the corresponding MC estimator based on n points. This type of analysis has been done for different kinds of problems, using different quasi-random point sets and randomizations (see, e.g., Tan and Boyle, 2000; L'Ecuyer and Lemieux, 2000; Kollig and Keller, 2002; Lemieux and L'Ecuyer, 2003). In most (if not all) cases, it was observed that the RQMC estimator reduced the variance compared to the MC estimator. Moreover, it can be proved theoretically that in some cases, the variance of the RQMC estimator is smaller than the MC variance (Owen, 1997a, 1997b, 1998b). For this reason, one can think of RQMC methods as *variance reduction techniques* such as those discussed in Chapters 10 and 11. More precisely, because of the nature of the randomized point set $\widetilde{P}_n$, they can be seen as *correlation induction* methods.

The plan for the remainder of this chapter is as follows. We illustrate the main ideas of quasi-random number techniques in Section 2 by using a simple queueing example. In Section 3, we discuss the concept of effective dimension, which is important to understand the success of quasi-random number techniques in practice. The main tools used for designing highly-uniform point sets are presented in Section 4, where we also describe some of the constructions most commonly used in practice. Section 5 is devoted to recurrence-based point sets, which have been discussed in Chapter 3 and provide an especially useful type of highly-uniform point set in practice. In Section 6, we talk about randomization techniques and give some theoretical results supporting the use of RQMC methods as variance reduction techniques. The connections between these results and selection criteria for choosing good parameters for different constructions are also briefly mentioned. Issues that arise when quasi-random point sets are combined with other variance reduction techniques are discussed in Section 7. Finally, current and future avenues of investigation for the use of quasi-random point sets in simulation are presented in Section 8.

2 An example

Consider an $M/M/1$ queue with an arrival rate of $\lambda = 1$ customer per minute and a service rate of $\mu = 1.2$ customers per minute. Suppose the system starts empty, runs for 8 hours, and that we want to estimate the expected average waiting time in the queue for the customers that entered the system during that period. Formally, our goal is to estimate

$$\mu = \mathrm{E}\left(\frac{\sum_{i=1}^{N} w_i}{N}\right),$$

where N is the number of customers that have entered the system in 8 hours and w_i is the waiting time in the queue of the ith customer. The above ratio is defined to be 0 when $N = 0$. We have expressly chosen a finite-horizon measure of performance for this example, just so that the problem would fit more naturally in the general framework outlined in the Introduction.

Using the formulation (1), we see that here the dimension s is unbounded since it depends on the number N of customers entering the system, which is itself unbounded. To describe the function f in (1) that corresponds to this problem, we first use Lindley's equation

$$w_i = \max(0, w_{i-1} + s_{i-1} - a_i), \quad i \geqslant 1, \tag{5}$$

where s_i is the service time of the ith customer, a_i is the interarrival time between the $(i-1)$st and ith customer, and $w_0 = s_0 = 0$.

To write w_i as a function of $\mathbf{u} = (u_1, u_2, \ldots)$, we need to decide what nonuniform generation method (see Chapter 4) to use for the a_i's and the s_i's. With RQMC methods, *inversion* is the most natural choice because it often helps minimizing the number s of uniform numbers required for each simulation. Thus we let

$$a_i = -\frac{\ln(1 - u_{2i-1})}{\lambda} \tag{6}$$

and

$$s_i = -\frac{\ln(1 - u_{2i})}{\mu}. \tag{7}$$

Using (5) along with (6) and (7), it is easy to see that each w_i can be written as a function g_i of $u_1, \ldots, u_{2i-1}$. Similarly, the number N of customers entering the system can be written as

$$N = \sum_{i=1}^{\infty} \mathbf{I}_{a_1+\cdots+a_i<480},$$

where $\mathbf{I}_X$ is the indicator function for the event X. Since a_i is a function of u_{2i-1} for each i, N itself becomes a function of $\mathbf{u}$. We can then write

$$f(\mathbf{u}) = \frac{1}{N(\mathbf{u})} \sum_{i=1}^{N(\mathbf{u})} g_i(u_1, \ldots, u_{2i-1}).$$

A possibly more intuitive way of understanding how f takes the vector $\mathbf{u}$ as input and outputs an observation of the quantity of interest is to use pseudocode as follows:

```
**OneSim**($u_1, u_2, \ldots$)
  TotWait = 0          // Total waiting time
  $w = 0$
  $a = -\ln(1 - u_1)/\lambda$
  $j = 2$              // Indexes the coordinate of **u** to be used next
  time = $a$           // Current time
  NbCust = 1           // Number $N$ of customers that entered so far
  **while**(time < 480) **do**
    $s = -\ln(1 - u_j)/\mu$
    $a = -\ln(1 - u_{j+1})/\lambda$
    NbCust = NbCust + 1
    time = time + $a$
    $w = \max(0, w + s - a)$
    **if**(time < 480) **then** TotWait = TotWait + $w$
    $j = j + 2$
  **return**(TotWait/NbCust)
```

This describes how this simple simulation problem fits the framework of multidimensional integration over $[0, 1)^s$, with $s = \infty$ here. Now we need to explain how quasi-random numbers can be used to estimate μ. Probably the simplest method to describe is the one based on a *randomly shifted Korobov point set*. This is a special case of a randomized recurrence-based point set where the underlying highly-uniform point set is obtained by taking the successive overlapping vectors produced by a linear congruential generator (LCG). The point set is then randomized by adding a random vector $\mathbf{v} = (v_1, v_2, \ldots)$ modulo 1 to each point.

In practice, using the LCG formulation to construct P_n is very useful (see Section 5). Here however, we describe P_n using the "Korobov formulation" (Korobov, 1959). The ith point in the randomized point set $\widetilde{P}_n$ is given by

$$\mathbf{u}_i = \left(\frac{i(1, a, a^2, \ldots) \bmod n}{n} + \mathbf{v} \right) \bmod 1,$$

where $a \in \{1, \ldots, n-1\}$ is the generator of the point set. Hence $u_{i,j} = ((i \times a^{j-1} \bmod n)/n + v_j) \bmod 1$ for each $i = 0, \ldots, n-1$, $j \geqslant 1$.

Putting everything together, the following algorithm estimates μ by using 10 i.i.d. copies of an RQMC estimator based on a randomly shifted Korobov point set. It returns an estimator for μ and its estimated standard deviation. We assume below that Rand01$(\cdot)$ returns a (pseudo)random uniform number between 0 and 1, and that ave($\mathbf{x}$) and std($\mathbf{x}$) return the average and sample standard deviation of the vector $\mathbf{x}$, respectively. In the pseudocode below, the definition of $\mathbf{v}$ suggests it is a vector of infinite length. In practice, it suffices to generate the coordinates only as they are needed. More details are given in Section 6.

```
QueueKorobov
  for $l = 1$ to 10 do
    sum = 0
    $\mathbf{v} = (v_1, v_2, \ldots)$        // where $v_j = \text{Rand01}(\cdot)$
    for $i = 0$ to $n - 1$ do
      sum = sum + OneSim$(u_{i,1}, u_{i,2}, \ldots)$
                                  // where $u_{i,j} = ((1/n)(i \times a^{j-1} \bmod n) + v_j) \bmod 1$
    end for
    $x[l] = \text{sum}/n$
  end for
  return(ave($\mathbf{x}$), std($\mathbf{x}$))   // $\mathbf{x} = (x[1], x[2], \ldots, x[10])$
```

In Table 1, we give the results obtained both with a randomly shifted Korobov point set and the MC method for this example. The generators used are $a = 76$ and $a = 1516$ for $n = 1021$ and $n = 4093$, respectively (L'Ecuyer and Lemieux, 2000). The first number in each entry for "Korobov" (MC) is the mean of 10 i.i.d. copies of $\hat{\mu}_{\text{RQMC}}(\hat{\mu}_{\text{MC}})$ based on n points, and the number in parentheses is the estimated standard deviation of $\hat{\mu}_{\text{RQMC}}(\hat{\mu}_{\text{MC}})$. As we can see there, the RQMC estimator has an estimated standard deviation less than half the size of its MC counterpart. In addition, the computation time required is slightly smaller for this RQMC estimator than for the MC estimator, so in terms of efficiency, the gains are even higher. Note that for this example, since the average number of clients entering the system in 8 hours is 480 and we need to generate two random variables per client, the average number of required components in $\mathbf{u}$ is 960.

Table 1.
Simulation results with a randomly shifted Korobov point set and MC for the $M/M/1$ problem: each entry gives the estimated mean (standard deviation)

n	Korobov	MC
1021	3.84 (0.04)	3.87 (0.10)
4093	3.85 (0.02)	3.85 (0.04)

3 A key concept: Effective dimension

We saw in the previous section an example where the RQMC estimator has an empirical standard deviation less than half the size of its MC counterpart on a problem where the dimension s is close to 1000. The "classical" analysis that we briefly outlined in the Introduction (e.g., the Koksma–Hlawka inequality (3)) cannot explain this kind of result. Moreover, when the dimension s is close to 1000, it seems impossible that a set of only 1021 points could have sampled the unit hypercube $[0,1)^s$ very evenly. To explain the success of the RQMC estimator in this example, we must look at the function f that is integrated, and how it interacts with the point set $\widetilde{P}_n$ used. (Since the features of $\widetilde{P}_n$ that are relevant when studying this interaction are usually inherited from P_n, we consider the unrandomized point set P_n in what follows.)

Intuitively, it makes sense to think that in this example, the waiting time of a customer will probably not be affected by the waiting times of customers that were not close to him in the system, e.g., that arrived one hour before or after him. In terms of the function f, this translates into saying that the interaction of, say, two coordinates u_j and u_l where $|j-l|$ is large does not contribute very much to the variability of f. Going one step further, one could say that for this example, it is reasonable to assume that f can be well approximated by a sum of functions defined over subspaces of low dimensions. Consequently, as long as the set P_n is highly-uniform over these subspaces, the RQMC estimator based on P_n will be a good approximation for f. These ideas can be made more formal by using the *functional ANOVA decomposition* of a function (Hoeffding, 1948), which we now describe.

3.1 A functional ANOVA decomposition

This decomposition allows us to write any square-integrable function f defined over $[0,1)^s$ as a sum

$$f(\mathbf{u}) = \sum_{I \subseteq \{1,\ldots,s\}} f_I(\mathbf{u}),$$

where f_I only depends on those variables u_j such that $j \in I$. This decomposition is such that $\int_{[0,1)^s} f_I(\mathbf{u})\,\mathrm{d}\mathbf{u} = 0$ for any nonempty I, and $f_\emptyset(\mathbf{u}) = \mu$. Also, it is orthogonal, i.e., for any $I \neq J$, we have that

$$\int_{[0,1)^s} f_I(\mathbf{u}) f_J(\mathbf{u})\,\mathrm{d}\mathbf{u} = 0.$$

We assume s is finite here, but ANOVA decompositions can also be used to analyze integrands f of infinite dimension (Owen, 1998a).

Each component is defined as follows (Owen, 1998a): $f_\emptyset(\mathbf{u}) = \mu$, and for $I \neq \emptyset$, $f_I(\mathbf{u}) = \int_{[0,1)^{s-|I|}} f(\mathbf{u})\,\mathrm{d}\mathbf{u}_{I^c} - \sum_{J \subset I} f_J(\mathbf{u})$, where I^c is the complement of I and $\subset$ denotes strict set inclusion. For example, if $f(u_1,u_2) = u_1 + 2u_1u_2^2 +$

u_2^3, then $f_\emptyset(u_1, u_2) = 13/12$ $f_{\{1\}}(u_1, u_2) = 5u_1/3 - 5/6$, $f_{\{2\}}(u_1, u_2) = u_2^2 + u_2^3 - 7/12$, and $f_{\{1,2\}}(u_1, u_2) = 2u_1u_2^2 - 2u_1/3 - u_2^2 + 1/3$.

For each nonempty subset I, the variance σ_I^2 of the ANOVA component f_I is given by

$$\sigma_I^2 = \mathrm{Var}\big(f_I(\mathbf{U})\big) = \int_{[0,1)^s} f_I^2(\mathbf{u})\,\mathrm{d}\mathbf{u}.$$

Because of the orthogonality of the f_I's, we have that $\sum_{I\subseteq\{1,\dots,s\}} \sigma_I^2 = \sigma^2$, where $\sigma^2 = \mathrm{Var}(f(\mathbf{U}))$. Therefore we can view σ_I^2/σ^2 – these are called *sensitivity indices* in Sobol' (2001) – as a measure of the relative importance of the ANOVA component f_I. Note that the best mean-square approximation of f by a sum of d-dimensional (or less) functions is given by $\sum_{I:|I|\leqslant d} f_I$.

In many situations – for example, when evaluating financial products – the function f of interest is such that the components f_I with $|I|$ small are the most important. For instance, Lemieux and Owen (2001) estimate that the one- and two-dimensional components of the 30-dimensional function representing the price f of an Asian option (with specific parameters) contribute about 97% of the variance of f. In some sense, this suggests that for this example, the *effective dimension* of f is 2 rather than 30. In Paskov (1997), this concept of effective dimension was used to explain the success of QMC methods on a 360-dimensional integrand resulting from the evaluation of a Collateralized Mortgage Obligation (CMO). Other investigations of the effective dimension in finance problems are reported in, e.g., Wang and Fang (2003), Wang and Sloan (2003). Because of its crucial importance in our understanding of the success of QMC methods in practice, we now discuss this concept in more detail.

3.2 *Effective dimension*

Our treatment here closely follows Caflisch et al. (1997) and Hickernell (1998b), which provide two definitions of the effective dimension of f based on its ANOVA decomposition.

Definition 1. The effective dimension of f (in proportion p), in the superposition sense, is the smallest integer d_S such that $\sum_{I:|I|\leqslant d_S} \sigma_I^2 \geqslant p\sigma^2$.

In the 30-dimensional Asian option mentioned above, f has an effective dimension of 2 in the superposition sense, in proportion 0.97.

Definition 2. The effective dimension of f (in proportion p), in the truncation sense, is the smallest integer d_T such that $\sum_{I:I\subseteq\{1,\dots,d_T\}} \sigma_I^2 \geqslant p\sigma^2$.

When a function f has an effective dimension of d in the superposition sense in proportion p, it means that it can be approximated by a sum $\tilde{f}$ of

d-dimensional (or less) functions, and that the approximation explains at least $100p\%$ of the variance. Now, if we have a point set P_n such that all projections $P_n(I)$ of the form

$$P_n(I) = \{(u_{i,i_1}, \ldots, u_{i,i_j})\colon \mathbf{u}_i = (u_{i,1}, \ldots, u_{i,s}) \in P_n, I = \{i_1, \ldots, i_j\}\}$$

with $|I| = j \leqslant d$ are highly-uniform, then the estimator $\hat{\mu}_{\text{RQMC}}$ based on P_n should approximate the integral of $\tilde{f}$ very well. Furthermore, if p is close to 1 then $\tilde{f}$ is "close" to f (in a mean-square sense), and thus $\hat{\mu}_{\text{RQMC}}$ should also be a very good estimator of μ. A more formal treatment of this kind of argument can be found in Wang and Fang (2003).

The definition of the effective dimension in the truncation sense can lead to similar arguments describing how the interaction between P_n and f can lead to a successful application of RQMC methods. An important difference with the definition in the superposition sense is that the ordering of the variables $u_1, \ldots, u_s$ matters here. The motivation behind this definition is that for some QMC constructions such as Sobol' and Halton sequences, it has been observed that the projections $P_n(I)$ deteriorate as the indices in I increase (see, e.g., Morokoff and Caflisch, 1994). In order for the estimator $\hat{\mu}_{\text{RQMC}}$ to be good for such point sets, the function f to be integrated must be such that the components f_I with $|I|$ or $i_{\min I} = \min\{j\colon j \in I\}$ large should be unimportant. Not all constructions for highly-uniform point sets have this undesirable feature. For instance, recurrence-based point sets have projections $P_n(I)$ that do not deteriorate as $i_{\min I}$ increases (all things being equal elsewhere) because they are *dimension-stationary*. This means that the projections $P_n(I)$ depend only on the spacing between the indices in I, e.g., $P_n(\{1,3,4\}) = P_n(\{2,4,5\}) = P_n(\{10,12,13\})$.

Another possible definition for the effective dimension is given in L'Ecuyer and Lemieux (2000).

Definition 3. The effective dimension of f (in proportion p), in the successive-dimensions sense, is the smallest integer d_{SD} such that

$$\sum_{i=1}^{s-d_{\text{SD}}+1} \sum_{I:I\subseteq\{i,\ldots,i+d_{\text{SD}}-1\}} \sigma_I^2 \geqslant p\sigma^2.$$

As for d_T, the value of d_{SD} depends on the ordering of the variables u_j because of the restriction on the subsets I that the range $r_I = \max\{j\colon j \in I\} - \min\{j\colon j \in I\}$ be upper bounded by d_{SD}. The motivation behind this definition is that in some problems, especially in the simulation context, a variable u_j does not interact very strongly with variables u_l such that $|l - j|$ is large. We made this point (intuitively) at the beginning of this section, when discussing the $M/M/1$ example. In addition, when designing highly-uniform point sets in high dimensions, it is not possible in practice to make sure that all projections

$P_n(I)$ are good, even if we only consider those with $|I| \leqslant d$ for some $d < s$. However, if the point set is designed for classes of functions that are believed to have a small effective dimension in the above sense, then it is reasonable to decrease the number of projections considered by adding the corresponding restriction on the range of I. We refer the reader to Chapter 3 for more on selection criteria that use such restrictions.

3.3 Relevance in the simulation context

In the context of simulation, it often happens that the integrand f in (1) has a very large nominal dimension, but a small or moderate effective dimension. Intuitively, the reason for this is that although a large number of input variables are used in each simulation, each of them usually interacts strongly with only a few other variables. For this reason, RQMC methods can often be used successfully in the simulation context. For example, in Lemieux and L'Ecuyer (1999), RQMC methods are used to reduce the variance by factors between 2 and 5 for a ruin probability problem where the nominal dimension is 8000.

It is important to note that in the formulation (1), there are two things that can influence the size of the effective dimension of f: (i) which nonuniform generation methods are used; (ii) how the variables u_j are assigned to the input variables in the problem. As we mentioned in Section 2, with RQMC methods it is preferable to use inversion for generating nonuniform random variables, as it often helps minimizing the dimension (nominal, and thus effective) of f. Inversion also helps with the assignment mentioned in (ii), in the same way it helps achieve synchronization when common random numbers or antithetic variates are used (Law and Kelton, 2000, pp. 586–599). When using inversion, there is often a "natural" way of assigning the coordinates u_j to the input variables: one simply assigns the u_j's in the chronological order produced by the simulation. This is what we did in Section 2: u_1 was used to generate the first interarrival time, then u_2 was used for the service time of the first client, then u_3 for the next interarrival time, etc. Assignment done in this way typically contributes to reduce the effective dimension (in the successive-dimensions sense) since the u_j's whose indices are far apart are associated with events that are far apart in time and thus do not interact strongly together. This is especially true when the simulation has regenerative cycles (see Chapter 16). In this case, if all the variables u_j associated with one cycle have successive indices, then we can say that roughly, the effective dimension (in the successive-dimensions sense) is bounded above by the expected number of input variables associated with one regenerative cycle.

The assignment choice may not be that crucial in some cases and, as mentioned before, it does not affect the value of the effective dimension in the superposition sense. Also, if the point set P_n is such that for any fixed size d, each projection $P_n(I)$ with $|I| = d$ is of the same quality, then the assignment can be done arbitrarily. This is true when P_n is a set of i.i.d. uniform points as in the MC method, but typical highly-uniform point sets do not have this property. Instead, for a fixed size $|I|$, the projections $P_n(I)$ with either a small minimal index $i_{\min I}$ or range r_I are usually more uniform. In this case, the assignment choice may make a difference, but what may really help with such point sets is to use *dimension reduction techniques*, which we now discuss.

3.4 Reducing the effective dimension

For some problems, it is possible to use techniques that can reduce the effective dimension. One kind of application where this is especially true is when the problem requires the simulation of an underlying Brownian motion. In this case, one can use the *Brownian Bridge Technique* (Caflisch et al., 1995; Morokoff and Caflisch, 1997), which uses the Brownian bridge property to generate the Brownian motion's steps in an arbitrary order.

Let $0 \leqslant t_1 < \cdots < t_s$. For a standard Brownian motion $B(\cdot)$, the usual way of generating a path $B(t_1), \dots, B(t_s)$ from a uniform point $\mathbf{u} = (u_1, \dots, u_s)$ is as follows:

```
$t_0 = B(t_0) = 0$
**for** $j = 1$ **to** $s$
    $B(t_j) = B(t_{j-1}) + \sqrt{t_j - t_{j-1}}\Phi^{-1}(u_j)$    // where $\Phi(x) = \mathrm{P}(\mathrm{N}(0,1) \leqslant x)$
**end for**
```

An alternative idea is to try using the first few coordinates of $\mathbf{u}$ to specify as much as possible the behavior of $B(\cdot)$, so that functions of $B(\cdot)$ will (hopefully) have a small effective dimension in the truncation sense. The Brownian bridge technique does that by first generating $B(t_s)$, then $B(t_{\lfloor s/2\rfloor})$, then $B(t_{\lfloor s/4\rfloor})$ and $B(t_{\lfloor 3s/4\rfloor})$, and so on. This can be done easily since the Brownian bridge property tells us that for any $u < v < w$, we have that $B(v) \mid (B(u) = a, B(w) = b)$ has a normal distribution with mean $a(w-v)/(w-u) + b(v-u)/(w-u)$ and variance $(v-u)(w-v)/(w-u)$. Similar techniques can be used to generate Poisson processes, as discussed in Fox (1999).

For Brownian motion, the above technique can be generalized by observing that the standard method to generate $B(\cdot)$ can be written as

$$\begin{pmatrix} B(t_1) \\ \vdots \\ B(t_s) \end{pmatrix} = \mathbf{A} \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix},$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ & & \vdots & & \\ 1 & 1 & 1 & \dots & 1 \end{pmatrix},$$

if we assume that $t_j - t_{j-1} = 1$ for each $j = 1, \dots, s$, and the $z_j = \Phi^{-1}(u_j)$ are i.i.d. standard normal variables. Replacing $\mathbf{A}$ by a matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{B}^{\mathrm{T}} =$

$\mathbf{AA}^{\mathrm{T}} =: \mathbf{\Sigma}$ is called a *generalized Brownian bridge technique* in (Morokoff and Caflisch, 1997). For example, Acworth et al. (1997) use a principal components analysis to define $\mathbf{B}$, i.e., they take $\mathbf{B} = \mathbf{PD}^{1/2}$, where $\mathbf{P}$'s columns are formed by the eigenvectors of the covariance matrix $\mathbf{\Sigma}$, and $\mathbf{D}$ is a diagonal matrix containing the corresponding eigenvalues of $\mathbf{\Sigma}$ in decreasing order. This method was shown to numerically outperform the Brownian bridge technique in Acworth et al. (1997), but its computation time is much longer since to simulate n Brownian motion paths, it runs in $\mathrm{O}(ns^2)$ rather than the $\mathrm{O}(ns)$ required for the standard and Brownian bridge methods. Following this work, Åkesson and Lehoczy (2000) proposed a modification reducing the computation time for the principal components' method.

It is important to be aware that these dimension reduction methods do not provide estimators with reduced variance for all problems. They aim at improving the simulation of the Brownian motion paths, but they do not take into account how these paths contribute to the value of the function to be estimated. For example, Papageorgiou (2002) provides numerical results showing that for a certain type of digital option in finance, the Brownian bridge technique produces estimators with a far worse error than the standard method does.

For simulation problems that do not rely on Brownian motions or Poisson processes, it is still possible to come up with dimension reduction techniques. For example, using *Conditional Monte Carlo* (CMC) typically amounts to reducing the number of input variables that need to be generated, thereby resulting in an automatic reduction of the (nominal) dimension. See for example L'Ecuyer and Lemieux (2000), where CMC and RQMC are used to simulate a stochastic activity network.

4 Constructing quasi-random point sets

As seen in the previous sections, the success of RQMC methods in practice relies on the availability of highly-uniform point sets that interact in a constructive way with the integrand. Of course, the applicability of these methods would be greatly diminished if one had to choose a specific point set for each problem in order for this interaction to work properly. A more reasonable approach is to have a few good constructions whose parameters have been chosen so that for a large number of integrands, they should produce good results, that is, estimators with lower variance than MC.

Currently, most of the highly-uniform point sets used in practice come from two families of constructions. *Lattice Rules* (Niederreiter, 1992; Sloan and Joe, 1994), which generalize the early construction of Korobov (1959), and *digital nets and sequences* (Niederreiter, 1992; Tezuka, 1995), which generalize the Sobol' sequence. Further, we describe these families and some of their most widely used constructions. We do not discuss polynomial integration lattices and their extensions, and refer the reader to Lemieux and L'Ecuyer (2003), Niederreiter (2003) and L'Ecuyer (2004) for information on those.

4.1 Lattice rules

Lattice rules construct approximations for (1) by using a *lattice point set* of the form $P_n = L \cap [0, 1)^s$, with L an *integration lattice* defined as

$$L = \left\{ \mathbf{x} = \sum_{j=1}^{s} z_j \mathbf{v}_j \colon \mathbf{z} = (z_1, \ldots, z_s) \in \mathbb{Z}^s \right\}, \tag{8}$$

where the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_s \in \mathbb{R}^s$ are linearly independent and form a *basis* for L. In other words, a lattice is obtained by taking all integer linear combinations of the vectors in its basis. In addition, L must contain $\mathbb{Z}^s$ – this is what makes it an *integration* lattice – which implies that each $\mathbf{v}_j$ contains only rational numbers. As an example, a Korobov rule is based on a lattice with a basis of the form $\mathbf{v}_1 = (1, a, a^2, \ldots, a^{s-1})/n \bmod 1$, $\mathbf{v}_j = \mathbf{e}_j$ for $j = 2, \ldots, s$, where $\mathbf{e}_j$ is a vector of zeros with a one in the jth position.

A given lattice does not have a unique basis. However, its *rank* r and *invariants* $n_1, \ldots, n_r$ are uniquely determined integers having the property that (i) P_n can be written as

$$P_n = \left\{ \frac{j_1}{n_1}\mathbf{z}_1 + \cdots + \frac{j_r}{n_r}\mathbf{z}_r \colon 0 \leqslant j_l < n_l, l = 1, \ldots, r \right\}, \tag{9}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_r$ are linearly independent integer vectors; (ii) n_{j+1} divides n_j, for $j = 1, \ldots, r-1$ and $n = n_1, \ldots, n_r$ (Sloan and Joe, 1994). The formulation (9) is easier to work with than (8) is when the time comes to choose specific lattice constructions. More precisely, one can choose the rank and invariants beforehand, and then proceed to a search for "good" vectors $\mathbf{z}_1, \ldots, \mathbf{z}_r$ according to some criterion. When the rank is larger than one, these vectors are not uniquely determined in (9), and so computer searches must be done carefully in order to avoid too much duplication. See Sloan and Walsh (1990) for examples of computer searches where the rank is 2.

In practice, most lattice rules are based on lattices of rank 1. In this case, the lattice point set can be written as

$$P_n = \left\{ \frac{i}{n}(z_1, \ldots, z_s) \bmod 1 \colon 0 \leqslant i < n \right\},$$

where $(z_1, \ldots, z_s) \in \mathbb{Z}^s$ is the *generating vector*. The parameters to be determined are thus $z_1, \ldots, z_s$, and there is no loss of generality in assuming that $z_1 = 1$ and $1 \leqslant z_j < n$ for $j = 2, \ldots, s$. Restricting the search to rank-1 rules certainly simplifies the search procedure. In addition, rank-1 rules are *fully projection regular*, a property meaning that each projection $P_n(I)$ contains n distinct points. Lattices of higher rank do not provide point sets with this desirable property.

Korobov rules are an example of rank-1 rules. As seen in Section 2, only one parameter, the generator a of P_n, needs to be specified. This provides additional speedup for the search procedure. Moreover, Korobov point sets are dimension-stationary. Tables of good generators a for various values of n can be found in, e.g., Haber (1983), L'Ecuyer and Lemieux (2000).

In our discussion of lattice rules, we have assumed that the number of points n was fixed. In applications where a certain level of accuracy is desired, the required number of evaluation points may not be known in advance. In this case, constructions whose number of points can be increased indefinitely – just like with digital sequences – are better suited. Recently, lattice rules whose size n can be extended have been studied (Hickernell and Hong, 1997; Hickernell et al., 2001; Hickernell and Niederreiter, 2003) and parameters for such sequences can be found in Hickernell et al. (2001). Another recent advance in the theory of lattice rules is the development of component-by-component constructions for rank-1 rules that minimize a certain criterion (more precisely, a worst-case error in some class of functions). Tables of parameters for such point sets can be found in, e.g., Sloan et al. (2002), Sloan and Rezstov (2002). The constructions studied in Sloan et al. (2002) have the additional advantage of achieving a *strong tractability error bound* in some space of functions. This roughly means that the number of points required to keep the error bounded as the dimension s increases does not grow exponentially with s.

4.2 Digital nets and sequences

Before giving a general description of digital nets and sequences, let us first present a one-dimensional low-discrepancy sequence due to van der Corput (1935). This sequence makes use of the *radical-inverse function* φ_b, a central tool in the design of digital nets. Let n be a nonnegative integer and consider its unique digital expansion in base b given by

$$n = \sum_{i=0}^{\infty} a_i b^i, \tag{10}$$

where $0 \leqslant a_i < b$, and $a_i = 0$ for all sufficiently large i, i.e., the sum in (10) is actually finite. Then we have

$$\varphi_b(n) = \sum_{i=0}^{\infty} a_i b^{-i-1}.$$

The van der Corput sequence in base 2 $u_0, u_1, \dots$ is defined by

$$u_n = \varphi_2(n), \quad n \geqslant 0,$$

i.e., the nth term in the sequence is obtained by applying the radical inverse function in base 2 to n.

To generalize this construction to point sets or sequences of higher dimensions, at least two ideas can be exploited: (i) for each dimension $j = 1, \dots, s$, apply a different linear transformation to the coefficients a_i before applying φ_b; (ii) use a different base b for each dimension. The Sobol' sequence is based on the first idea, while the Halton sequence uses the second one. More generally, applying the first idea in an arbitrary base b is what gives rise to the general notion of digital net and sequence, which we now describe following Niederreiter (1992). (Our description is not as general as the one given there, but general enough to cover most constructions used in practice.)

Let b be a prime and $k \geqslant 0$ be an integer. A *digital net in base* b with $n = b^k$ points is a point set P_n defined by *s generating matrices* $\mathbf{C}_1, \dots, \mathbf{C}_s$ of size $k \times k$, with entries in $\mathbb{Z}_b$. The jth coordinate $u_{i,j}$ of the ith point is obtained as $u_{i,j} = \sum_{l=1}^{k} y_{j,l} b^{-l}$, where

$$\begin{bmatrix} y_{j,1} \\ \vdots \\ y_{j,k} \end{bmatrix} = \mathbf{C}_j \begin{bmatrix} a_0 \\ \vdots \\ a_{k-1} \end{bmatrix}, \tag{11}$$

and the a_l's come from the digital expansion $i = \sum_{l=0}^{k-1} a_l b^l$ of i in base b. So $u_{i,j}$ is obtained by first applying a linear transformation determined by the matrix $\mathbf{C}_j$ to the coefficients $a_0, \dots, a_{k-1}$ in the digital expansion of i, and then by applying the radical-inverse function to these transformed coefficients $y_{j,1}, \dots, y_{j,k}$. It is possible to define digital nets in a base b that is not prime. One simply needs to choose a commutative ring R of cardinality b. The generating matrices then contain elements in that ring R, and bijections going from $\mathbb{Z}_b$ to R and from R to $\mathbb{Z}_b$ must be applied to the a_i's and the $y_{j,l}$, respectively. This setup can actually be used for a prime base b as well.

Related to digital nets are *digital sequences*, which are infinite sequences of points rather than point sets P_n of finite size n. Points from a digital sequence are obtained as above, but with generating matrices of infinite size.

We now describe three special cases of digital sequences: the Halton, Sobol' and Faure sequences. As we outlined previously, an s-dimensional Halton sequence is obtained by juxtaposing s van der Corput sequences in different bases. More precisely, the jth coordinate of the ith point is obtained as

$$u_{ij} = \varphi_{b_j}(i),$$

where the b_j, $j = 1, \dots, s$, are integers larger than 1 typically chosen as the first s prime numbers. Formally, this is not a digital sequence, but we feel it is important to present this construction because of its wide use in practice (e.g., see Spanier, 1995; Kollig and Keller, 2002). Many improvements have been proposed for this sequence since its introduction in 1960 (Braaten and Weller, 1979; Struckmeier, 1995; Tuffin, 1998).

The Sobol' sequence is a digital sequence in base 2. It was proposed by Sobol' (1967) before the concept of digital net was defined in Niederreiter

(1987). Thus, its definition does not directly use generating matrices, although it can be defined that way. Instead, it relies on *direction numbers* $v_{j,1}, v_{j,2}, \ldots$ that must be chosen for each coordinate $j = 1, \ldots, s$. These direction numbers are rational numbers of the form

$$v_{j,l} = \frac{m_{j,l}}{2^l} = \sum_{p=1}^{l} v_{j,l,p} 2^{-p}, \quad l \geqslant 1,$$

where $m_{j,l}$ is an odd integer smaller than 2^l. Also needed is a primitive polynomial $f_j(z) = z^q + \alpha_{j,1} z^{q-1} + \cdots + \alpha_{j,q}$ over $\mathbb{F}_2$, the finite field with two elements, for each $j = 1, \ldots, s$. The method described in Sobol' (1967) chooses $f_j(z)$ to be the jth polynomial in a list of primitive polynomials over $\mathbb{F}_2$ sorted by increasing degree.

For a given j, if $f_j(z)$ is of degree $q(j)$, then only the first $q(j)$ direction numbers $v_{j,1}, \ldots, v_{j,q(j)}$ must be chosen. The next ones are obtained through the recurrence

$$v_{j,l} = \alpha_{j,1} v_{j,l-1} \oplus \cdots \oplus \alpha_{j,q(j)-1} v_{j,l-q(j)+1} \oplus \frac{v_{j,l-q(j)}}{2^{q(j)}}, \quad l > q(j),$$

where $\oplus$ denotes a bit-by-bit exclusive-or operation and $v_{j,l-q}/2^{q(j)}$ means that the binary expansion of $v_{j,q(j)}$ is shifted by $q(j)$ positions to the right. The jth coordinate of the ith point in the sequence is then defined as

$$u_{i,j} = a_0 v_{j,1} \oplus \cdots \oplus a_{k-1} v_{j,k},$$

where $a_0, \ldots, a_{k-1}$ are the coefficients in the binary expansion of i.

Hence the parameters that must be chosen are the $q(j)$ first direction numbers for each $j = 1, \ldots, s$. Equivalently, one must specify $m_{j,1}, \ldots, m_{j,q(j)}$ for each j. Sobol' and Levitan (1976) give a table of values of $m_{j,l}$ for $j \leqslant 40$ that is used in the implementation given by Bratley and Fox (1988). Other implementations of the Sobol' sequence can be found in the Finder software (*http://www.cs.columbia.edu/~ap/html/finder.html*), the SamplePack software (*http://www.uni-kl.de/AG-Heinrich/SamplePack.html*) and the RandQMC library (*http://www.math.ucalgary.ca/~lemieux/randqmc.html*, Lemieux et al., 2002).

The quality of digital nets and sequences is often measured by a quantity called t. Niederreiter (1992) calls an s-dimensional digital net in base b with b^m points a (t, m, s)-net, where t refers to this quality parameter. The smaller t is, the more uniform the net is. Similarly, an s-dimensional digital (t, s)-sequence is a digital sequence such that for each $m \geqslant 0$, the first b^m points form a (u, m, s)-net, for some $u \leqslant t$. Sobol' (1967) calls his sequence an LP_τ-sequence, where τ refers to this value of t. The Faure sequence is a digital sequence with $t = 0$ (Faure, 1982). For this property to hold, we must have $b \geqslant s$, and thus the base b increases with s. This sequence uses the Pascal matrix $\mathbf{P}$, whose entry on the kth row and lth column is $\binom{k}{l}$ for $k \geqslant l$, and 0 otherwise. More precisely, the generating matrix $\mathbf{C}_j$ for the Faure sequence is taken to be the transpose of the Pascal matrix $\mathbf{P}$ raised to the power $j-1$, and where each entry is reduced modulo b. Tezuka (1995) has introduced a construction called a *generalized Faure sequence*, in which $\mathbf{C}_j = \mathbf{A}_j(\mathbf{P}^{\mathrm{T}})^{j-1}$, where each $\mathbf{A}_j$ is a non-singular lower-triangular matrix. Specific choices for these $\mathbf{A}_j$ can be found in Tezuka and Tokuyama (1994) and Faure (2001).

5 Recurrence-based point sets

We have seen in Chapter 3 how to define highly-uniform point sets by using a pseudorandom number generator (PRNG). Let us first recall briefly how this construction works.

Let R be a finite set. Assume we have a PRNG based on a transition function ψ from R to R and an output function g from R to $[0,1)$. The sequence $x_0, x_1, \ldots,$ obtained using the recurrence

$$x_i = \psi(x_{i-1}), \quad i \geqslant 1, \tag{12}$$

has a period of at most $|R|$. The associated *recurrence-based point set* is defined by

$$P_n = \bigl\{\bigl(g(x_0), g(x_1), \ldots, g(x_{s-1})\bigr)\colon x_0 \in R\bigr\}.$$

Thus $n = |R|$ for this construction, as long as g is one-to-one. As seen in Chapter 3, different kinds of PRNGs can been used in this manner to define highly-uniform point sets. For example, recall from Section 2 that using an LCG yields a Korobov point set.

Let us now discuss properties of recurrence-based point sets that are useful in practice. First, these point sets are fully projection regular and dimension-stationary. In addition, they can handle problems where the integrand f has an infinite dimension. This is because a given point in P_n is of the form

$$\mathbf{u}_i = \bigl(g(x_0), \ldots, g(x_{s-1})\bigr)$$

for some $x_0 \in R$. Therefore the size s of $\mathbf{u}_i$ can be increased indefinitely simply by continuing to run the recurrence (12). Of course, this means that if s exceeds $|R|$, the coordinates in $\mathbf{u}_i$ will eventually repeat. This would be bad if no randomization was applied, since it would mean that a simulation based on $\mathbf{u}_i$ reuses the same input numbers. However, by randomizing P_n appropriately, there will not be any repetition. For example, suppose we use a random shift $\mathbf{v} = (v_1, v_2, \ldots, v_s)$ as in Section 2 to obtain $\tilde{\mathbf{u}}_i = (\mathbf{u}_i + \mathbf{v}) \bmod 1$. Then even if, say, $u_{i,j} = u_{i,j+|R|}$, since $v_j \neq v_{j+|R|}$ with probability 1, then $\tilde{u}_{i,j} \neq \tilde{u}_{i,j+|R|}$ with probability 1 as well.

This brings us to the discussion of some implementation ideas that can be useful in the simulation context. As noted above, recurrence-based point sets can be used for infinite-dimensional problems. One way to implement this is to choose an a priori size S for the dimension of the point set. When a point

$\mathbf{u}_i$ needs to be of a larger dimension than S (i.e., the simulation based on that point needs more than S random numbers), we just need to know which element of R produced the last coordinate of $\mathbf{u}_i$, i.e., which $x_{S-1} \in R$ is such that $u_{i,S} = g(x_{S-1})$. Then we can obtain $u_{i,S+1}, u_{i,S+2}, \ldots,$ by computing $x_S, x_{S+1}, \ldots$ from (12) and putting $u_{i,S+l+1} = g(x_{S+l})$, for $l \geqslant 0$.

Another possibility is to use an alternative definition for P_n, which says that P_n contains the overlapping vectors of all the cycles of a PRNG. More precisely,

$$P_n = \bigcup_{x_0 \in R} C(x_0), \tag{13}$$

where

$$C(x_0) = \bigl\{\bigl(g(x_i), g(x_{i+1}), \ldots, g(x_{i+s-1})\bigr)\colon 0 \leqslant i < \tau(x_0)\bigr\},$$

and $\tau(x_0)$ is the period of the sequence $\{x_l, l \geqslant 0\}$. Of course, if $x, y \in R$ are in the same cycle, then $C(x) = C(y)$, so the number of distinct sets in the union (13) is equal to the number of distinct cycles of the PRNG. The idea is then to store these distinct cycles into separate arrays and record their respective lengths. Arbitrary coordinates $u_{i,j}$ can then be easily retrieved by doing a simple arithmetic computation to figure out which cycle contains $\mathbf{u}_i$ and in what position. Here is an example to illustrate how it works.

Example 4. Suppose we have a PRNG over a finite set R of size 128, with 4 cycles of respective length $\tau_1 = 1$, $\tau_2 = 7$, $\tau_3 = 15$ and $\tau_4 = 105$. Let u_l^c be the lth output number in the cth cycle. Then P_n can be defined so that for $i = 0, \ldots, n-1$,

$$\mathbf{u}_i = \bigl(u_{l(i)}^{c(i)}, u_{l(i)+1 \bmod \tau_{c(i)}}^{c(i)}, \ldots, u_{l(i)+s-1 \bmod \tau_{c(i)}}^{c(i)}\bigr),$$

where

$$\bigl(c(i), l(i)\bigr) = \begin{cases} (1,1) & \text{if } i = 0, \\ (2,i) & \text{if } 1 \leqslant i < 8, \\ (3, i-7) & \text{if } 8 \leqslant i < 23, \\ (4, i-22) & \text{if } i \geqslant 23. \end{cases}$$

This alternative implementation has been used in Dembeck (2003). It is very useful in cases where the dimension is very large, e.g., $s \gg n$, because instead of having to store an s-dimensional vector representing the point, coordinates can be generated online, and only the n numbers output by the PRNG in all its cycles need to be stored. This is also useful when one needs to partition points into blocks that are not used at the same time. This can happen when information is sequentially gathered through time to update the simulation, such as in *sequential Monte Carlo methods* (Doucet et al., 2001), or in the context of *perfect simulation* (Propp and Wilson, 1996). When s is not too large and points of a fixed dimension s are used one after the other in the simulation, the $\mathbf{u}_i$'s should instead be generated by shifting the coordinates by one position in the current cycle (unless the index i corresponds to a change of cycle $c(i)$, in which case the point has to be filled in from the first s numbers of the next cycle). This kind of implementation is used in the QMC portion of the SSJ package (L'Ecuyer et al., 2002).

6 Randomization techniques and variance results

We already saw in Section 2 a simple randomization method that consists in adding a random uniform vector $\mathbf{v} \in [0, 1)^s$ to each point in P_n. Although this method can be used to randomize any highly-uniform point set (Lemieux and L'Ecuyer, 2000; Morohosi and Fushimi, 2000; Kollig and Keller, 2002), it was originally proposed for lattice rules by Cranley and Patterson (1976). Below, we discuss this randomization and two other ones that are better suited for digital nets. But first, we outline a general framework that includes most randomization methods.

6.1 General principles

Let $P_n = \{\mathbf{u}_0, \dots, \mathbf{u}_{n-1}\}$ be a highly-uniform point set, and let $\mathbf{v}$ be a uniform random vector in some space Ω. To randomize P_n, a randomization function $r : \Omega \times [0, 1)^s \to [0, 1)^s$ is needed in order to obtain a randomized version $\widetilde{P}_n = \{\tilde{\mathbf{u}}_0, \dots, \tilde{\mathbf{u}}_{n-1}\}$ of P_n defined by

$$\tilde{\mathbf{u}}_i = r(\mathbf{v}, \mathbf{u}_i).$$

For example, in the Cranley–Patterson method, $\Omega = [0, 1)^s$ and $r(\mathbf{v}, \mathbf{u}_i) = (\mathbf{u}_i + \mathbf{v}) \bmod 1$.

As mentioned in the Introduction, the function r should be chosen so that (i) $r(\mathbf{v}, \mathbf{u})$ is uniformly distributed over $[0, 1)^s$ for each $\mathbf{u}$, and (ii) $\widetilde{P}_n$ has the same highly-uniform properties as P_n. Another property that holds for most randomization techniques is that they can be written in a product form, i.e., Ω is of the form $\Omega = \widetilde{\Omega} \times \cdots \times \widetilde{\Omega}$, and $r(\mathbf{v}, \mathbf{u}) = (\tilde{r}(v_1, u_1), \dots, \tilde{r}(v_s, u_s))$, where $v_j \in \widetilde{\Omega}$, and $\tilde{r} : \widetilde{\Omega} \times [0, 1) \to [0, 1)$. This property simplifies the implementation when s is infinite, since when additional coordinates are required, one simply needs to generate additional random components v in $\widetilde{\Omega}$. In practice, this can be done by using similar ideas as those discussed in Section 5, i.e., one can fix an a priori bound S, generate and store $v_1, \dots, v_S$, memorize the state x_S of the PRNG used to generate the v_j. Then, when additional coordinates need to be randomized, set the PRNG's state to x_S and generate v_{S+1}, v_{S+2}, etc.

6.2 Shift modulo 1

This is the Cranley–Patterson method that we have already discussed. It is easy to see that this randomization method satisfies property (i) above. As for (ii), assume that P_n is a lattice point set and that its uniformity is measured by the largest distance between adjacent parallel hyperplanes that together cover the points in P_n (this is the quantity measured by the spectral test discussed in Chapter 3). Obviously, the uniformity of $\widetilde{P}_n$ is then the same as that of P_n. This randomization also satisfies the product-form property.

In the case where $P_n = L \cap [0, 1)^s$ is a lattice point set, the variance of the estimator $\hat{\mu}_{\text{RQMC}}$ based on a randomly shifted lattice point set $\widetilde{P}_n$ satisfies the following (L'Ecuyer and Lemieux, 2000):

Proposition 5. *If f is square-integrable, then for a randomly shifted lattice point set we have*

$$\text{Var}(\hat{\mu}_{\text{RQMC}}) = \sum_{\mathbf{0} \neq \mathbf{h} \in L^{\perp}} \left|\hat{f}(\mathbf{h})\right|^2,$$

where $L^{\perp} = \{\mathbf{h} \in \mathbb{R}^s\colon\ \mathbf{h} \cdot \mathbf{u}_i \in \mathbb{Z}$ for each $\mathbf{u}_i \in P_n\}$ is the dual lattice of L and $\hat{f}(\mathbf{h})$ is the Fourier coefficient of f in $\mathbf{h}$.

In comparison, for the MC estimator we have

$$\text{Var}(\hat{\mu}_{\text{MC}}) = \frac{1}{n} \sum_{\mathbf{0} \neq \mathbf{h} \in \mathbb{Z}^s} \left|\hat{f}(\mathbf{h})\right|^2.$$

It can be shown that the dual lattice $L^{\perp}$ has n times less elements than $\mathbb{Z}^s$ (see, e.g., Sloan and Joe, 1994). Using this fact and the above proposition, we can see that for the randomly shifted lattice estimator to have a smaller variance than the MC estimator, the squared Fourier coefficients of f must be smaller on average over the dual lattice than they are over $\mathbb{Z}^s$. In practice, f is often such that the largest Fourier coefficients are those associated with small **h**'s. Hence one could argue that to get an estimator with a smaller variance than the MC estimator, the lattice L on which P_n is based should be chosen so that its dual lattice does not contain short vectors. Recall from Chapter 3 that the quantity measured by the spectral test is also equal to the inverse of the length of the shortest vector in the dual lattice $L^{\perp}$. This means that using criteria based on the spectral test such as those discussed in Chapter 3 can be justified from the above variance analysis.

We refer the reader to L'Ecuyer and Lemieux (2002) and the references therein for more details on criteria that can be used to choose lattice point sets, and to Morohosi and Fushimi (2000) for a variance analysis of randomly shifted digital nets. Note that although this randomization method does not provide a guaranteed variance reduction, it is often used in practice because of its simplicity.

6.3 Digital shift

The analog of the Cranley–Patterson method for a digital net in base b is to use a digital shift in base b (see L'Ecuyer and Lemieux (2002) and the references therein). In this case, the randomization function r is defined by

$$r(\mathbf{v}, \mathbf{u}) = \mathbf{u} \bigoplus_b \mathbf{v},$$

where $\mathbf{v} \in \Omega = [0, 1)^s$ and $\bigoplus_b$ is a coordinate-wise addition of the base b expansion of $\mathbf{u}$ and $\mathbf{v}$, i.e., for $\mathbf{u} = (\sum_{l=1}^{\infty} u_{1,l} b^{-l}, \dots, \sum_{l=1}^{\infty} u_{s,l} b^{-l})$ and $\mathbf{v} = (\sum_{l=1}^{\infty} v_{1,l} b^{-l}, \dots, \sum_{l=1}^{\infty} v_{s,l} b^{-l})$, we have that

$$\mathbf{u} \bigoplus_b \mathbf{v} = \left(\sum_{l=1}^{\infty} \big((u_{1,l} + v_{1,l}) \bmod b\big) b^{-l}, \right.$$
$$\left. \dots, \sum_{l=1}^{\infty} \big((u_{s,l} + v_{s,l}) \bmod b\big) b^{-l} \right).$$

The uniformity of P_n as measured by the parameter t and the *resolution* (see Chapter 3) is preserved by this randomization.

The variance of a digitally shifted net can be analyzed in a way that closely mimics the results for shifted lattice rules: the Fourier coefficients are instead coefficients from the Walsh expansion of f, and a notion of *dual space* of the digital net P_n replaces the dual lattice in the above analysis. We refer the reader to L'Ecuyer and Lemieux (2002) and L'Ecuyer (2004) for more details. In particular, more details can be found in these papers on selection criteria for digital nets that can be related to the variance expression obtained in that context.

6.4 Scrambling

The scrambling method was proposed by Owen (1995) to randomize digital nets. Alternative scrambling strategies have been studied in Matoušek (1998), Faure et al. (2001), Hong and Hickernell (2003), Faure and Tezuka (2003) and Owen (2003). Here we briefly discuss two of them, using the terminology from Owen (2003): the *nested uniform scrambling* of Owen (1995) and the *affine matrix scrambling* of Matoušek (1998) (called *random linear scrambling* in that paper).

Let P_n be a digital net in base b. Nested uniform scrambling applies random uniform permutations to the digits of each coordinate $u_{i,j}$ in its base b expansion. More precisely, if we write

$$u_{i,j} = \sum_{l=1}^{\infty} y_l b^{-l},$$

then the nested uniform scrambling maps $u_{i,j}$ to

$$\tilde{u}_{i,j} = \sum_{l=1}^{\infty} \pi_{\cdot, y_1, \dots, y_{l-1}}(y_l) b^{-l},$$

where $\pi_{\cdot,y_1,\ldots,y_{l-1}}(\cdot)$ is a random uniform permutation of $[0, 1, \ldots, b-1]$ that depends on $y_1, \ldots, y_{l-1}$. All permutations used for a given j and across different values of j are independent.

The cost in time and space required by this method is quite large, and for that reason several alternatives have been proposed. Here we discuss only one and refer the reader to Owen (2003) for additional information. In affine matrix scrambling, we have that $\tilde{u}_{i,j} = \sum_{l=1}^{\infty} \tilde{y}_l b^{-l}$, with

$$\tilde{y}_l = \sum_{k=1}^{l} L_{l,k} y_k + d_l,$$

where $L_{l,l} \in \mathbb{Z}_b \setminus \{0\}$, $L_{l,k} \in \mathbb{Z}_b$ for $k < l$, and $d_l \in \mathbb{Z}_b$ are randomly and uniformly chosen. An alternative description is to say that with this scrambling, each generating matrix is multiplied by a random lower-triangular matrix of infinite size with elements in $\mathbb{Z}_b$ and nonzero elements on the diagonal. A digital shift is then performed.

These two scrambling methods produce estimators $\hat{\mu}_{\mathrm{RQMC}}$ satisfying the following proposition.

Proposition 6 (Owen, 1998b; Hong and Hickernell, 2003; Owen, 2003). *Under either scrambling method described above, if P_n is a (t, m, s)-net in base b, then for any square-integrable function f,*

$$\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}) \leqslant b^t \left(\frac{b+1}{b-1}\right)^s \frac{\sigma^2}{n}.$$

If f is sufficiently smooth (i.e., its mixed partial derivatives satisfy a Lipschitz condition: see Owen (1998b) *for the details), then*

$$\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}) \in \mathrm{O}\big(n^{-3} \log^s n\big).$$

What this result says is that estimators based on scrambled nets cannot do worse than the MC estimator (up to a constant with respect to n), and that for a function that is smooth enough, the variance is in $\mathrm{O}(n^{-3} \log^s n)$, which is significantly better than the $\mathrm{O}(n^{-1})$ that we get for the MC estimator.

7 Combination with other variance reduction techniques

It is a natural idea in simulation to try combining different variance reduction techniques, hoping that their individual beneficial effect will add up (or even be enhanced) in the combination. Since RQMC methods can be seen as a way to produce estimators with lower variance than MC, it makes sense to try combining them with more standard variance reduction techniques such as those discussed in Chapters 10 and 11. As one would expect, care must be taken when doing so if we want to make sure that the combination will not backfire and produce undesirable effects. To illustrate this point, we briefly discuss in this section the combination of RQMC methods with control variates (CV), as studied in Hickernell et al. (2005).

When RQMC is combined with CV, the optimal control variate coefficients are not necessarily the same as when CV is used with plain MC. The idea is as follows. Assume we have a control variable described by a function $g(\mathbf{u})$ over $[0, 1)^s$, with known expectation μ_g. We are looking for the value of β that minimizes the variance of

$$\hat{\mu}_{\mathrm{RQMC}}(f) + \beta\big(\mu_g - \hat{\mu}_{\mathrm{RQMC}}(g)\big), \tag{14}$$

where $\hat{\mu}_{\mathrm{RQMC}}(f)$ and $\hat{\mu}_{\mathrm{RQMC}}(g)$ are the RQMC estimators of μ and μ_g, respectively. As seen in Chapter 10, with plain MC the optimal β is given by $\beta^*_{\mathrm{MC}} = \mathrm{Cov}(f(\mathbf{u}), g(\mathbf{u}))/\mathrm{Var}(g(\mathbf{u}))$. In the RQMC context, it is clear from (14) that the optimal β should instead be

$$\beta^*_{\mathrm{RQMC}} = \frac{\mathrm{Cov}(\hat{\mu}_{\mathrm{RQMC}}(f), \hat{\mu}_{\mathrm{RQMC}}(g))}{\mathrm{Var}(\hat{\mu}_{\mathrm{RQMC}}(g))},$$

which in general is not equal to β^*_{MC}. The optimal β^*_{RQMC} can be estimated using i.i.d. replications of $\hat{\mu}_{\mathrm{RQMC}}(f)$ and $\hat{\mu}_{\mathrm{RQMC}}(g)$, as is done to estimate the variance of $\hat{\mu}_{\mathrm{RQMC}}(f)$. Although there are some pathological cases for which using β^*_{MC} instead of β^*_{RQMC} can lead to a significant increase in variance, experiments in Hickernell et al. (2005) suggest that for problems frequently encountered in practice, using the wrong optimal coefficient may not significantly affect the variance. We refer the reader to Hickernell et al. (2005) for more on this topic, including asymptotic variance analyses and alternative ways to estimate β^*_{RQMC}.

8 Future directions

There is still much work to do in order to make quasi-random numbers a well-known and widely used tool in simulation. First, it would be useful to have simulation packages that incorporate several RQMC methods as alternatives to the use of pseudorandom numbers. Progress in this direction may come from a package developed by Pierre L'Ecuyer and his collaborators (L'Ecuyer et al., 2002).

As seen in Section 3, the interplay between the function f to be integrated and the highly-uniform point set P_n used plays an important role in the success of RQMC methods. Being able to extract important features of f in an online fashion would thus be useful for choosing an appropriate construction for P_n. For example, being able to know what are the important projections of f in its ANOVA decomposition, or what its effective dimension is in some sense would be helpful. Recent papers (Lemieux and Owen, 2001; Jiang and Owen, 2003; Liu and Owen, 2003; Wang and Fang, 2003; Wang and Sloan, 2003) take some steps in that direction.

The search for better constructions is another area where new contributions would be helpful. For example, the Sobol' sequence is widely used but in practice, its implementation requires an upper bound on the dimension s. It would be desirable to have alternative constructions in base 2 (for a quick implementation) with an infinite number of points and dimension. Niederreiter (2003) gives theoretical results proving the existence of such sequences, but no constructions have been suggested so far.

Acknowledgements

The author wishes to thank the editors and anonymous reviewer for their helpful comments and suggestions. This work was supported by NSERC Grant #RGP238959.

References

Acworth, P., Broadie, M., Glasserman, P. (1997). A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In: Hellekalek, P., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 1–18.

Åkesson, F., Lehoczy, J.P. (2000). Path generation for quasi-Monte Carlo simulation of mortgage-backed securities. *Management Science* 46, 1171–1187.

Braaten, E., Weller, G. (1979). An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration. *Journal of Computational Physics* 33, 249–258.

Bratley, P., Fox, B.L. (1988). Algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software* 14 (1), 88–100.

Business Week (1994). Suddenly, number theory makes sense to industry. *Business Week* (June 20, 1994), 172–174.

Caflisch, R.E., Moskowitz, B. (1995). Modified Monte Carlo methods using quasi-random sequences. In: Niederreiter, H., Shiue, P.J.-S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 1–16.

Caflisch, R.E., Morokoff, W., Owen, A.B. (1997). Valuation of mortgage-backed securities using Brownian bridges to reduce effective dimension. *The Journal of Computational Finance* 1 (1), 27–46.

Cranley, R., Patterson, T.N.L. (1976). Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis* 13 (6), 904–914.

Davis, P., Rabinowitz, P. (1984). *Methods of Numerical Integration*, 2nd edition. Academic Press, New York.

Dembeck, D.C. (2003). Dynamic numerical integration using randomized quasi-Monte Carlo methods. Master's thesis, University of Calgary.

Doucet, A., de Freitas, N., Gordon, N. (Eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, Berlin.

Faure, H. (1982). Discrépance des suites associées à un système de numération. *Acta Arithmetica* 61, 337–351.

Faure, H. (2001). Variations on $(0, s)$-sequences. *Journal of Complexity* 17, 741–753.

Faure, H., Tezuka, S. (2001). Another random scrambling of digital (t, s) sequences. In: Fang, K.-T., Hickernell, F., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 242–256.

Faure, H., Tezuka, S. (2003). I-binomial scrambling of digital nets and sequences. *Journal of Complexity* 19, 744–757.

Fox, B.L. (1999). *Strategies for Quasi-Monte Carlo*. Kluwer Academic Publishers, Boston, MA.

Haber, S. (1983). Parameters for integrating periodic functions of several variables. *Mathematics of Computation* 41, 115–129.

Halton, J.H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik* 2, 84–90.

Hickernell, F.J. (1998a). A generalized discrepancy and quadrature error bound. *Mathematics of Computation* 67, 299–322.

Hickernell, F.J. (1998b). Lattice rules: How well do they measure up? In: Hellekalek, P., Larcher, G. (Eds.), *Random and Quasi-Random Point Sets*. *Lecture Notes in Statistics*, vol. 138. Springer-Verlag, New York, pp. 109–166.

Hickernell, F.J. (1998c). What affects accuracy of quasi-Monte Carlo quadrature? In: Niederreiter, H., Spanier, J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, New York, pp. 16–55.

Hickernell, F.J., Hong, H.S. (1997). Computing multivariate normal probabilities using rank-1 lattice sequences. In: Golub, G.H., Lui, S.H., Luk, R.J., Plemmons, F.T. (Eds.), *Proceedings of the Workshop on Scientific Computing (Hong Kong)*. Springer-Verlag, Singapore, pp. 209–215.

Hickernell, F.J., Niederreiter, H. (2003). The existence of good extensible rank-1 lattices. *Journal of Complexity* 19, 286–300.

Hickernell, F.J., Wang, X. (2001). The error bounds and tractability of quasi-Monte Carlo methods in infinite dimension. *Mathematics of Computation* 71, 1641–1661.

Hickernell, F.J., Hong, H.S., L'Ecuyer, P., Lemieux, C. (2001). Extensible lattice sequences for quasi-Monte Carlo quadrature. *SIAM Journal on Scientific Computing* 22, 1117–1138.

Hickernell, F.J., Lemieux, C., Owen, A.B. (2005). Control variates for quasi-Monte Carlo. *Statistical Science* 20, 1–31.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distributions. *Annals of Mathematical Statistics* 19, 293–325.

Hong, H., Hickernell, F. (2003). Algorithm 823: Implementing scrambled digital sequences. *ACM Transactions on Mathematical Software* 29, 95–109.

Jiang, T., Owen, A.B. (2003). Quasi-regression with shrinkage. *Mathematics and Computers in Simulation* 62, 231–241.

Kollig, T., Keller, A. (2002). Efficient bidirectional path tracing by randomized quasi-Monte Carlo integration. In: Fang, T., Hickernell, F.J., Niederreiter, H. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 290–305.

Korobov, N.M. (1959). The approximate computation of multiple integrals. *Doklady Akademii Nauk SSSR* 124, 1207–1210 (in Russian).

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

L'Ecuyer, P. (2004). Polynomial integration lattices. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 2002*. Springer-Verlag, Berlin, pp. 73–98.

L'Ecuyer, P., Lemieux, C. (2000). Variance reduction via lattice rules. *Management Science* 46 (9), 1214–1235.

L'Ecuyer, P., Lemieux, C. (2002). Recent advances in randomized quasi-Monte Carlo methods. In: Dror, M., L'Ecuyer, P., Szidarovszki, F. (Eds.), *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*. Kluwer Academic Publishers, Boston, pp. 419–474.

L'Ecuyer, P., Meliani, L., Vaucher, J. (2002). SSJ: A framework for stochastic simulation in Java. In: Dror, M., Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, New York, pp. 234–242.

Lemieux, C., L'Ecuyer, P. (1999). Lattice rules for the simulation of ruin problems. In: *Proceedings of the 1999 European Simulation Multiconference, vol. 2*. The Society for Computer Simulation, Ghent, Belgium, pp. 533–537.

Lemieux, C., L'Ecuyer, P. (2000). A comparison of Monte Carlo, lattice rules and other low-discrepancy point sets. In: Niederreiter, H., Spanier, J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, Berlin, pp. 326–340.

Lemieux, C., L'Ecuyer, P. (2003). Randomized polynomial lattice rules for multivariate integration and simulation. *SIAM Journal on Scientific Computing* 24 (5), 1768–1789.

Lemieux, C., Owen, A. (2001). Quasi-regression and the relative importance of the ANOVA components of a function. In: Fang, T., Hickernell, F.J. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods 2000*. Springer-Verlag, Berlin, pp. 331–344.

Lemieux, C., Cieslak, M., Luttmer, K. (2002). RandQMC user's guide: A package for randomized quasi-Monte Carlo methods in C. Technical Report 2002-712-15, Department of Computer Science, University of Calgary.

Liu, R., Owen, A.B. (2003). Estimating mean dimensionality. Technical Report 2003-14, Department of Statistics, Stanford University.

Matoušek, J. (1998). On the L_2-discrepancy for anchored boxes. *Journal of Complexity* 14, 527–556.

Morohosi, H., Fushimi, M. (2000). A practical approach to the error estimation of quasi-Monte Carlo integration. In: Niederreiter, H. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods 1998*. Springer-Verlag, Berlin, pp. 377–390.

Morokoff, W.J., Caflisch, R.E. (1994). Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing* 15, 1251–1279.

Morokoff, W.J., Caflisch, R.E. (1997). Quasi-Monte Carlo simulation of random walks in finance. In: Hellekalek, P. (Ed.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 340–352.

New York Times (1995). From I.B.M., help in intricate trading. *The New York Times* (September 25, 1995).

Niederreiter, H. (1987). Point sets and sequences with small discrepancy. *Monatshefte für Mathematik* 104, 273–337.

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 63. SIAM, Philadelphia.

Niederreiter, H. (2003). The existence of good extensible polynomial lattice rules. *Monatshefte für Mathematik* 139, 295–307.

Niederreiter, H., Xing, C. (1996). Low-discrepancy sequences and global function fields with many rational places. *Finite Fields and Their Applications* 2, 241–273.

Ninomiya, S., Tezuka, S. (1996). Toward real-time pricing of complex financial derivatives. *Applied Mathematical Finance* 3, 1–20.

Owen, A.B. (1995). Randomly permuted (t, m, s)-nets and (t, s)-sequences. In: Niederreiter, H., Shiue, S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 299–317.

Owen, A.B. (1997a). Monte Carlo variance of scrambled equidistribution quadrature. *SIAM Journal on Numerical Analysis* 34 (5), 1884–1910.

Owen, A.B. (1997b). Scrambled net variance for integrals of smooth functions. *The Annals of Statistics* 25 (4), 1541–1562.

Owen, A.B. (1998a). Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation* 8 (1), 71–102.

Owen, A.B. (1998b). Scrambling Sobol and Niederreiter–Xing points. *Journal of Complexity* 14, 466–489.

Owen, A.B. (2002). Necessity of low effective dimension. Manuscript.

Owen, A.B. (2003). Variance and discrepancy with alternative scramblings. *ACM Transactions on Modeling and Computer Simulation* 13, 363–378.

Papageorgiou, A. (2002). The Brownian bridge does not offer a consistent advantage in quasi-Monte Carlo integration. *Journal of Complexity* 18 (1), 171–186.

Papageorgiou, A. (2003). Sufficient conditions for fast quasi-Monte Carlo convergence. *Journal of Complexity* 19 (3), 332–351.

Paskov, S. (1997). New methodologies for valuing derivatives. In: Pliska, S., Dempster, M. (Eds.), *Mathematics of Derivative Securities*. Cambridge University Press, Cambridge, UK, pp. 545–582.

Paskov, S., Traub, J. (1995). Faster valuation of financial derivatives. *Journal of Portfolio Management* 22, 113–120, see also *http://www.cs.columbia.edu/~ap/html/information.html*.

Propp, J.G., Wilson, D.B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms* 9 (1–2), 223–252.

Richtmyer, R.D. (1951). On the evaluation of definite integrals and a quasi-Monte Carlo method based on properties of algebraic numbers. Technical Report LA-1342, Los Alamos Scientific Laboratory.

Sloan, I.H., Joe, S. (1994). *Lattice Methods for Multiple Integration*. Clarendon Press, Oxford.

Sloan, I.H., Rezstov, A. (2002). Component-by-component construction of good lattice rules. *Mathematics of Computation* 71, 263–273.

Sloan, I.H., Walsh, L. (1990). A computer search of rank 2 lattice rules for multidimensional quadrature. *Mathematics of Computation* 54, 281–302.

Sloan, I.H., Woźniakowski, H. (1998). When are quasi-Monte Carlo algorithms efficient for high dimensional integrals? *Journal of Complexity* 14, 1–33.

Sloan, I.H., Kuo, F.Y., Joe, S. (2002). On the step-by-step construction of quasi-Monte Carlo integration rules that achieve strong tractability error bounds in weighted Sobolev spaces. *Mathematics of Computation* 71, 1609–1640.

Sobol', I.M. (1967). The distribution of points in a cube and the approximate evaluation of integrals. *Computational Mathematics and Mathematical Physics (Moscow)* 7, 86–112.

Sobol', I.M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation* 55, 271–280.

Sobol', I.M., Asostsky, D.I. (2003). One more experiment on estimating high-dimensional integrals by quasi-Monte Carlo methods. *Mathematics and Computers in Simulation* 62, 255–263.

Sobol', I.M., Levitan, Y.L. (1976). The production of points uniformly distributed in a multidimensional cube. Preprint 40, Institute of Applied Mathematics, USSR Academy of Sciences. (In Russian.)

Spanier, J. (1995). Quasi-Monte Carlo methods for particle transport problems. In: Niederreiter, H., Shiue, S. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*. *Lecture Notes in Statistics*, vol. 106. Springer-Verlag, New York, pp. 121–148.

Struckmeier, J. (1995). Fast generation of low-discrepancy sequences. *Journal of Computational and Applied Mathematics* 91, 29–41.

Tan, K.S., Boyle, P. (2000). Applications of randomized low discrepancy sequences to the valuation of complex securities. *Journal of Economic Dynamics and Control* 24, 1747–1782.

Tezuka, S. (1995). *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, Norwell, MA.

Tezuka, S., Tokuyama, T. (1994). A note on polynomial arithmetic analogue of Halton sequences. *ACM Transactions on Modeling and Computer Simulation* 4, 279–284.

The Economist (1995). Is Monte Carlo Bust? *The Economist* (August 12, 1995), 63.

Tuffin, B. (1998). A new permutation choice in Halton sequences. In: Hellekalek, P., Larcher, G., Niederreiter, H., Zinterhof, P. (Eds.), *Monte Carlo and Quasi-Monte Carlo Methods*. *Lecture Notes in Statistics*, vol. 127. Springer-Verlag, New York, pp. 427–435.

van der Corput, J.G. (1935). Verteilungsfunktionen: I, II. *Koninklijke Nederlandse Akademie van Wetenschappen, Proceedings* 38, 813–821, 1058–1066.

Wang, X., Fang, K.-T. (2003). The effective dimension and quasi-Monte Carlo integration. *Journal of Complexity* 19, 101–124.

Wang, X., Sloan, I.H. (2003). Why are high-dimensional finance problems of low effective dimension? Applied Mathematics Report AMR03/20, University of New South Wales.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13013-3

Chapter 13

Analysis for Design

Ward Whitt

Department of Industrial Engineering and Operations Research, Columbia University, USA
E-mail: ww2040@columbia.edu

Abstract

In this chapter we discuss analysis for the design of simulation experiments. By that we mean, not the traditional (important) methods to design statistical experiments, but rather techniques that can be used, before a simulation is conducted, to estimate the computational effort required to obtain desired statistical precision for contemplated simulation estimators. In doing so, we represent computational effort by simulation time, and that in turn by either the number of replications or the run length within a single simulation run. We assume that the quantities of interest will be estimated by sample means. In great generality, the required length of a single simulation run can be determined by computing the asymptotic variance and the asymptotic bias of the sample means. Existing theory supports this step for a sample mean of a function of a Markov process. We would prefer to do the calculations directly for the intended simulation model, but that usually is prevented by model complexity. Thus, as a first step, we usually approximate the original model by a related Markovian model that is easier to analyze. For example, relatively simple diffusion-process approximations to estimate required simulation run lengths for queueing models can often be obtained by heavy-traffic stochastic-process limits.

1 Introduction

Simulations are *controlled experiments*. Before we can run a simulation program and analyze the output, we need to choose a simulation model and decide what output to collect; i.e., we need to *design* the simulation experiment. Since (stochastic) simulations require statistical analysis of the output, it is often appropriate to consider the perspective of *experimental design*, e.g., as in Cochran and Cox (1992), Montgomery (2000) and Wu and Hamada (2000).

Simulations are also *explorations*. We usually conduct simulations because we want to learn more about a complex system we inadequately understand. To head in the right direction, we should have some well-defined goals and questions when we start, but we should expect to develop new goals and questions as we go along. When we think about experimental design, we should observe that the time scale for computer simulation experiments tends to be much shorter than the time scale for the agricultural and medical experiments that led to the theory of experimental design. With the steadily increasing power of computers, computer simulation has become a relatively rapid process. After doing one simulation, we can quickly revise it and conduct others. Therefore, it is almost always best to think of simulation as an *iterative process*: We conduct a simulation experiment, look at the results and find as many new questions as answers to our original questions. Each simulation experiment suggests subsequent simulation experiments. Through a succession of these experiments, we gradually gain the better understanding we originally sought. To a large extent, it is fruitful to approach simulation in the spirit of *exploratory data analysis*, e.g., as in Tukey (1977), Velleman and Hoaglin (1981) and Chapter 1 of NIST/SEMATECH (2003).

Successful simulation studies usually involve an artful mix of both experimental design and exploration. We would emphasize the spirit of exploration, but we feel that some experimental design can be a big help. When we plan to hike in the mountains, in addition to knowing what peak we want to ascend, it is also good to have a rough idea how long it will take to get there: Should the hike take two hours, two days or two weeks?

That is just the kind of rough information we need for simulations. A major purpose of simulation experiments, often as a means to other ends, is to estimate unknown quantities of interest. When we plan to conduct a simulation experiment, in addition to knowing what quantities we want to estimate, it is also good to have a rough idea how long it will take to obtain a reliable estimate: Should the experiment take two seconds, two hours or two years?

As in Whitt (1989), in this chapter we discuss techniques that can be used, before a simulation is conducted, to estimate the computational effort required to obtain desired statistical precision for contemplated simulation estimators. Given information about the required computational effort, we can decide what cases to consider and how much computational effort to devote to each. We can even decide whether to conduct the experiment at all. We can also decide if we need to exploit *variance-reduction techniques* (or *efficiency-improvement techniques*), see Chapters 10–12 and 14–16.

The theoretical analysis we discuss should complement the experience we gain from conducting many simulation experiments. Through experience, we learn about the amount of computational effort required to obtain desired statistical precision for simulation estimators in various settings. The analysis and computational experience should reinforce each other, giving us better judgment. The methods in this chapter are intended to help develop more reliable expectations about statistical precision. We can use this knowledge, not only to design better simulation experiments, but also to evaluate simulation output analysis, done by others or ourselves.

At first glance, the experimental design problem may not seem very difficult. First, we might think, given the amazing growth in computer power, that the computational effort rarely needs to be that great, but that is not the case: Many simulation estimation goals remain out of reach, just like many other computational goals; e.g., see Papadimitriou (1994).

Second, we might think that we can always get a rough idea about how long the runs should be by doing one *pilot run* to estimate the required simulation run lengths. However, there are serious difficulties with that approach. First, such a preliminary experiment requires that we set up the entire simulation before we decide whether or not to conduct the experiment. Nevertheless, if such a sampling procedure could be employed consistently with confidence, then the experimental design problem would indeed not be especially difficult. In typical simulation experiments, we want to estimate steady-state means for several different input parameters. Unfortunately, doing a pilot run for one set of parameters may be very misleading, because the required run length may change dramatically when the input parameters are changed.

To illustrate how misleading one pilot run can be, consider a simulation of a *queueing model*. Indeed, we shall use queueing models as the context examples throughout the chapter. Now consider the simulation of a single-server queue with unlimited waiting space (the $G/G/1/\infty$ model, e.g., see Cohen (1982) or Cooper (1982)), with the objective of estimating the mean steady-state (or long-run average) number of customers in the system, as a function of basic model data such as the arrival stochastic process and the service-time distribution. This queueing experimental design problem is interesting and important primarily because a uniform allocation of data over all cases (parameter values) is not nearly appropriate. Experience indicates that, for given statistical precision, the required amount of data increases dramatically as the traffic intensity ρ (arrival rate divided by the service rate) increases toward the critical level for stability and as the arrival-and-service variability (appropriately quantified) increases. For example, the required simulation run length to obtain 5% relative error (width of confidence interval divided by the estimated mean) at a high traffic intensity such as 0.95 tends to be 100 times greater than at a lower traffic intensity such as 0.50. (The operative formula underlying this rough estimate is $f(\rho) \equiv (1-\rho)^{-2}$; note that $f(0.95)/f(0.50) = 400/4 = 100$. If we consider the more extreme case $\rho = 0.995$, then the factor is 10,000. If we used a criterion of absolute error instead of relative error, then the operative formula becomes even more impressive: then $f(\rho) \equiv (1-\rho)^{-4}$.)

In this queueing example, and throughout this paper, we use simulation time as our characterization of computational effort. (For a theoretical discussion of this issue, see Glynn and Whitt, 1992.) Some computational experience or additional experiments on the selected computer are needed to convert simulation time into computational effort. Since there is a degree of freedom in choosing the measuring units for time, it is important to normalize these time units. For example, in a queueing model we might measure time in terms of the number of arrivals that enter the system or we might stipulate that a representative service-time distribution has mean 1. On the positive side, focusing on required simulation time has the advantage that it yields characterizations of computational effort that are independent of the specific computer used to conduct the simulation. It seems best to try to account for that important factor separately.

We assume that the quantities of interest will be estimated by *sample means*. (There are other estimation procedures; e.g., see Chapters 8 and 9.) With sample means, in great generality the required amount of simulation time can be determined by computing quantities called the *asymptotic variance* and the *asymptotic bias* of the sample means. Thus, we want to estimate these quantities before conducting the simulation. In general, that is not so easy to do, but existing theory supports this step for a sample mean of a function of a Markov process. However, the stochastic processes of interest in simulation models are rarely Markov processes. Thus, it is usually necessary to first approximate the given stochastic process by a Markov process in order to apply the techniques in this paper.

It is important to approach this approximation step with the right attitude. Remember that we usually only want to obtain a rough estimate of the required simulation run length. Thus, we may well obtain the desired insight with only a very rough approximation. We do not want this analysis step to take longer than it takes to conduct the simulation itself. So we want to obtain the approximation quickly and we want to be able to do the analysis quickly. Fortunately, it is often possible to meet these goals.

For example, we might be interested in simulating a non-Markovian open network of single-server queues. We might be interested in the queue-length distributions at the different queues. To obtain a rough estimate of the required simulation run length, we might first solve the traffic-rate equations to find the net arrival rate at each queue. That step is valid for non-Markovian queueing networks as well as Markovian queueing networks; e.g., see Chen and Yao (2001), Kelly (1979) or Walrand (1988). Given the net arrival rate at each queue, we can calculate the traffic intensity at each queue by multiplying the arrival rate times the mean service time. Then we might focus on the *bottleneck queue*, i.e., the queue with the highest traffic intensity. We do that because the overall required run length is usually determined by the bottleneck queue. Then we analyze the bottleneck queue separately (necessarily approximately).

We might approximate the bottleneck queue by the Markovian $M/M/1$ queue with the same traffic intensity, and apply the techniques described in this paper to the Markovian queue-length process in order to estimate the required simulation run length. Alternatively, to capture the impact of the arrival and service processes beyond their means, we might use heavy-traffic limit theorems to approximate the queue-length process of the bottleneck queue by a reflected Brownian motion (RBM); e.g., see Chen and Yao (2001) and Whitt (2002). We then apply the techniques described in this paper to the limiting RBM, which is also a Markov process. By the methods described in these last two paragraphs, we can treat quite general queueing-network models, albeit roughly.

Here is how the rest of the chapter is organized: We start in Section 2 by describing the *standard statistical framework*, allowing us to estimate the statistical precision of sample-mean estimators, both before and after the simulation experiment is conducted. In Section 2 we define the asymptotic variance and the asymptotic bias of a sample mean. We relate these asymptotic quantities to the ordinary variance and bias of a sample mean. We show the critical role played by the asymptotic variance in confidence intervals and thus for the required sample size to obtain desired statistical precision. We first discuss the classical statistical case of independent and identically distributed (i.i.d.) random variables, which arises naturally when the simulation estimate is based on *independent replications*. For i.i.d. random variables, the asymptotic variance coincides with the variance of a single random variable. Finally, we discuss the problem of initial transients and correlations that arise when we form the sample mean from a stochastic process observed over time within a single run.

In Section 3, following Whitt (1992), we indicate how to compute the asymptotic variance and the asymptotic bias of functions of continuous-time Markov chains. We describe a recursive algorithm for functions of birth-and-death processes. In Section 4 we consider several birth-and-death process examples, including the $M/M/1$ and $M/M/\infty$ queueing models. These examples show that model structure can make a big difference in the computational effort required for estimation by simulation.

In Section 5 we consider diffusion processes, which are continuous analogues of birth-and-death processes. We give integral representations of the asymptotic parameters for diffusion processes, which enable computation by numerical integration. In Section 6 we discuss applications of stochastic-process limits to the planning process. Following Whitt (1989) and Srikant and Whitt (1996), we show how heavy-traffic limits yield relatively simple diffusion approximations for the asymptotic variance and the asymptotic bias of sample-mean estimators for single-server and many-server queues. The time scaling in the heavy-traffic limits plays a critical role. In Section 7 we consider not collecting data for an initial portion of a simulation run to reduce the bias. Finally, in Section 8 we discuss directions for further research.

2 The standard statistical framework

2.1 Probability model of a simulation

We base our discussion on a probability model of a (stochastic) simulation experiment: In the model, the simulation experiment generates an initial segment of a stochastic process, which may be a discrete-time stochastic process $\{X_n\colon\ n \geqslant 1\}$ or a continuous-time stochastic process $\{X(t)\colon\ t \geqslant 0\}$. We form the relevant sample mean

$$\overline{X}_n \equiv n^{-1} \sum_{i=1}^{n} X_i \quad \text{or} \quad \overline{X}_t \equiv t^{-1} \int_0^t X(s)\,\mathrm{d}s,$$

and use the sample mean to estimate the long-run average,

$$\mu = \lim_{n\to\infty} \overline{X}_n \quad \text{or} \quad \mu = \lim_{t\to\infty} \overline{X}_t,$$

which is assumed to exist as a proper limit with probability one (w.p.1). Under very general regularity conditions, the long-run average coincides with the expected value of the limiting steady-state distribution of the stochastic process. For example, supporting theoretical results are available for regenerative processes, Chapter VI of Asmussen (2003); stationary marked point processes, Section 2.5 of Sigman (1995); and generalized semi-Markov processes (GSMPs), Glynn (1989).

These stochastic processes arise in both observations from a single run and from independent replications. For example, in observations from a single run, a discrete-time stochastic process $\{X_n: n \geqslant 1\}$ arises if we consider the waiting times of successive arrivals to a queue. The random variable X_n might be the waiting time of the nth arrival before beginning service; then μ is the long-run average waiting time of all arrivals, which usually coincides with the mean steady-state waiting time. On the other hand, X_n might take the value 1 if the nth arrival waits less than or equal to x minutes, and take the value 0 otherwise; then $\mu \equiv \mu(x)$ is the long-run proportion of customers that wait less than or equal to x minutes, which usually corresponds to the probability that the steady-state waiting time is less than or equal to x minutes.

Alternatively, in observations from a single run, a continuous-time stochastic process $\{X(t): t \geqslant 0\}$ arises if we consider the queue length over time, beginning at time 0. The random variable $X(t)$ might be the queue length at time t or $X(t)$ might take the value 1 if the queue length at time t is less than or equal to k, and take the value 0 otherwise.

With independent replications (separate independent runs of the experiment), we obtain a discrete-time stochastic process $\{X_n: n \geqslant 1\}$. Then X_n represents a random observation obtained from the nth run. For example, X_n might be the queue length at time 7 in the nth replication or X_n might be the average queue length over the time interval [0, 7] in the nth replication. Then the limit μ represents the long-run average over many independent replications, which equals the expected value of the random variable in any single run. Such expected values describe the expected *transient* (or time-dependent) behavior of the system.

2.2 Bias, mean-squared error and variance

By assuming that the limits exist, we are assuming that we would obtain the exact answer if we devoted unlimited computational effort to the simulation experiment. In statistical language, e.g., see Lehmann and Casella (1998), we are assuming that the estimators $\overline{X}_n$ and $\overline{X}_t$ are *consistent estimators* of the quantity to be estimated, μ. For finite sample size, we can describe the statistical precision by looking at the bias and the mean-squared error. The *bias*, which we denote by $\bar{\beta}_n$ in the discrete-time case and $\bar{\beta}_t$ in the continuous-time case, indicates how much the expected value of the estimator differs from the quantity being estimated, and in what direction. For example, in the discrete-time case, the bias of $\overline{X}_n$ is

$$\bar{\beta}_n = \mathrm{E}[\overline{X}_n] - \mu.$$

The mean-squared error (MSE_n or MSE_t) is the expected squared error, e.g.,

$$\mathrm{MSE}_n = \mathrm{E}\bigl[|\overline{X}_n - \mu|^2\bigr].$$

If there is no bias, then the MSE coincides with the variance of $\overline{X}_n$, which we denote by $\bar{\sigma}_n^2$, i.e.,

$$\bar{\sigma}_n^2 \equiv \mathrm{Var}(\overline{X}_n) \equiv \mathrm{E}\bigl[\bigl|\overline{X}_n - \mathrm{E}[\overline{X}_n]\bigr|^2\bigr].$$

Then we can write

$$\bar{\sigma}_n^2 \equiv \mathrm{Var}(\overline{X}_n) = n^{-2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{Cov}(X_i, X_j),$$

where $\mathrm{Cov}(X_i, X_j)$ is the *covariance*, i.e.,

$$\mathrm{Cov}(X_i, X_j) \equiv \mathrm{E}[X_i X_j] - \mathrm{E}[X_i]\,\mathrm{E}[X_j].$$

Analogous formulas hold in continuous time. For example, then the variance of the sample mean $\overline{X}_t$ is

$$\bar{\sigma}_t^2 \equiv \mathrm{Var}(\overline{X}_t) = t^{-2}\int_0^t\int_0^t \mathrm{Cov}\bigl(X(u), X(v)\bigr)\,\mathrm{d}u\,\mathrm{d}v.$$

Unfortunately, these general formulas usually are too complicated to be of much help when doing preliminary planning. What we will be doing in the rest of this paper is showing how to develop simple approximations for these quantities.

2.3 The classical case: independent replications

In statistics, the classical case arises when we have a discrete-time stochastic process $\{X_n\colon n \geqslant 1\}$, where the random variables X_n are mutually independent and identically distributed (i.i.d.) with mean μ and finite variance σ^2, and we use the sample mean $\overline{X}_n$ to estimate the mean μ. Clearly, the classical case arises whenever we use independent replications to do estimation, which of course is the great appeal of independent replications.

In the classical case, the sample mean $\overline{X}_n$ is a consistent estimator of the mean μ by the *law of large numbers* (LLN). Then there is no bias and the MSE coincides with the variance of the sample mean, $\bar{\sigma}_n^2$, which is a simple function of the variance of a single observation X_n,

$$\bar{\sigma}_n^2 = \text{MSE}(\overline{X}_n) = \frac{\sigma^2}{n}.$$

Moreover, by the *central limit theorem* (CLT), $\overline{X}_n$ is asymptotically normally distributed as the sample size n increases, i.e.,

$$n^{1/2}[\overline{X}_n - \mu] \Rightarrow \text{N}(0, \sigma^2) \quad \text{as } n \to \infty,$$

where $\text{N}(a, b)$ is a normal random variable with mean a and variance b, and "$\Rightarrow$" denotes convergence in distribution.

We thus use this *large-sample theory* to justify the approximation

$$\text{P}(\overline{X}_n \leqslant x) \approx \text{P}\left(\text{N}\left(\mu, \frac{\sigma^2}{n}\right) \leqslant x\right) = \text{P}\left(\text{N}(0, 1) \leqslant \frac{x - \mu}{\sqrt{\sigma^2/n}}\right).$$

Based on this normal approximation, a $(1 - \alpha)100\%$ *confidence interval* for μ based on the sample mean $\overline{X}_n$ is

$$\left[\overline{X}_n - z_{\alpha/2}\frac{\sigma}{\sqrt{n}}, \overline{X}_n + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right],$$

where

$$\text{P}(-z_{\alpha/2} \leqslant \text{N}(0, 1) \leqslant +z_{\alpha/2}) = 1 - \alpha.$$

A common choice is a 95% confidence interval, which corresponds to $\alpha = 0.05$; then $z_{\alpha/2} = 1.96 \approx 2$.

The *statistical precision* is typically described by either the *absolute width* or the *relative width* of the confidence interval, denoted by $w_a(\alpha)$ and $w_r(\alpha)$, respectively, which are

$$w_a(\alpha) = \frac{2z_{\alpha/2}\sigma}{\sqrt{n}} \quad \text{and} \quad w_r(\alpha) = \frac{2z_{\alpha/2}\sigma}{\mu\sqrt{n}}.$$

There are circumstances where each measure is preferred. For specified *absolute width* or *relative width* of the confidence interval, ε, and for specified level of precision α, the *required sample size* $n_a(\varepsilon, \alpha)$ or $n_r(\varepsilon, \alpha)$ is then

$$n_a(\varepsilon, \alpha) = \frac{4\sigma^2 z_{\alpha/2}^2}{\varepsilon^2} \quad \text{or} \quad n_r(\varepsilon, \alpha) = \frac{4\sigma^2 z_{\alpha/2}^2}{\mu^2\varepsilon^2}. \tag{1}$$

From these formulas, we see that $n_a(\varepsilon, \alpha)$ and $n_r(\varepsilon, \alpha)$ are both inversely proportional to ε^2 and directly proportional to σ^2 and $z_{\alpha/2}^2$.

Standard statistical theory describes how observations (data) can be used to estimate the unknown quantities μ and σ^2. We might use a two-stage sampling procedure, exploiting the first stage to estimate the required sample size. However, here we are concerned with what we can do without any data at all.

We propose applying additional information about the model to obtain rough preliminary estimates for these parameters without data. Following the general approach of this paper, we suggest trying to estimate μ and σ^2 before conducting the simulation by analyzing the probability distribution of the outcome of a single replication, X_n (using knowledge about the model). Admittedly, this preliminary estimation is often difficult to do; our approach is usually more useful in the context of one long run, which is discussed in the next section.

However, more can be done in this context than is often thought. Again, we must remember that we are only interested in making a rough estimate. Thus, we should be ready to make back-of-the-envelope calculations.

To illustrate what can be done, suppose that we focus on the relative-width criterion. With the relative-width criterion, it suffices to estimate the *squared coefficient of variation* (SCV, variance divided by the square of the mean)

$$c^2 \equiv \frac{\sigma^2}{\mu^2},$$

instead of both μ and σ^2. With the relative-width criterion, the required sample size is

$$n_r(\varepsilon, \alpha) = \frac{4c^2 z_{\alpha/2}^2}{\varepsilon^2}.$$

From the analysis above, we see that we only need to estimate a single parameter, the SCV c^2, in order to carry out this preliminary analysis. In many cases, we can make reasonable estimates based on "engineering judgment". For that step, it helps to have experience with variability as quantified by the SCV. First, note that the SCV measures the level of variability *independent of the mean*: The SCV of a random variable is unchanged if we multiply the random variable by a constant. We are thus focusing on the variability independent of the mean. Clearly, it is important to realize that the mean itself plays no role with the relative-width criterion.

Once we learn to focus on the SCV, we quickly gain experience about what to expect. A common reference case for the SCV is an exponential distribution, which has $c^2 = 1$. A unit point mass (deterministic distribution) has $c^2 = 0$. Distributions relatively more (less) variable than exponential have $c^2 >(<) 1$. In many instances we actually have a rough idea about the SCV. We might be able to judge in advance that the SCV is one of: (i) less than 1, but probably not less than 0.1, (ii) near 1, (iii) bigger than 1, but probably not bigger than 10, or (iv) likely to be large, even bigger than 10. In other words, it is not unreasonable to be able to estimate the SCV to within an order of magnitude (within a factor of 10). And that may be good enough for the desired rough estimates we want to make.

In lieu of information about the SCV, to obtain a rough estimate we can just let $c^2 = 1$. To proceed, we can also let $\alpha = 0.05$, so that $z_{\alpha/2} \approx 2$. Then, if we set $\varepsilon = 10^{-k}$, the required simulation run length is

$$n_r\big(10^{-k}, 0.05\big) = 16 \times (10)^{2k}.$$

Thus, when $c^2 = 1$, 10% relative precision requires about 1600 replications, while 1% relative precision requires 160,000 replications. If $c^2 \neq 1$, then we would multiply the number of required replications above by c^2. We thus can easily factor in the only unknown, the SCV c^2.

We have just reviewed the classical i.i.d. case, which is directly relevant when we use independent replications. However, in this chapter we concentrate on the more complicated case in which we consider an initial segment of a stochastic process from a single simulation run. It is good to have the classical i.i.d. case in mind, though, to understand the impact of bias and dependence upon the required computational effort.

2.4 *An initial segment from a single run*

Now suppose that we intend to estimate a long-run average within a single run by the sample mean from an initial segment of a stochastic process, which could evolve in either discrete time or continuous time. The situation is now much more complicated, because the random observations need not be i.i.d. Indeed, they need not be either independent or identically distributed. Unlike the case of independent replications, we now face the problems of bias and dependence among the random variables.

Fortunately, there are generalizations of the classical i.i.d. framework that enable us to estimate the bias and the mean squared error as a function of the sample size in terms of only two fundamental parameters: the asymptotic bias and the asymptotic variance; see Whitt (1992) and references therein. That theory tells us that, under regularity conditions, both the bias and the MSE are of order $1/n$.

Within a single run, the stochastic processes tend to become stationary as time evolves. Indeed, now we assume that $X_n \Rightarrow X(\infty)$ as $n \to \infty$ (in the discrete-time case) and $X(t) \Rightarrow X(\infty)$ as $t \to \infty$ (in the continuous-time case). The stochastic processes fail to be stationary throughout all time primarily because it is necessary (or at least more convenient) to start the simulation in a special initial state. We thus can reduce the bias by choosing a good initial state or by deleting (not collecting statistics over) an initial portion of the simulation run. Choosing an appropriate initial state can be difficult if the stochastic process of interest is not nearly Markov. For example, even for the relatively simple $M/G/s/\infty$ queueing model, with s servers and nonexponential service times, it is necessary to specify the remaining service time of all customers initially in service.

The asymptotic bias helps us to determine if it is necessary to choose a special initial state or delete an initial portion of the run. The asymptotic bias also helps us estimate the final bias, whether or not we choose a special initial state or delete an initial portion of the run. It also helps us determine what proportion of the full run should be deleted if we follow that procedure.

Under regularity conditions, there is a parameter $\bar{\beta}$ called the *asymptotic bias* such that

$$\bar{\beta} = \lim_{n\to\infty} n\bar{\beta}_n; \tag{2}$$

see Whitt (1992) and references therein. Given the definition of the bias $\bar{\beta}_n$, we see that the asymptotic bias must be

$$\bar{\beta} = \sum_{i=1}^{\infty}\bigl(\mathrm{E}[X_i] - \mu\bigr);$$

the regularity conditions ensure that the sum is absolutely convergent. We thus approximate the bias of $\overline{X}_n$ for any sufficiently large n by

$$\bar{\beta}_n \approx \frac{\bar{\beta}}{n}.$$

This approximation reduces the unknowns to be estimated from the function $\{\bar{\beta}_n\colon\, n \geqslant 1\}$ to the single parameter $\bar{\beta}$. Corresponding formulas hold in continuous time.

Given that we can ignore the bias, either because it is negligible or because it has been largely removed by choosing a good initial state or by deleting an initial portion of the run, we can use the asymptotic variance to estimate the width of confidence intervals and thus the required run length to yield desired statistical precision. Under regularity conditions, there is a parameter $\bar{\sigma}^2$ called the *asymptotic variance* such that

$$\bar{\sigma}^2 = \lim_{n\to\infty} n\bar{\sigma}_n^2, \tag{3}$$

where (under the assumption that $\{X_n\colon\, n \geqslant 1\}$ is a stationary process)

$$\bar{\sigma}^2 = \mathrm{Var}(X_1) + 2\sum_{i=1}^{\infty}\mathrm{Cov}(X_1, X_{1+i}),$$

with σ^2 being the variance of X_n and $\mathrm{Cov}(X_1, X_{1+i})$ being the lag-i *autocovariance*. Because of the dependence, $\bar{\sigma}^2$ often is much bigger than σ^2. We thus approximate $\bar{\sigma}_n^2$, the variance of $\overline{X}_n$, for any sufficiently large n by

$$\bar{\sigma}_n^2 \equiv \mathrm{Var}(\overline{X}_n) \approx \frac{\bar{\sigma}^2}{n}.$$

Again, this approximation reduces the unknowns to be estimated from the function $\{\bar{\sigma}_n^2\colon\, n \geqslant 1\}$ to the single parameter $\bar{\sigma}^2$.

In continuous time, we have the related asymptotic variance formula

$$\bar{\sigma}^2 = \lim_{t\to\infty} t\bar{\sigma}_t^2,$$

where (under the assumption that $\{X(t)\colon\ t \geqslant 0\}$ is a stationary process)

$$\bar{\sigma}^2 = 2\int_0^\infty \mathrm{Cov}\bigl(X(0), X(t)\bigr)\,\mathrm{d}t.$$

In continuous or discrete time, a critical assumption here is that the asymptotic variance $\bar{\sigma}^2$ is actually finite. The asymptotic variance could be infinite for two reasons: (i) heavy-tailed distributions and (ii) long-range dependence. In our context, we say that X_n or $X(t)$ has a heavy-tailed distribution if its variance is infinite. In our context, we say that there is long-range dependence (without heavy-tailed distributions) if the variance $\mathrm{Var}(X_n)$ or $\mathrm{Var}(X(t))$ is finite, but nevertheless the asymptotic variance is infinite because the autocovariances $\mathrm{Cov}(X_j, X_{j+k})$ or $\mathrm{Cov}(X(t), X(t+k))$ do not decay quickly enough as $k \to \infty$; e.g., see Beran (1994), Samorodnitsky and Taqqu (1994) and Chapter 4 of Whitt (2002).

Assuming that $\bar{\sigma}^2 < \infty$, we can apply CLTs for weakly dependent random variables (involving other regularity conditions, e.g., see Section 4.4 of Whitt, 2002) to deduce that $\overline{X}_n$ (as well as $\overline{X}_t$) is again asymptotically normally distributed as the sample size n increases, i.e.,

$$n^{1/2}[\overline{X}_n - \mu] \Rightarrow \mathrm{N}\bigl(0, \bar{\sigma}^2\bigr) \quad \text{as } n \to \infty,$$

so that the asymptotic variance $\bar{\sigma}^2$ plays the role of the ordinary variance σ^2 in the classical i.i.d. setting.

We thus again can use the large-sample theory to justify a normal approximation. The new $(1-\alpha)100\%$ confidence interval for μ based on the sample mean $\overline{X}_n$ is

$$\left[\overline{X}_n - z_{\alpha/2}\frac{\bar{\sigma}}{\sqrt{n}}, \overline{X}_n + z_{\alpha/2}\frac{\bar{\sigma}}{\sqrt{n}}\right],$$

which is the same as for independent replications except that the asymptotic variance $\bar{\sigma}^2$ replaces the ordinary variance σ^2.

The formulas for the confidence interval relative width, $w_r(\alpha)$, and the required run length, $n_r(\varepsilon, \alpha)$, are thus also the same as for independent replications in (1) except that the asymptotic variance $\bar{\sigma}^2$ is substituted for the ordinary variance σ^2; e.g., the required simulation run length with a relative-width criterion is

$$n_r(\varepsilon, \alpha) = \frac{4\bar{\sigma}^2 z_{\alpha/2}^2}{\mu^2\varepsilon^2} \quad \text{and} \quad n_r\bigl(10^{-k}, 0.05\bigr) \approx \frac{\bar{\sigma}^2}{\mu^2} 16 \times (10)^{2k}. \tag{4}$$

From (1) and (4), we immediately see that the required run length is approximately $\bar{\sigma}^2/\sigma^2$ times greater when sampling from one run than with independent sampling (assuming that we could directly observe independent samples from the steady-state distribution, which of course is typically not possible).

As with independent replications, established simulation methodology and statistical theory tells how to estimate the unknown quantities μ, $\bar{\beta}$ and $\bar{\sigma}^2$ from data; e.g., see Bratley et al. (1987) and Fishman (2001). Instead, we apply additional information about the model to obtain rough preliminary estimates for these parameters without data. For $\bar{\sigma}^2$, the representation of the asymptotic variance in terms of the autocovariances is usually too complicated to be of much help, but fortunately there is another approach, which we will describe in Section 3.

3 The asymptotic parameters for a function of a Markov chain

From the previous section, it should be apparent that we can do the intended preliminary planning if we can estimate the asymptotic bias and the asymptotic variance. We now start to describe how we can calculate these important parameters. We first consider functions of a Markov chain. That illustrates available general results. However, fast back-of-the-envelope calculations usually depend on diffusion approximations, based on stochastic-process limits, after doing appropriate scaling, which we discuss later in Sections 5 and 6. Indeed, the scaling is usually the key part, and that is so simple that back-of-the-envelope calculations are actually possible.

In this section, drawing on Whitt (1992), which itself is primarily a survey of known results (including Glynn (1984) and Grassman (1987a, 1987b) among others), we observe that (again under regularity conditions) we can calculate the asymptotic bias and the asymptotic variance whenever the stochastic process of interest is a function of a (positive-recurrent irreducible) Markov chain, i.e., when $X_n = f(Y_n)$ for $n \geqslant 1$, where f is a real-valued function and $\{Y_n\colon n \geqslant 1\}$ is a Markov chain or when $X(t) = f(Y(t))$ for $t \geqslant 0$, where again f is a real-valued function and $\{Y(t)\colon t \geqslant 0\}$ is a Markov chain. As noted before, we usually obtain the required Markov structure by approximating the given stochastic process by a related one with the Markov property.

In fact, as in Whitt (1992), we only discuss the case in which the underlying Markov chain has a finite state space (by which we mean countably finite, i.e., $\{0, 1, \ldots, m\}$, not $[c, d]$), but the theory extends to more general state spaces under regularity conditions. For illustrations, see Glynn (1994) and Glynn and Meyn (1996). But the finite-state-space condition is very useful: Under the finite-state-space condition, we can compute the asymptotic parameters numerically, without relying on special model structure. However, when we do have special structure, we can sometimes go further to obtain relatively simple closed-form formulas. We also obtain relatively simple closed-form formulas when we establish diffusion-process approximations via stochastic-process limits.

3.1 Continuous-time Markov chains

We will discuss the case of a continuous-time Markov chain (CTMC); similar results hold for discrete-time Markov chains. Suppose that the CTMC $\{Y(t)\colon t \geqslant 0\}$ is irreducible with finite state space $\{0, 1, \dots, m\}$ (which implies that it is positive recurrent). Our sample-mean estimator is

$$\overline{X}_t \equiv t^{-1} \int_0^t X(s)\,\mathrm{d}s, \quad t \geqslant 0,$$

where $X(t) = f(Y(t))$. (With the discrete state space, we write both $f(j)$ and f_j for the value of f at argument j.)

A finite-state CTMC is characterized by its *infinitesimal generator matrix* $Q \equiv (Q_{i,j})$; $Q_{i,j}$ is understood to be the derivative (from above) of the *probability transition function*

$$P_{i,j}(t) \equiv \mathrm{P}\bigl(Y(s+t) = j | Y(s) = i\bigr)$$

with respect to time t evaluated at $t = 0$. However, in applications the model is specified by defining the infinitesimal generator Q. Then the probability transition function can be obtained subsequently as the solution of the ordinary differential equations

$$P'(t) = P(t)Q = QP(t),$$

which takes the form of the matrix exponential

$$P(t) = \mathrm{e}^{Qt} \equiv \sum_{k=0}^{\infty} \frac{Q^k t^k}{k!}.$$

Key asymptotic quantities associated with a CTMC are the stationary probability vector π and the fundamental matrix Z. (By convention, we let vectors be row vectors.) The *stationary probability vector* π can be found by solving the system of equations

$$\pi Q = 0 \quad \text{with} \sum_{i=0}^{m} \pi_i = 1.$$

The quantity we want to estimate is the expected value of the function f (represented as a row vector) with respect to the stationary probability (row) vector π, i.e.,

$$\mu = \pi f^{\mathrm{T}} = \sum_{i=0}^{m} \pi_i f_i,$$

where "T" is the *transpose*.

We would not need to conduct a simulation to estimate μ if indeed we can calculate it directly as indicated above. As noted before, in intended applications of this planning approach, the actual model of interest tends to be more complicated than a CTMC, so that we cannot calculate the desired quantity μ directly. We introduce a CTMC that is similar to the more complicated model of interest, and use the CTMC analysis to get rough estimates of both μ and the required computational effort in order to estimate μ by simulation. We will illustrate for queueing models later.

To continue, the fundamental matrix Z describes the time-dependent deviations from the stationary vector, in particular,

$$Z_{i,j} \equiv \int_0^\infty \left[P_{i,j}(t) - \pi_j\right] \mathrm{d}t.$$

Given the stationary probability vector, π, the fundamental matrix Z can be found by first forming the square matrix Π, all of whose rows are the vector π, and then calculating

$$Z = (\Pi - Q)^{-1} - \Pi,$$

with the inverse always existing; again see Whitt (1992) and references therein. We now consider how the desired asymptotic parameters can be expressed in terms of the stationary probability vector π and the fundamental matrix Z, including ways that are especially effective for computation.

3.2 Poisson's equation

For that purpose, it is useful to introduce Poisson's equation. The stationary probability vector π and the fundamental matrix Z can be viewed as solutions x to *Poisson's equation*

$$xQ = y,$$

for appropriate (row) vectors y. It can be shown that Poisson's equation has a solution x if and only if $ye^{\mathrm{T}} = 0$, where e is the row vector of 1's and e^{T} is its transpose. Then all solutions are of the form

$$x = -yZ + \left(xe^{\mathrm{T}}\right)\pi.$$

For example, π is obtained by solving Poisson's equation when y is the zero vector (and normalizing by requiring that $xe^{\mathrm{T}} = 1$). Then elements of Z can be obtained by choosing other vectors y, requiring that $xe^{\mathrm{T}} = 0$.

In passing, we remark that there also is an alternate *column-vector form of Poisson's equation*, namely,

$$Qx^{\mathrm{T}} = y^{\mathrm{T}},$$

which has a solution if and only if $\pi y^{\mathrm{T}} = 0$. Then all solutions are of the form

$$x^{\mathrm{T}} = -Zy^{\mathrm{T}} + (\pi x^{\mathrm{T}})e^{\mathrm{T}}.$$

It is significant that, for a CTMC, the asymptotic bias $\bar{\beta}$ defined in (2) and the asymptotic variance $\bar{\sigma}^2$ defined in (3) can be expressed directly in terms of π, Z, the function f and (for β) the initial probability vector, say ξ, i.e.,

$$\bar{\beta}(\xi) = \xi Z f^{\mathrm{T}} \equiv \sum_{i=0}^{m}\sum_{j=0}^{m} \xi_i Z_{i,j} f_j$$

and

$$\bar{\sigma}^2 = 2(f\pi^{\mathrm{T}})Zf^{\mathrm{T}} \equiv 2\sum_{i=0}^{m}\sum_{j=0}^{m} f_i \pi_i Z_{i,j} f_j.$$

Moreover, the asymptotic parameters $\bar{\beta}(\xi)$ and $\bar{\sigma}^2$ are themselves directly solutions to Poisson's equation. In particular,

$$\bar{\beta}(\xi) = xf^{\mathrm{T}},$$

where x is the unique solution to Poisson's equation for $y = -\xi + \pi$ with $xe^{\mathrm{T}} = 0$. Similarly,

$$\bar{\sigma}^2 = 2xf^{\mathrm{T}},$$

where x is the unique solution to Poisson's equation for $y_i = -(f_i - \mu)\pi_i$ with $xe^{\mathrm{T}} = 0$.

3.3 Birth-and-death processes

Birth-and-death (BD) processes are special cases of CTMCs in which $Q_{i,j} = 0$ when $|i - j| > 1$; then we often use the notation $Q_{i,i+1} \equiv \lambda_i$ and $Q_{i,i-1} \equiv \mu_i$, and refer to λ_i as the birth rates and μ_i as the death rates. For BD processes and skip-free CTMCs (which in one direction can go at most one step), Poisson's equation can be efficiently solved *recursively*.

To describe the recursive algorithm for BD processes, we start by observing that for a BD process Poisson's equation $xQ = y$ is equivalent to the system of equations

$$x_{j-1}\lambda_{j-1} - x_j(\lambda_j + \mu_j) + x_{j+1}\mu_{j+1} = y_j, \quad j \geqslant 0,$$

where $x_{-1} = x_{m+1} = 0$. Upon adding the first $j + 1$ equations, we obtain the desired recursive algorithm,

$$x_{j+1} = \frac{\lambda_j x_j + s_j}{\mu_{j+1}},$$

where

$$s_j = \sum_{i=0}^{j} y_i.$$

Hence, Poisson's equation for BD processes can indeed be solved recursively.

For BD processes and their continuous-time relatives – diffusion processes – the asymptotic parameters can be expressed directly as sums and integrals, respectively. For BD processes,

$$\bar{\beta}(\xi) = \sum_{j=0}^{m-1} \frac{1}{\lambda_j \pi_j} \sum_{i=0}^{j} (f_i - \mu)\pi_i \sum_{k=0}^{j} (\xi_k - \pi_k)$$

and

$$\bar{\sigma}^2 = 2 \sum_{j=0}^{m-1} \frac{1}{\lambda_j \pi_j} \left[\sum_{i=0}^{j} (f_i - \mu)\pi_i \right]^2,$$

where, as for CTMCs, π is the steady-state probability vector, while μ is the expected value of f with respect to π. However, for BD processes, it is usually easier to use the recursive algorithm for computation. Indeed, the recursive algorithm for the asymptotic bias and the asymptotic variance parallels the well known recursive algorithm for the steady-state probability vector π.

4 Birth-and-death examples

In this section we consider examples of BD processes, primarily of queueing models. These examples show that the model structure can make a big difference in the computational effort required for estimation by simulation.

Example 1 (The $M/M/1$ queue). Consider the queue-length (number in system, including the customer in service, if any) process $\{Q(t)\colon t \geqslant 0\}$ in the $M/M/1$ queue with unlimited waiting space. This model has a Poisson arrival process with constant rate and i.i.d. service times with an exponential distribution. The state space here is infinite, but the theory for the asymptotic parameters extends to this example. The queue-length process is a BD process with constant arrival (birth) rate and constant service (death) rate.

Let the service rate be 1 and let the arrival rate and traffic intensity be ρ. Fixing the service rate gives meaning to time in the simulation run length. Let $f(i) = i$ for all i, so that we are estimating the steady-state mean. The steady-state mean and variance are

$$\mu = \frac{\rho}{1-\rho} \quad \text{and} \quad \sigma^2 = \frac{\rho}{(1-\rho)^2};$$

e.g., see Cohen (1982).

To estimate the required simulation run length from a single long run, we use the asymptotic bias and the asymptotic variance. Let the system start out empty, so that the initial state is 0. As an argument of $\bar{\beta}(\xi)$, let 0 also denote the initial probability vector ξ that puts mass 1 on the state 0. Then

$$\bar{\beta}(0) = \frac{-\rho}{(1-\rho)^3} \quad \text{and} \quad \bar{\sigma}^2 = \frac{2\rho(1+\rho)}{(1-\rho)^4}.$$

These formulas can be derived from the general BD formulas or directly; see Abate and Whitt (1987a, 1988a, 1988b).

Ignoring the initial transient (assuming that the queue-length process we observe is a stationary process), the required run length with a relative-width criterion, specified in general in (4), here is

$$t_r(\varepsilon, \alpha) = \frac{8(1+\rho)z_{\alpha/2}^2}{\rho(1-\rho)^2\varepsilon^2} \quad \text{and} \quad t_r\big(10^{-k}, 0.05\big) \approx \frac{32(1+\rho)(10)^{2k}}{\rho(1-\rho)^2}.$$

For 10% statistical precision ($\varepsilon = 0.1$) with 95% confidence intervals ($\alpha = 0.05$), when the traffic intensity is $\rho = 0.9$, the required run length is about 675,000 (mean service times, which corresponds to an expected number of arrivals equal to $0.9 \times 675{,}000 = 608{,}000$); when the traffic intensity is $\rho = 0.99$, the required run length is 64,300,000 (mean service times, which corresponds to an expected number of arrivals equal to $0.9 \times 64{,}300{,}000 = 57{,}900{,}000$). To summarize, for high traffic intensities, the required run length is of order 10^6 or more mean service times. We can anticipate great computational difficulty as the traffic intensity ρ increases toward the critical value for stability.

Compared to independent sampling of the steady-state queue length (which is typically not directly an option), the required run length is greater by a factor of

$$\frac{\bar{\sigma}^2}{\sigma^2} = \frac{2(1+\rho)}{\rho(1-\rho)^2},$$

which equals 422 when $\rho = 0.9$ and 40,200 when $\rho = 0.99$. Clearly, the dependence can make a great difference.

Now let us consider the bias. The relative bias is

$$\frac{\bar{\beta}_t(0)}{\mu} \approx \frac{\bar{\beta}(0)}{t\mu} = \frac{-1}{(1-\rho)^2 t},$$

so that, for $\rho = 0.9$ the relative bias starting empty is about $100/t$, where t is the run length. For a run length of 675,000, the relative bias is 1.5×10^{-4} or 0.015%, which is indeed negligible compared to the specified 10% relative width of the confidence interval. Hence the bias is in the noise; it can be ignored for high traffic intensities. The situation is even more extreme for higher traffic intensities such as $\rho = 0.99$.

Example 2 (A small example with large asymptotic parameters). It is interesting to see that the asymptotic bias $\bar{\beta}(\xi)$ and the asymptotic variance $\bar{\sigma}^2$ can be arbitrarily large in a very small BD model with bounded rates. Suppose that $m = 2$, so that the BD process has only 3 states: 0, 1 and 2. Consider the symmetric model in which $\lambda_0 = \mu_2 = x$ and $\lambda_1 = \mu_1 = 1$, where $0 < x \leqslant 1$. Then the stationary distribution is

$$\pi_0 = \pi_2 = \frac{1}{2+x} \quad \text{and} \quad \pi_1 = \frac{x}{2+x}.$$

Let $f_i = i$ for all i, so that we are estimating the mean μ. Then the mean is $\mu = 1$ and the asymptotic variance is

$$\bar{\sigma}^2 = \frac{4}{x(2+x)} \approx \frac{2}{x} \quad \text{for small } x.$$

This model has a high asymptotic variance $\bar{\sigma}^2$ for small x because the model is *bistable*, tending to remain in the states 0 and 2 a long time before leaving. To see this, note that the mean first passage time from state 0 or state 2 to state 1 is $1/x$.

Note that the large asymptotic variance $\bar{\sigma}^2$ cannot be detected from the variance of the steady-state distribution, σ^2. As $x \downarrow 0$, σ^2, the variance of π, increases to 1. Thus, the ratio $\bar{\sigma}^2/\sigma^2$ is of order $\mathrm{O}(1/x)$. The steady-state distribution has moderate variance, but the process has quite strong dependence (but not so strong that the asymptotic variance becomes infinite).

The asymptotic bias starting in state 0 (or state 2) is also large for small x. The asymptotic bias starting in state 0 is

$$\bar{\beta}(0) = \frac{-(x+1)^2}{x(x+2)^2} \approx \frac{-1}{4x} \quad \text{for small } x.$$

As a function of the key model parameter x, the bias is much more important here than it was for the previous $M/M/1$ queue example. Here, *both* the asymptotic bias and the asymptotic variance are of order $\mathrm{O}(1/x)$, so that as a function of x, for very small x, the width of the confidence interval is $\mathrm{O}(1/\sqrt{x}\,)$, while the bias is of order $\mathrm{O}(1/x)$. Thus the bias tends to be much more important in this example. In particular, the run length required to make the bias suitably small is of the *same order* as the run length required to make the width of a confidence interval suitably small. For this example, using simulation to estimate the mean μ when the parameter x is very small would be difficult at best.

This model is clearly pathological: For very small x, a relatively long simulation run of this model starting in state 0 could yield a sample path that is identically zero. We might never experience even a single transition! This example demonstrates that it can be very helpful to know something about model structure when conducting a simulation.

Example 3 (The $M/M/\infty$ queue). A queueing system with many servers tends to behave quite differently from a single-server queue. A queueing system with many servers can often be well approximated by an infinite-server queue. Thus we consider the number of busy servers at time t, also denoted by $Q(t)$, in an $M/M/\infty$ queue. As before, let the mean individual service time be 1, but now let the arrival rate be λ (since the previous notion of traffic intensity is no longer appropriate). Now the arrival rate λ can be arbitrarily large.

The first thing to notice is that as λ increases, the required computational effort for given simulation run length in the simulation increases, simply because the expected number of arrivals in the interval $[0, t]$ is λt. Thus, with many servers, we need to do a further adjustment to properly account for computational effort. To describe the computational effort, it is appropriate to multiply the time by λ. Thus, for the $M/M/\infty$ model with mean individual service rate 1, we let $c_r = \lambda t_r$ represent the required computational effort associated with the required run length t_r.

It is well known that the steady-state number of busy servers in the $M/M/\infty$ model, say $Q(\infty)$, has a Poisson distribution with mean λ; e.g., see Cooper (1982). Thus, the mean and variance of the steady-state distribution are

$$\mu \equiv \mathrm{E}\bigl[Q(\infty)\bigr] = \lambda \quad \text{and} \quad \sigma^2 \equiv \mathrm{Var}\bigl[Q(\infty)\bigr] = \lambda.$$

The asymptotic parameters also are relatively simple. As for the $M/M/1$ queue, we assume that we start with an empty system. Then the asymptotic bias and asymptotic variance are

$$\bar{\beta}(0) = -\lambda \quad \text{and} \quad \bar{\sigma}^2 = 2\lambda.$$

From the perspective of the asymptotic variance and relative error, we see that

$$\frac{\bar{\sigma}^2}{\mu^2} = \frac{2\lambda}{\lambda^2} = \frac{2}{\lambda},$$

so that simulation efficiency increases as λ increases. However, the required computational effort to achieve relative $(1 - \alpha)\%$ confidence interval width of ε is

$$c_r(\varepsilon, \alpha) \equiv \lambda t_r(\varepsilon, \alpha) = \frac{8z_\alpha^2}{\varepsilon^2}$$

which is *independent of* λ. Thus, from the perspective of the asymptotic variance, the required computational effort does not increase with the arrival rate, which is very different from the single-server queue.

Unfortunately, the situation is not so good for the relative bias. First, the key ratio is

$$\frac{\bar{\beta}(0)}{\mu} = \frac{-\lambda}{\lambda} = -1.$$

Thus the required run length to make the bias less than ε is $1/\varepsilon$, and the required computational effort is λ/ε, which is *increasing in* λ. Unlike for the

$M/M/1$ queue, as the arrival rate λ increases, the bias (starting empty) eventually becomes the dominant factor in the required computational effort.

For this $M/M/\infty$ model, it is natural to pay more attention to bias than we would with a single-server queue. A simple approach is to choose a different initial condition. The bias is substantially reduced if we start with a fixed number of busy servers not too different from the steady-state mean λ. Indeed, if we start with exactly λ busy servers (assuming that λ is an integer), then the bias is asymptotically negligible as λ increases. Note, however, that this special initial condition does not directly put the system into steady state, because the steady-state distribution is Poisson, not deterministic.

If, instead, we were working with the $M/G/\infty$ model, then in addition we would need to specify the remaining service times of all the λ customers initially in service at time 0. Fortunately, for the $M/G/\infty$ model, there is a natural way to do this: The steady-state distribution of the number of busy servers is again Poisson with mean λ, just as for the $M/M/\infty$ model. In addition, in steady-state, conditional upon the number of busy servers, the remaining service times of those customers in service are distributed as i.i.d. random variables with the *stationary-excess* (or equilibrium residual-life) cumulative distribution function (c.d.f.) G_{e} associated with the service-time c.d.f. G, i.e.,

$$G_{\mathrm{e}}(t) \equiv m^{-1} \int_0^t \bigl[1 - G(u)\bigr]\,\mathrm{d}u, \tag{5}$$

where m is the mean of G (here $m = 1$); e.g., see Takács (1962).

It is natural to apply this insight to more general many-server queueing models. Even in more general $G/G/s$ models, it is natural to initialize the simulation by putting s customers in the system at time 0 and letting their remaining service times be distributed as s i.i.d. random variables with c.d.f. G_{e}. For large s, that should be much better than starting the system empty.

For many-server queues, we may be interested in different congestion measures. By Little's law ($L = \lambda W$), we know that the expected steady-state number of busy servers in the $G/G/s/\infty$ model is exactly λ (provided that $\lambda < s$). Thus, in simulation experiments, we usually are more interested in estimating quantities such as $\mathrm{E}[(Q(\infty) - s)^+]$, where $(x)^+ \equiv \max\{0, x\}$, or $\mathrm{P}(Q(\infty) > s + k)$. Note that we can calculate the asymptotic bias and the asymptotic variance for these quantities in the $M/M/s$ model by applying the BD recursion with appropriate functions f. With large s, it often helps to start the recursion at s and move away in both directions. The initial value at s can be taken to be 1; afterwards the correct value is obtained by choosing the appropriate normalizing constant.

5 Diffusion processes

Diffusion processes are continuous analogues of BD processes; e.g., see Karlin and Taylor (1981) and Browne and Whitt (1995). In this chapter we discuss diffusion processes because we are interested in them as approximations of other processes that we might naturally simulate using discrete-event simulation. We want to use the diffusion processes to approximate the asymptotic bias and the asymptotic variance of sample means in the original process.

Diffusion processes tend to be complicated to simulate directly because they have continuous, continuously fluctuating, sample paths. Nevertheless, there also is great interest in simulating diffusion processes and stochastic differential equations, e.g., for finance applications, and special techniques have been developed; see Kloeden et al. (1994) and Kloeden and Platen (1995). Hence the analysis in this section may have broader application.

For diffusion processes, there are integral representations of the asymptotic parameters, paralleling the sums exhibited for BD processes. Corresponding to the finite-state-space assumption for the BD processes, assume that the state space of the diffusion is the finite interval $[s_1, s_2]$ and let the diffusion be reflecting at the boundary points s_1 and s_2, but under regularity conditions the integral representations will be valid over unbounded intervals. Let $\{Y(t)\colon\ t \geqslant 0\}$ be the diffusion process and let $X(t) = f(Y(t))$ for a real-valued function f. The diffusion process is characterized by its drift function $\mu(x)$ and its diffusion function $\sigma^2(x)$.

Let π be the stationary probability density. The stationary probability density can be represented as

$$\pi(y) = \frac{m(y)}{M(s_2)}, \quad s_1 \leqslant y \leqslant s_2,$$

where

$$m(y) \equiv \frac{2}{\sigma^2(y)s(y)}$$

is the *speed density*,

$$s(y) \equiv \exp\left\{-\int_{s_1}^{y} \frac{2\mu(x)}{\sigma^2(x)}\,\mathrm{d}x\right\}$$

is the *scale density* and

$$M(y) = \int_{s_1}^{y} m(x)\,\mathrm{d}x, \quad s_1 \leqslant y \leqslant s_2,$$

provided that the integrals are finite.

Let $p(t, x, y)$ be the transition kernel. Then, paralleling the fundamental matrix of a CTMC, we can define the *fundamental function of a diffusion process*, $Z \equiv Z(x, y)$, by

$$Z(x, y) \equiv \int_0^{\infty} \bigl[p(t, x, y) - \pi(y)\bigr]\,\mathrm{d}t.$$

As before, let μ be the average of f with respect to the stationary probability density π, i.e.,

$$\mu = \int_{s_1}^{s_2} \pi(x) f(x)\,\mathrm{d}x.$$

Then the integral representations for the asymptotic bias $\bar{\beta}(\xi)$ starting with initial probability density ξ and the asymptotic variance $\bar{\sigma}^2$ are

$$\bar{\beta}(\xi) = 2\int_{s_1}^{s_2} \frac{1}{\sigma^2(y)\pi(y)} \times \left[\int_{s_1}^{y} \bigl(f(x) - \mu\bigr)\pi(x)\,\mathrm{d}x \int_{s_1}^{y} \bigl(\xi(z) - \pi(z)\bigr)\,\mathrm{d}z\right]\mathrm{d}y$$

and

$$\bar{\sigma}^2 = 4\int_{s_1}^{s_2} \frac{1}{\sigma^2(y)\pi(y)}\left[\int_{s_1}^{y} \bigl(f(x) - \mu\bigr)\pi(x)\,\mathrm{d}x\right]^2 \mathrm{d}y.$$

We now discuss two examples of diffusion processes, which are especially important because they arise as limit processes for queueing models, as we explain in Section 6.

Example 4 (RBM). Suppose that the diffusion process is reflected Brownian motion (RBM) on the interval $[0, \infty)$ with drift function $\mu(x) = a$ and diffusion function $\sigma^2(x) = b$, where $a < 0 < b$, which we refer to by RBM(a, b); see Harrison (1985), Whitt (2002) and references therein for more background. RBM is the continuous analog of the queue-length process for the $M/M/1$ queue (as we will explain in the next section). It is a relatively simple stochastic process with only the two parameters a and b.

In fact, we can analyze the RBM(a, b) processes by considering only the special case in which $a = -1$ and $b = 1$, which we call *canonical* RBM because there are *no free parameters*. We can analyze RBM(a, b) in terms of RBM$(-1, 1)$ because we can relate the two RBMs by appropriately scaling time and space. For that purpose, let $\{R(t; a, b, X)\colon t \geqslant 0\}$ denote RBM(a, b) with initial distribution according to the random variable X. The key relation between the general RBM and canonical RBM is

$$\bigl\{R(t; a, b, X)\colon t \geqslant 0\bigr\} \stackrel{\mathrm{d}}{=} \bigl\{c^{-1}R\bigl(d^{-1}t; -1, 1, cX\bigr)\colon t \geqslant 0\bigr\}$$

or, equivalently,

$$\bigl\{R(t; -1, 1, X)\colon t \geqslant 0\bigr\} \stackrel{\mathrm{d}}{=} \left\{cR\left(dt; a, b, \frac{X}{c}\right)\colon t \geqslant 0\right\},$$

where

$$c = \frac{|a|}{b}, \qquad d = \frac{b}{a^2}, \qquad a = \frac{-1}{cd} \quad \text{and} \quad b = \frac{1}{c^2 d},$$

where "$\overset{d}{=}$" means equal in distribution (here as stochastic processes).

Hence it suffices to focus on canonical RBM. It has stationary density $\pi(x) = 2e^{-2x}$, $x \geqslant 0$. If we initialize RBM with its stationary distribution, then we obtain a stationary process. Let $R^* \equiv \{R^*(t; a, b)\colon\ t \geqslant 0\}$ denote *stationary* RBM, initialized by the stationary distribution.

If $f(x) = x$ for canonical RBM, then we would be estimating the steady-state mean $\mu = 1/2$. In this case, the asymptotic bias is $\bar{\beta}(0) = -1/4$ (Theorem 1.3 of Abate and Whitt, 1987a) and the asymptotic variance (for R^*) is $\bar{\sigma}^2 = 1/2$ (Abate and Whitt, 1988b).

To describe the general RBM with parameters a and b, we apply the scaling relations in Section 6.1. As a consequence of those scaling properties, the mean and variance of the steady-state distribution of RBM(a, b) are

$$\mu_{a,b} = \frac{b}{2|a|} \quad \text{and} \quad \sigma^2_{a,b} = \mu^2_{a,b} = \frac{b^2}{4a^2},$$

and the asymptotic parameters are

$$\bar{\beta}_{a,b}(0) = \frac{-b^2}{4|a|^3} \quad \text{and} \quad \bar{\sigma}^2_{a,b} = \frac{b^3}{2a^4}.$$

For the relative-width criterion, the key ratios are

$$\frac{\bar{\beta}_{a,b}(0)}{\mu_{a,b}} = \frac{-b}{2a^2} \quad \text{and} \quad \frac{\bar{\sigma}^2_{a,b}}{\mu^2_{a,b}} = \frac{2b}{a^2}.$$

Thus we see that the relative asymptotic bias is about the same as the relative asymptotic variance. Since the bias of the sample mean $\overline{X}_t$ is of order $O(1/t)$, while the square root of the variance of the sample mean $\overline{X}_t$ is of order $O(1/\sqrt{t}\,)$, the bias tends to be negligible for large t.

Example 5 (OU). Suppose that the diffusion process is the Ornstein–Uhlenbeck (OU) diffusion process on the interval $(-\infty, \infty)$ with drift function $\mu(x) = ax$ and diffusion function $\sigma^2(x) = b$, where $a < 0 < b$, which we refer to as OU(a, b). It is the continuous analog of the queue-length process in the $M/M/\infty$ queue when we center appropriately.

We also can analyze the OU(a, b) processes by considering only the special case in which $a = -1$ and $b = 1$, which we call *canonical* OU. We can analyze OU(a, b) in terms of OU$(-1, 1)$ because we can relate the two OUs by appropriately scaling time and space, just as we did for RBM. For that purpose, let $\{Z(t; a, b, X)\colon\ t \geqslant 0\}$ denote OU(a, b) with initial distribution according to the random variable X. The key relation between the general OU(a, b) and canonical OU$(-1, 1)$ is

$$\bigl\{Z(t; a, b, X)\colon\ t \geqslant 0\bigr\} \overset{d}{=} \bigl\{c^{-1} Z\bigl(d^{-1}t; -1, 1, cX\bigr)\colon\ t \geqslant 0\bigr\}$$

or, equivalently,

$$\{Z(t; -1, 1, X)\colon\ t \geqslant 0\} \overset{\mathrm{d}}{=} \left\{cZ\left(\mathrm{d}t; a, b, \frac{X}{c}\right)\colon\ t \geqslant 0\right\},$$

where

$$c = \frac{|a|}{b}, \qquad d = \frac{b}{a^2}, \qquad a = \frac{-1}{cd} \quad \text{and} \quad b = \frac{1}{c^2 d}.$$

Then the stationary density of canonical OU is normal with mean 0 and variance $1/2$. The mean of canonical OU starting at x is

$$\mathrm{E}\big[Z(t; -1, 1, x)\big] = x\mathrm{e}^{-t}, \quad t \geqslant 0.$$

Paralleling our treatment of RBM, let $Z^* \equiv \{Z^*(t; a, b)\colon\ t \geqslant 0\}$ be *stationary* OU, obtained by initializing the $\mathrm{OU}(a, b)$ with the stationary normal distribution. For stationary canonical OU, the autocovariance function is

$$\mathrm{Cov}\big(Z^*(0), Z^*(t)\big) = \frac{1}{2}\mathrm{e}^{-t}, \quad t \geqslant 0.$$

Hence, the asymptotic parameters for canonical OU are

$$\bar{\beta}(\xi) = \xi \quad \text{and} \quad \bar{\sigma}^2 = \frac{1}{2}.$$

Just as with RBM, we can apply Section 6.1 to determine the effect of scaling. The mean and variance of the steady-state distribution of $\mathrm{OU}(a, b)$ are

$$\mu_{a,b} = 0 \quad \text{and} \quad \sigma^2_{a,b} = \frac{b}{2|a|},$$

and the asymptotic parameters are

$$\bar{\beta}_{a,b}(x) = x\frac{b^2}{|a|^3} \quad \text{and} \quad \bar{\sigma}^2_{a,b} = \frac{b^3}{2a^4}.$$

The relative-width criterion makes less sense here because the random variables are not nonnegative.

6 Stochastic-process limits

In this section we discuss stochastic-process limits that make the RBM and OU diffusion processes serve as useful approximations for queueing models. We start by discussing the impact of scaling space and time. The scaling is often the key part.

6.1 Scaling of time and space

To obtain relatively simple approximate stochastic processes, we often consider stochastic-process limits, as in Whitt (2002). (We elaborate further.) To establish appropriate stochastic-process limits, we usually consider not just one stochastic process but a family of stochastic processes constructed by scaling time and space. It is thus important to know how the asymptotic parameters change under such scaling.

Suppose that we have a stochastic process $Z \equiv \{Z(t)\colon\ t \geqslant 0\}$ and we want to consider the scaled stochastic process $Z_{u,v} \equiv \{Z_{u,v}(t)\colon\ t \geqslant 0\}$, where

$$Z_{u,v}(t) \equiv uZ(vt), \quad t \geqslant 0,$$

for positive real numbers u and v. Suppose that $Z(t) \Rightarrow Z(\infty)$ as $t \to \infty$. Then $Z_{u,v}(t) \Rightarrow Z_{u,v}(\infty)$ as $t \to \infty$, where

$$Z_{u,v}(\infty) = uZ(\infty).$$

Let μ be the mean and σ^2 the variance of $Z(\infty)$; let $\mu_{u,v}$ be the mean and $\sigma^2_{u,v}$ the variance of $Z_{u,v}(\infty)$. Then

$$\mu_{u,v} = u\mu \quad \text{and} \quad \sigma^2_{u,v} = u^2\sigma^2.$$

The relation is different for the asymptotic parameters: Observe that $\mathrm{E}Z_{u,v}(t) = u\,\mathrm{E}Z(vt)$ for $t \geqslant 0$ and, under the assumption that Z is a stationary process,

$$\operatorname{Cov}\bigl(Z_{u,v}(0), Z_{u,v}(t)\bigr) = u^2 \operatorname{Cov}\bigl(Z(0), Z(vt)\bigr), \quad t \geqslant 0.$$

As a consequence, the asymptotic bias and the asymptotic variance are

$$\bar{\beta}_{u,v} = \frac{u}{v}\bar{\beta} \quad \text{and} \quad \bar{\sigma}^2_{u,v} = \frac{u^2}{v}\bar{\sigma}^2.$$

Thus, once we have determined the asymptotic parameters of a stochastic process of interest, it is easy to obtain the asymptotic parameters of associated stochastic processes constructed by scaling time and space. If the scaling parameters u and v are either very large or very small, then the scaling can have a great impact on the required run length. Indeed, as we show below, in standard queueing examples the scaling is dominant.

6.2 RBM approximations

Consider the queue-length (number in system) stochastic process $\{Q_\rho(t)\colon\ t \geqslant 0\}$ in the $G/G/s/\infty$ with traffic intensity (rate in divided by maximum rate out) ρ, with time units fixed by letting the mean service time be 1, without the usual independence assumptions. As reviewed in Whitt (1989, 2002), in remarkable generality (under independence assumptions and beyond), there is a *heavy-traffic stochastic-process limit* for the scaled queue-length processes, obtained by dividing time t by $(1-\rho)^2$ and multiplying space by $(1-\rho)$, i.e.,

$$\{(1-\rho)Q_\rho(t(1-\rho)^{-2})\colon\ t \geqslant 0\} \Rightarrow \{R(t;a,b)\colon\ t \geqslant 0\} \quad \text{as } \rho \uparrow 1$$

for appropriate parameters a and b, where $\{R(t;a,b)\colon\ t \geqslant 0\}$ is $\mathrm{RBM}(a,b)$ and again "$\Rightarrow$" denotes convergence in distribution, but here in the function space D containing all sample paths.

The limit above is very helpful, because the number of relevant parameters has been greatly reduced. We see that the queue behavior for large ρ should primarily depend upon only ρ and the two parameters a and b. Moreover, it turns out the parameters a and b above can be conveniently characterized (in terms of scaling constants in central limit theorems for the arrival and service processes). For example, in the standard $GI/GI/s/\infty$ model the heavy-traffic limit holds with

$$a = -s \quad \text{and} \quad b = s(c_a^2 + c_s^2),$$

where c_a^2 and c_s^2 are the SCVs of an interarrival time and a service time, respectively (provided that the second moments are finite). Similar limits hold for workload processes, recording the amount of remaining unfinished work in service time in the system.

We thus can apply the stochastic-process limit with the scaling properties in Section 6.1 and the properties of RBM to obtain approximations paralleling the exact results for the $M/M/1$ queue. We apply the stochastic-process limit to obtain the approximation

$$\{Q_\rho(t)\colon\ t \geqslant 0\} \approx \{(1-\rho)^{-1}R(t(1-\rho)^2;a,b)\colon\ t \geqslant 0\}.$$

The resulting approximations for the mean and variance of the steady-state distribution of the queue-length process are thus

$$\mathrm{E}[Q_\rho(\infty)] \approx \frac{b}{2|a|(1-\rho)} \quad \text{and} \quad \sigma_\rho^2 \equiv \mathrm{Var}(Q_\rho(\infty)) \approx \frac{b^2}{4a^2(1-\rho)^2};$$

the approximations for the asymptotic parameters are

$$\bar{\beta}_\rho(0) \approx \frac{-b^2}{4|a|^3(1-\rho)^3} \quad \text{and} \quad \bar{\sigma}_\rho^2 \approx \frac{b^3}{2a^4(1-\rho)^4}.$$

In the $GI/GI/s/\infty$ case, we just substitute the specific parameters a and b above. The resulting approximate asymptotic variance is

$$\bar{\sigma}_\rho^2 \equiv \bar{\sigma}^2_{(s,\rho,c_a^2,c_s^2)} = \frac{(c_a^2+c_s^2)^3}{2s(1-\rho)^4}.$$

Note that these formulas agree with the limits of the $M/M/1$ formulas as $\rho \uparrow 1$. Thus, we see that the $M/M/1$ formulas are remarkably descriptive more generally. But we also see the impact of s servers and the GI arrival and service processes: The asymptotic variance is directly proportional to $1/s$ and to the *third power* of the overall "variability parameter" $(c_a^2 + c_s^2)$ as well as to the *fourth* power of $(1-\rho)^{-1}$.

More generally, we see how the parameters s, ρ, a and b in more general $G/G/s/\infty$ models (with nonrenewal arrival processes and non-i.i.d. service times) will affect the required simulation run length. Once we have established the corresponding heavy-traffic limit and identified the new values of a and b for these alternative models, we can apply the results above. For the relative-width criterion, the key ratios are

$$\frac{\bar{\beta}_{a,b}(0)}{\mu_{a,b}} \approx \frac{-b}{2a^2(1-\rho)^2} \quad \text{and} \quad \frac{\bar{\sigma}^2_{a,b}}{\mu^2_{a,b}} \approx \frac{2b}{a^2(1-\rho)^2}.$$

Values of the key parameters a and b in alternative models have been determined; e.g., see Sections 5.2–5.5 of Whitt (1989) and Fendick et al. (1989).

6.3 Many-server queues

The analysis above applies to multiserver queues, but when the number of servers is large, the RBM approximation tends to be inappropriate. When the number of servers is large, it is often preferable to consider different limits in which the number s of servers is allowed to increase as the arrival rate λ increases; see Halfin and Whitt (1981), Chapter 10 of Whitt (2002), Whitt (2005) and references therein. Alternatively, when there is a large number of servers, a more elementary direct approach is to consider an infinite-server model as an approximation for the model with finitely many servers. We thus might approximate the queue-length (number in system) process in the $G/G/s$ model (with finite or infinite waiting room) by the stochastic process representing the number of busy servers in the associated $G/G/\infty$ model.

When we do consider the $G/G/\infty$ model, it is natural to develop approximations based on heavy-traffic stochastic-process limits, where now heavy-traffic is defined by having the arrival rate λ increase without bound. For that purpose, let $Q_\lambda(t)$ denote the number of busy servers at time t as a function of the arrival rate λ. Again, under regularity conditions, there is a heavy-traffic stochastic-process limit, but it takes a different form. Now

$$\left\{\frac{Q_\lambda(t)-\lambda}{\sqrt{\lambda}}\colon\, t \geqslant 0\right\} \Rightarrow \left\{L(t)\colon\, t \geqslant 0\right\} \quad \text{as } \lambda \to \infty,$$

where the limit process $L \equiv \{L(t)\colon\, t \geqslant 0\}$ is a zero-mean Gaussian process, i.e., for which $(L(t_1), \ldots, L(t_k))$ has a k-dimensional normal distribution for all positive integers k and all time points $0 < t_1 < \cdots < t_k$; see Section 10.3 of Whitt (2002), Glynn and Whitt (1991) and references therein. As with the previous RBM limit, this limit generates a natural approximation; here it is

$$\left\{Q(t)\colon\, t \geqslant 0\right\} \approx \left\{\lambda + \sqrt{\lambda}L(t)\colon\, t \geqslant 0\right\}.$$

For the $G/M/\infty$ special case, when the service times are i.i.d. and exponentially distributed (still with mean 1), the Gaussian limit process L is $\mathrm{OU}(-1, 1 + c_a^2)$, where c_a^2 is the normalization constant in the central limit theorem for the arrival process, corresponding to the SCV of an interarrival time in the renewal (GI) case. Thus the approximate asymptotic variance is

$$\bar{\sigma}^2 \equiv \bar{\sigma}^2_{Q_\lambda} \approx \lambda \frac{(1 + c_a^2)^3}{2}.$$

For the more general $G/GI/\infty$ model, when the service times are *not* exponentially distributed, the limiting Gaussian process is *not* Markov (except if G is a mixture of an exponential and a point mass at 0; see Glynn (1982)). If G denotes the c.d.f. of the service time and $G^{\mathrm{c}}(t) \equiv 1 - G(t)$ is the associated complementary c.d.f. (c.c.d.f.), then the autocovariance function of the stationary version L^* of the limit process L is

$$\begin{aligned}&\operatorname{Cov}\bigl(L^*(0), L^*(t)\bigr)\\ &\quad = \int_0^\infty G(u) G^{\mathrm{c}}(t + u)\,\mathrm{d}u + c_a^2 \int_0^\infty G^{\mathrm{c}}(t + u) G^{\mathrm{c}}(u)\,\mathrm{d}u.\end{aligned}$$

In the special case of the $M/GI/\infty$ model, $c_a^2 = 1$ and the autocovariance function simplifies, becoming

$$\operatorname{Cov}\bigl(L^*(0), L^*(t)\bigr) = \int_0^\infty G^{\mathrm{c}}(t + u)\,\mathrm{d}u = G^{\mathrm{c}}_{\mathrm{e}}(t),$$

where $G^{\mathrm{c}}_{\mathrm{e}}$ is c.c.d.f. associated with the stationary-excess c.d.f. G_{e} in (5). Since G_{e} has mean $(c_s^2 + 1)/2$, the asymptotic variance of L^* is $(c_s^2 + 1)/2$. Thus, for the $M/GI/\infty$ model the approximate asymptotic variance of Q_λ is

$$\bar{\sigma}^2 \equiv \bar{\sigma}^2_{Q_\lambda} \approx \frac{\lambda(c_s^2 + 1)}{2}.$$

With many-server queues, we are often less interested in the queue-length process than other stochastic processes. For example in the $M/M/s/0$ (Erlang loss or B) model, which has s servers, no extra waiting room and arrivals blocked or lost without affecting future arrivals when arrivals find all servers busy, instead of the number of busy servers, we are often interested in the steady-state blocking probability. In the corresponding $M/M/s/\infty$ (Erlang delay or C) model, which has s servers and unlimited extra waiting room, we are often interested in the steady-state delay probability, i.e., the probability that an arrival must wait before beginning service, or the probability that an arrival must have to wait more than some designated time, such as 20 seconds (a common target in telephone call centers).

To consider the simulation run length required to estimate these alternative characteristics, we may nevertheless use the infinite-server model and the analysis above as a rough guide. To get better estimates we can consider the multi-server model with s servers (instead of letting $s = \infty$). It has been found useful to consider limits in which s increases along with λ. It turns out to be appropriate to let s and λ increase so that

$$\frac{\lambda - s}{\sqrt{\lambda}} \to \gamma \quad \text{as } \lambda \to \infty.$$

Then, just as in the infinite-server case, under regularity conditions there is again a heavy-traffic limit for $[Q_\lambda(t) - \lambda]/\sqrt{\lambda}$ but now with a different limit process L; see Halfin and Whitt (1981), Srikant and Whitt (1996, 1999), Puhalskii and Reiman (2000) and Whitt (2005). That in turn allows us to approximate the asymptotic variance and estimate the required simulation run length. The issue of required simulation run lengths for many-server loss systems is the main focus of Srikant and Whitt (1996, 1999).

7 Deleting an initial portion of the run to reduce bias

We have seen that for various Markov processes we can estimate the bias of a sample mean associated with any contemplated initial distribution and simulation run length. If the estimated bias is too large, then we can try to reduce the bias by choosing alternative initial conditions. We can estimate the bias reduction gained by choosing alternative initial distributions, because the asymptotic bias $\bar{\beta}(\xi)$ is a function of the initial probability distribution ξ.

If the estimated bias is too large, and it is difficult to change the initial conditions, then we might instead consider not collecting data for an initial portion of the simulation run, given the natural initial conditions. However, it is more difficult to estimate the bias reduction from not collecting data from an initial portion of the run. For that purpose, we need to know the time-dependent mean $\mathrm{E}[X(t)]$, where $\{X(t)\colon\ t \geqslant 0\}$ is the stochastic process being observed. The asymptotic bias when we do not collect data over an initial interval $[0, c]$ is

$$\bar{\beta}(\xi, c) = \int_c^\infty \bigl(\mathrm{E}X(t) - \mu\bigr)\,\mathrm{d}t.$$

A rough approximation for the asymptotic bias $\bar{\beta}(\xi, c)$ can be based on the exponential approximation

$$\mathrm{E}\bigl[X(t)\bigr] - \mu \approx \mathrm{e}^{-t/\bar{\beta}}, \quad t \geqslant 0,$$

where the parameter $\bar{\beta}$ is chosen to yield the correct asymptotic bias $\bar{\beta} = \bar{\beta}(\xi, 0)$. Then we obtain the approximation

$$\bar{\beta}(\xi, c) \approx \int_c^\infty \mathrm{e}^{-t/\bar{\beta}}\,\mathrm{d}t = \bar{\beta}\mathrm{e}^{-c/\bar{\beta}}.$$

Unfortunately, however, the exponential approximation is not very reliable, because the time-dependent mean rarely has such a simple exponential form. For better estimates of the reduced bias, we need to estimate the time-dependent mean $\mathrm{E}X(t)$. Fortunately, for some commonly occurring stochastic processes, expressions for the time-dependent mean are available. For example, exact and approximate expressions for the time-dependent mean for $M/M/1$ and RBM are contained in Abate and Whitt (1987a, 1987b, 1988b).

For the $M/GI/\infty$ model with arrival rate λ and service-time c.d.f. G, starting empty,

$$\mathrm{E}\bigl[Q_\lambda(t)\bigr] = \mathrm{E}\bigl[Q_\lambda(\infty)\bigr]G_{\mathrm{e}}(t) = \lambda G_{\mathrm{e}}(t), \quad t \geqslant 0,$$

where G_{e} is the stationary-excess c.d.f., just as in the covariance function; see Section 4 of Eick et al. (1993). So the asymptotic bias is $-\lambda(c_s^2+1)/2$, just like the asymptotic variance.

8 Directions for further research

We described two classes of models that have been analyzed rather thoroughly to understand the required simulation run lengths: single-server queues and many-server queues (here approximated by infinite-server queues). Other important classes of stochastic models should be analyzed in the same way.

The analysis here is based on the normal approximation for the sample mean reviewed in Section 2. The conditions in the central limit theorem yielding the normal approximation are not satisfied when there are heavy-tailed distributions or long-range dependence. Since these features tend to arise in practice, it is natural to include them in simulations. As can be seen from Chapter 4 of Whitt (2002), there is alternative asymptotic theory for heavy-tailed distributions or long-range dependence, and there is a body of statistical techniques, as in Beran (1994), but more needs to be done to plan simulations in that context. In general, simulations in face of such stochastic complexity are difficult. Work to cope with such complexity is described in Chapter 11.

Acknowledgement

The author was supported by National Science Foundation Grant DMS-02-2340.

References

Abate, J., Whitt, W. (1987a). Transient behavior of regulated Brownian motion, I and II. *Advances in Applied Probability* 19, 560–631.

Abate, J., Whitt, W. (1987b). Transient behavior of the $M/M/1$ queue: Starting at the origin. *Queueing Systems* 2, 41–65.

Abate, J., Whitt, W. (1988a). Transient behavior of the $M/M/1$ queue via Laplace transforms. *Advances in Applied Probability* 20, 145–178.

Abate, J., Whitt, W. (1988b). The correlation functions of *RBM* and $M/M/1$. *Stochastic Models* 4, 315–359.

Asmussen, S. (2003). *Applied Probability and Queues*, 2nd edition. Springer-Verlag, New York.

Beran, J. (1994). *Statistics for Long-Memory Processes*. Chapman and Hall, London.

Bratley, P., Fox, B.L., Schrage, L.E. (1987). *A Guide to Simulation*, 2nd edition. Springer-Verlag, New York.

Browne, S., Whitt, W. (1995). Piecewise-linear diffusion processes. In: Dshalalow, J. (Ed.), *Advances in Queueing*. CRC Press, Boca Raton, FL, pp. 463–480.

Chen, H., Yao, D.D. (2001). *Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization*. Springer-Verlag, New York.

Cochran, W.G., Cox, G.M. (1992). *Experimental Designs*, 2nd edition. Wiley, New York.

Cohen, J.W. (1982). *The Single Server Queue*, 2nd edition. North-Holland, Amsterdam.

Cooper, R.B. (1982). *Introduction to Queueing Theory*, 2nd edition. North-Holland, New York.

Eick, S.G., Massey, W.A., Whitt, W. (1993). The physics of the $M_t/G/\infty$ queue. *Operations Research* 41, 731–742.

Fendick, K.W., Saksena, V.R., Whitt, W. (1989). Dependence in packet queues. *IEEE Transactions on Communications* 37, 1173–1183.

Fishman, G.S. (2001). *Discrete Event Simulation*. Springer-Verlag, New York.

Glynn, P.W. (1982). On the Markov property of the $GI/G/\infty$ Gaussian limit. *Advances in Applied Probability* 14, 191–194.

Glynn, P.W. (1984). Some asymptotic formulas for Markov chains with application to simulation. *Journal of Statistical Computation and Simulation* 19, 97–112.

Glynn, P.W. (1989). A GSMP formalism for discrete-event systems. *Proceedings of the IEEE* 77, 14–23.

Glynn, P.W. (1994). Poisson's equation for the recurrent $M/G/1$ queue. *Advances in Applied Probability* 26, 1044–1062.

Glynn, P.W., Meyn, S. (1996). A Liapounov bound for solutions of the Poisson equation. *The Annals of Probability* 24, 916–931.

Glynn, P.W., Whitt, W. (1991). A new view of the heavy-traffic limit for infinite-server queues. *Advances in Applied Probability* 23, 188–209.

Glynn, P.W., Whitt, W. (1992). The asymptotic efficiency of simulation estimators. *Operations Research* 40, 505–520.

Grassman, W.K. (1987a). The asymptotic variance of a time average for a birth–death process. *Annals of Operations Research* 8, 165–174.

Grassman, W.K. (1987b). Means and variances of time averages in Markovian environments. *European Journal of Operations Research* 31, 132–139.

Halfin, S., Whitt, W. (1981). Heavy-traffic limits for queues with many exponential servers. *Operations Research* 29, 567–588.

Harrison, J.M. (1985). *Brownian Motion and Stochastic Flow Systems*. Wiley, New York.

Karlin, S., Taylor, H.M. (1981). *A Second Course in Stochastic Processes*. Academic Press, San Diego, CA.

Kelly, F.P. (1979). *Reversibility and Stochastic Networks*. Wiley, Chichester.

Kloeden, P.E., Platen, E. (1995). *Numerical Solution of Stochastic Differential Equations*. Springer-Verlag, Berlin.

Kloeden, P.E., Platen, E., Schurz, H. (1994). *Numerical Solution of SDE Through Computer Experiments*. Springer-Verlag, Berlin.

Lehmann, E.L., Casella, G. (1998). *Theory of Point Estimation*, 2nd edition. Wiley, New York.

Montgomery, D.C. (2000). *Design and Analysis of Experiments*, 5th edition. Wiley, New York.

NIST/SEMATECH (2003). Exploratory data analysis. In: *e-Handbook of Statistical Methods*, Chapter 1. Available at *http://www.itl.nist.gov/div898/handbook/*.

Papadimitriou, C.H. (1994). *Computational Complexity*. Addison–Wesley, Reading, MA.

Puhalskii, A.A., Reiman, M.I. (2000). The multiclass $GI/PH/N$ queue in the Halfin–Whitt regime. *Advances in Applied Probability* 32, 564–595.

Samorodnitsky, G., Taqqu, M.S. (1994). *Stable Non-Gaussian Random Processes*. Chapman and Hall, New York.

Sigman, K. (1995). *Stationary Marked Point Processes, An Intuitive Approach*. Chapman and Hall, New York.

Srikant, R., Whitt, W. (1996). Simulation run lengths to estimate blocking probabilities. *ACM Transactions on Modeling and Computer Simulation* 6, 7–52.

Srikant, R., Whitt, W. (1999). Variance reduction in simulations of loss models. *Operations Research* 47, 509–523.

Takács, L. (1962). *Introduction to the Theory of Queues*. Oxford University Press, New York.

Tukey, J. (1977). *Exploratory Data Analysis*. Addison–Wesley, Reading, MA.

Velleman, P., Hoaglin, D. (1981). *The ABC's of EDA: Applications, Basics, and Computing of Exploratory Data Analysis*. Duxbury, Belmont, CA.

Walrand, J. (1988). *An Introduction to Queueing Networks*. Prentice-Hall, Englewood Cliffs, NJ.

Whitt, W. (1989). Planning queueing simulations. *Management Science* 35, 1341–1366.

Whitt, W. (1992). Asymptotic formulas for Markov processes. *Operations Research* 40, 279–291.

Whitt, W. (2002). *Stochastic-Process Limits*. Springer-Verlag, New York.

Whitt, W. (2005). Heavy-traffic limits for the $G/H_2^*/n/m$ queue. *Mathematics of Operations Research* 30 (1), 1–27.

Wu, C.F.J., Hamada, M.S. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. Wiley, New York.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13014-5

Chapter 14

Resampling Methods

R.C.H. Cheng

Faculty of Mathematical Studies, University of Southampton, Southampton SO17 1BJ, UK
E-mail: R.C.H.Cheng@maths.soton.ac.uk

Abstract

Bootstrap resampling is an extremely practical and effective way of studying the distributional properties of statistical data. It requires computer intensive methods, but these are often already incorporated as data gathering procedures in the simulations themselves, and there should be little additional work in commandeering these procedures for use in the statistical analysis itself. Resampling is therefore often easy and straightforward to carry out in simulations. More importantly there is ample evidence both in theory and in practice that such methods can be more effective than classical statistical inference, especially in small samples. The purpose of this chapter is to review existing bootstrap methodology and to highlight the main methods and accompanying theoretical results that are of most use in the simulation context. We consider parametric as well as the classical nonparametric version of the bootstrap and show that they can handle a very wide variety of problems. The main message is that on the grounds of simplicity and effectiveness bootstrapping should be a statistical approach of first resort in simulation output analysis.

1 Introduction

Stochastic simulations typically generate data from which we compute a statistic of interest. This statistic is a random variable and the immediate, all-embracing question is: *What is its distribution*? A simple way of answering this question is as follows. We generate, not one, but a large number of independent and identically distributed data sets and for each data set we calculate the statistic of interest. This gives us a (large) random sample of the statistic. The empirical distribution function (EDF) formed from this random sample provides an immediate answer to our question.

Where the computer simulation runs are quick to make, the approach just described is an attractive and effective one for analyzing simulation output. It allows the distribution of even complicated statistical quantities to be obtained without having to resort to elaborate statistical procedures.

However, there are many simulation experiments where the simulation model is complex and it takes a significant time to make just one run, so that making a large number of runs is out of the question. The approach above of generating an EDF by repeated sampling does not then seem available.

Bootstrapping is a very effective way of recovering the situation by enabling a large number of data sets to be generated simply by *resampling from the original data set*. It is therefore an attractive approach that avoids having to make too many simulation runs when these are time consuming.

Given that sampling is very much part of the overall simulation experiment, it is very natural to turn to resampling methods for carrying out the statistical analysis. Used in this way the bootstrap approach often enables statistical procedures to be applied in a simple and painless manner without having to draw heavily on specialist statistical knowledge. This is an important advantage. However, there are common situations where bootstrapping, when applied properly, produces adjustments to estimators or confidence intervals with properties *superior* to those derived from asymptotic theory. Thus, there are clear theoretical as well as practical reasons for using bootstrapping.

The basic ideas of bootstrapping are quite general and apply not simply to simulation experiments. Our discussion reflects this. Many of the main results are applicable to any bootstrapping situation and not simply to simulation experiments. It should nevertheless be apparent how they can be applied in the simulation context.

Many of the situations where bootstrapping is appropriate are connected with the estimation of key quantities that characterize a system, such as average output or utilization rates. We are in consequence interested in estimating quantities or parameters whose estimators often have an asymptotic normal distribution. Chapter 8 discusses such estimators and their properties.

The discussion of this chapter focuses on independent samples. But in time series, and more general stochastic processes, the observations can be dependent. We discuss this in Section 9. For a more comprehensive view on correlation based methods, see Chapter 15.

An immediate issue is the balance between making ‘real’ simulation runs as opposed to bootstrapping. As will become evident, bootstrapping is usually cheap to do compared with making simulation runs. Therefore, it is generally best to expend most of the available computing effort on making ‘real’ simulation runs. In classical experiments all the real data gathering is done first, with the analysis carried out subsequently. However, there is no need to follow such a rigid procedure in simulation experiments. Even when simulation runs are relatively cheap, we can still alternate between simulation runs and bootstrapping. The bootstrapping gives information about the accuracy of the simulation results and thus provides a guide as to how best to expend subsequent simulation effort.

In compiling this chapter four sources were especially useful, and this is freely acknowledged here. Hjorth (1994) gives a very readable introductory account, though the early discussion of model selection and cross-validation gives a rather different focus to the material than that presented here. The book by Davison and Hinkley (1997) contains a wealth of practical advice on methodological and practical issues. Shao and Tu (1995) gives a detailed survey of theoretical work undertaken on the bootstrap. Hall (1992) provides a thorough study of the bootstrap and its properties in a very instructive and unified way via the theory of Edgeworth expansions. To these four should also be added the book by Chernick (1999) which contains a thorough survey of the major areas of application of the bootstrap together with a comprehensive bibliography. Finally we should point out that Efron and Tibshirani (1993) is a popular and very good recommendation for an introductory text, but we did not make significant use of it as a reference source, preferring instead to dwell a little more on slightly more mathematical aspects than on practical methodology.

Most of the literature on bootstrapping takes a general standpoint and does not have simulation studies specifically in mind. This chapter is not all that different in this regard especially in the sections up to Section 5. However, in reducing the material to chapter length we have attempted to select topics that seem most relevant and useful from the viewpoint of the analysis of simulation experiments, with a more explicit simulation viewpoint adopted from Section 5 onwards.

2 The bootstrap

2.1 The bootstrap concept

To start with we shall just consider samples $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_n)$ and statistics $s(\mathbf{Y})$ calculated from them. The Y_i may themselves be vectors, and they are not necessarily independent, though often they will be. However, we shall assume initially that the Y_i are mutually independent so that $\mathbf{Y}$ is a random sample. (The case of several samples is considered in Section 6 and the case when the Y_i are dependent will be discussed in Section 8.)

The reader anxious to relate this immediately to simulation may wish to have the following typical simulation situation in mind: Let Y be the observed value of a performance index of the system under investigation, obtained from one run of a simulation model. In a queueing system Y might represent average utilization. We make n runs, obtaining an observation Y_i from each. This is the random sample, $\mathbf{Y}$, under consideration. In this case $s(\mathbf{Y})$ might well be the sample mean $\overline{Y}$. For some initial discussion of bootstrapping as applied to simulation see Barton and Schruben (1993), Kim et al. (1993), Shiue et al. (1993) and Cheng (1995).

In fact practically all of inferential statistics reduces to trying to answer the question: 'What is $G(\cdot)$, the probability distribution of a statistic $s(\mathbf{Y})$, calculated from a sample $\mathbf{Y} = \mathbf{Y}(Y_1, Y_2, \ldots, Y_n)$?' Bootstrapping is simply a numerical way of answering this question. It might appear at first sight that bootstrapping attempts to get ‘something for nothing’. This is quite erroneous. Any statistical inference makes use of the observed values in some way and bootstrapping is no different. Repeated sampling of the observed values does lend bootstrapping its distinctive nature, but the inferences that are then drawn from the results of this resampling cannot provide any more information than is already in the sample, and in this respect bootstrapping is no different than any other method of analysis.

If we knew $F(\cdot)$, the distribution of Y, the above question, ‘What is $G(\cdot)$?’, is easily answered numerically. We simply generate a number, B, of independent samples of $\mathbf{Y}$: $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_B$, and calculate the statistic $s_j = s(\mathbf{y}_j)$ from each sample. Then the *empirical distribution function* (EDF) of the sample $\mathbf{s} = (s_1, s_2, \ldots, s_B)$,

$$G_n(s|\mathbf{s}) = \frac{\#\text{ of } s_j \leqslant s}{B},$$

will converge pointwise with probability one to $G(s)$. (Here and in what follows we shall use the notation $G_n(\cdot)$ to denote the EDF of a sample of size n drawn from the distribution $G(\cdot)$. Where the precise sample $\mathbf{s}$ used to construct G_n needs to be made explicit then we use the notation $G_n(\cdot|\mathbf{s})$ or $G_n(s|\mathbf{s})$). The basic sampling process is as shown in Figure 1.

The problem is that we usually do not know $F(\cdot)$. However we do have the EDF, $F_n(\cdot|\mathbf{y})$, formed from the sample $\mathbf{y}$. The bootstrap idea is to use $F_n(\cdot|\mathbf{y})$ instead of $F(\cdot)$ in the basic process above. To obtain a sample we draw values, not from $F(\cdot)$, but from $F_n(\cdot|\mathbf{y})$. This is equivalent to drawing a sample of the same size n, *with replacement*, from the original set of y's. We call such a sample a *bootstrap sample* to distinguish it from the original sample, and write it as $\mathbf{y}^* = (y_1^*, y_2^*, \ldots, y_n^*)$. As in the basic process, B such bootstrap samples are drawn, and the statistic is calculated from each of these samples: $s_j^* = s(\mathbf{y}_j^*)$. The EDF, $G_n(\cdot|\mathbf{s}^*)$, of these bootstrap statistics s_j^*, is our estimate of $G(\cdot)$. The bootstrap process is as shown in Figure 2. The underlying hope and expectation is that the bootstrap process will under many conditions reproduce the behavior of the original process. This is what we examine in detail in the remainder of the chapter.

```
The Basic Sampling Process
For $j = 1$ to $B$
    For $i = 1$ to $n$
        Draw $y_{ij}$ from $F(\cdot)$
    Next $i$
    Calculate $s_j = s(\mathbf{y}_j)$
Next $j$
Form $G_n(\cdot|\mathbf{s})$
```

Fig. 1. The basic sampling process.

```
The Bootstrap Sampling Process
Given a random sample $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ from $F(\cdot)$
Form the EDF $F_n(\cdot|\mathbf{y})$
For $j = 1$ to $B$
    For $i = 1$ to $n$
        Draw $y_{ij}^*$ from $F_n(\cdot|\mathbf{y})$
    Next $i$
    Calculate $s_j^* = s(\mathbf{y}_j^*)$
Next $j$
Form $G_n(\cdot|\mathbf{s}^*)$
```

Fig. 2. The bootstrap sampling process.

2.2 Basic method

We shall assume that the objective of the experiment (simulation or otherwise) is to use the random sample $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ to estimate a parameter $\eta = \eta(F(\cdot))$ that describes some quantity of interest of the distribution of Y. Typical examples are the mean

$$\eta = \int_{-\infty}^{\infty} y \, \mathrm{d}F(y)$$

or some other moment of Y, or a quantile $\eta = q(p)$, defined by

$$\int_{-\infty}^{q(p)} \mathrm{d}F(y) = p.$$

The statistic $s(\mathbf{Y})$ can then be regarded as any appropriate quantity that we may care to use for estimating η. We shall emphasize this view by using the alternative but entirely equivalent notation $\hat{\eta}(\mathbf{Y}) \equiv s(\mathbf{Y})$, with the 'hat' indicating an estimated quantity. Thus, when η is the mean, an obvious statistic is the sample mean

$$s(\mathbf{Y}) = \overline{Y}.$$

It is important to realize that we do not have to use the sample version as the estimator, assuming we can calculate it at all. In the present example we might use a trimmed sample mean, or even the median rather than the sample mean. What we *shall* do is regard the sample version of the parameter as being the quantity of interest in the bootstrap process, i.e., the parameter of interest in the bootstrap process is

$$\eta^* = \eta\big(F_n(\cdot)\big).$$

Thus we think of the bootstrap process as one where we are generating bootstrap samples $\mathbf{y}_j^*$ from which bootstrap estimates $s_j^* = s(\mathbf{y}_j^*)$ are obtained that estimate the bootstrap parameter η^*.

With this viewpoint, the bootstrap process is a prime example of the so-called *plug-in approach*, being a precise analogue of the original process with the only difference being that the known $F_n(\cdot)$ is plugged-in for, i.e., replaces the unknown $F(\cdot)$. The *bootstrap principle* is that the bootstrap process reproduces the properties of the original basic process.

Useful quantities that are obtained from the bootstrap process are the sample mean and variance of the bootstrap estimates

$$\bar{s}^* = \frac{1}{B}\sum_{j=1}^{B} s_j^*,$$

$$\sigma^{*2} = \frac{1}{B-1}\sum_{j=1}^{B}\left(s_j^* - \bar{s}^*\right)^2.$$

We also have an estimate of the bias of the mean of the bootstrap estimates: $b^* = \bar{s}^* - \eta^*$. A *bootstrap bias adjusted estimate* for η is thus

$$\breve{\eta} = \hat{\eta} - b^*.$$

If the bootstrap expectation satisfies $\mathrm{E}^*(s_j^*) = \hat{\eta}$ (here E^* denotes the conditional expectation $\mathrm{E}[s_j^* | F_n(\cdot|\mathbf{y})]$, where $F_n(\cdot)$ is treated as being the parent distribution), then $\bar{s}^* \to \hat{\eta}$ in probability as $B \to \infty$, and we have

$$\tilde{\eta} = \hat{\eta} - b^* = \hat{\eta} - \left(\bar{s}^* - \eta^*\right) \to \hat{\eta} - \left(\hat{\eta} - \eta^*\right) = \eta^*.$$

Thus, in this case, adjusting the original estimator for bias using the bootstrap bias is equivalent to using the bootstrap parameter η^* as the initial estimate.

2.3 *Parametric bootstrap*

We now consider parametric bootstrapping. Again we focus on a random sample $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ drawn from a distribution $F(y, \boldsymbol{\theta})$, with the difference that the form of F is known but which depends on a vector $\boldsymbol{\theta}$ of parameters that is unknown. We denote the unknown true value of $\boldsymbol{\theta}$ by $\boldsymbol{\theta}_0$. We then usually *estimate* $\boldsymbol{\theta}_0$ by some estimator $\hat{\boldsymbol{\theta}}$ calculated from the sample $\mathbf{y}$. Clearly $F(y, \hat{\boldsymbol{\theta}})$ will estimate $F(y, \boldsymbol{\theta}_0)$. Bootstrap samples $\mathbf{y}^*$ can then be drawn, not from the EDF $F_n(y|\mathbf{y})$, but from the distribution $F(y, \hat{\boldsymbol{\theta}})$. Sampling from $F(y, \hat{\boldsymbol{\theta}})$ instead of $F_n(y|\mathbf{y})$ is called *parametric bootstrapping*.

As with the basic bootstrap, the distribution of any statistic $s(\mathbf{Y})$ can be estimated from a sample of parametric bootstrap statistics $s(\mathbf{y}^{*(j)})$, $j = 1, 2, \ldots, B$. This includes where we might be interested in the distribution of the estimator $\hat{\boldsymbol{\theta}}$ itself, so that $s(\mathbf{Y}) = \hat{\boldsymbol{\theta}}$ in this case.

By far the most often employed estimator is where $\hat{\boldsymbol{\theta}}$ is the *maximum likelihood* (ML) estimator. In this particular case, under quite general regularity conditions, the asymptotic distribution of $\hat{\boldsymbol{\theta}}$ is known. We will later be considering situations where we wish to bootstrap from this asymptotic distribution.

We therefore gather a few key results together here. We have

$$n^{1/2}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \Rightarrow \mathrm{N}\big(\mathbf{0}, V(\boldsymbol{\theta}_0)\big) \quad \text{as } n \to \infty, \tag{1}$$

where $V(\boldsymbol{\theta}_0) = [I(\boldsymbol{\theta}_0)]^{-1}$, and $I(\boldsymbol{\theta}_0)$ is the *information matrix*. A reasonable approximation for $I(\boldsymbol{\theta}_0)$ is

$$I(\hat{\boldsymbol{\theta}}) \simeq -n^{-1} \frac{\partial^2 L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\bigg|_{\hat{\boldsymbol{\theta}}}. \tag{2}$$

More generally, suppose that η is a smooth function of $\boldsymbol{\theta}$, i.e., $\eta = \eta(\boldsymbol{\theta})$. The ML estimate of η is then simply $\hat{\eta} = \eta(\hat{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{\theta}}$ is the ML estimate of $\boldsymbol{\theta}$. The asymptotic normality of $\hat{\eta}$ follows from that of $\hat{\boldsymbol{\theta}}$, assuming that $\eta(\boldsymbol{\theta})$ can be approximated by a linear Taylor series in $\boldsymbol{\theta}$. This is the so-called *delta-method* of obtaining the asymptotic behavior of $\hat{\eta}$ and it yields the result

$$n^{1/2}(\hat{\eta} - \eta_0) \Rightarrow \mathrm{N}\big(0, \sigma^2(\boldsymbol{\theta}_0)\big) \quad \text{as } n \to \infty, \tag{3}$$

where

$$\sigma^2(\boldsymbol{\theta}_0) = \mathbf{g}^{\mathrm{T}}(\boldsymbol{\theta}_0) V(\boldsymbol{\theta}_0) \mathbf{g}(\boldsymbol{\theta}_0) \tag{4}$$

and

$$\mathbf{g}(\boldsymbol{\theta}_0) = \frac{\partial \eta(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}_0}. \tag{5}$$

An approximation for the distribution of $\hat{\eta}$ is given by (3) with $\hat{\boldsymbol{\theta}}$ used in place of the unknown $\boldsymbol{\theta}_0$.

The asymptotic results above are firmly established and much used in practice. For this reason parametric bootstrapping is not especially needed when parametric models are fitted to large samples. However, as will be discussed, there are certain situations where parametric bootstrapping is competitive. For example, it provides a neat way of obtaining the distribution of a goodness-of-fit statistic in a parametric setting where this distribution is otherwise difficult to obtain.

In practice, when the sample size is small, Monte Carlo methods are often used in parametric problems to examine the behavior of estimators and fitted distributions. Such methods are actually parametric bootstrapping under a different name and as such have been known and used for a long time. In this sense parametric bootstrapping significantly predates the direct nonparametric bootstrap. Parametric bootstrapping lacks the computational novelty of direct sampling in that resampling is not involved. Perhaps it is because of this that the significance of parametric bootstrapping was not fully appreciated, and its theoretical importance not recognized until the landmark paper of Efron (1979).

If we view bootstrapping as a numerical sampling approach involving models that may include parameters, then its potential is significantly extended. We shall therefore consider its use in more complex situations, where different kinds of parameters, including Bayesian ones are involved.

3 Quantiles and confidence intervals

3.1 Quantiles

In this section we discuss quantile estimation and the closely related problem of calculating confidence intervals. (We shall refer to *quantiles* rather than *percentiles* throughout, but many authors, including two of our main references Hjorth (1994) and Davison and Hinkley (1997), favor the term percentiles. The only real import is that we shall for consistency use the term quantile in the *names* of certain methods. Thus, for example, the reader should be aware that the *bootstrap quantile method* is precisely the same as the *bootstrap percentile method* referred to in other texts.)

We shall denote the estimate of a quantile q_p, with quantile probability p, by $\hat{q}_p(\mathbf{y})$.

The obvious choice for estimating a quantile nonparametrically is to use the corresponding quantile of the EDF. For a discrete distribution no special adjustment is needed as the EDF is already a step function. However for a continuous distribution, use of the EDF would mean that all quantiles in the interval $(\frac{k-1}{n}, \frac{k}{n})$ are estimated by the same observed-order statistic, $y_{(k)}$. Conversely the quantile $\frac{k}{n}$ is estimated by any point y for which $F_n(y) = k/n$. Such ambiguities are removed if the EDF is smoothed. One way is to use a kernel estimator of the density but in most simulations this is perhaps unnecessarily elaborate.

The simplest smoothing procedure is to note that if $Y_{(k)}$ is the kth-order statistic (as opposed to the observed value $y_{(k)}$) then $\mathrm{E}[F(Y_{(k)})] = k/(n+1)$ and to use this value to estimate the value of F at the observed order statistic $y_{(k)}$. We can now simply interpolate between these estimated points $(y_{(k)}, k/(n+1))$ for $k = 1, 2, \ldots, n$. The range can be extended to $(0, 0)$ (using the line segment joining $(0, 0)$ and $(y_{(1)}, 1/(n+1))$ if Y is known to be a positive random variable. If we denote this smoothed version of $F_n(\cdot)$ by $\widetilde{F}_n(\cdot)$ then an obvious estimator for q_p is

$$\hat{q}_p(\mathbf{y}) = \widetilde{F}_n^{-1}(p).$$

Estimating $F(\cdot)$ in the range $(y_{(n)}, \infty)$, or equivalently, q_p for $p > n/(n+1)$ is not advisable unless the tail behavior of F is known.

The bootstrap analogue of $\hat{q}_p(\mathbf{y})$ is obtained in the usual way. We draw $\mathbf{y}^*$, a bootstrap sample from $\mathbf{y}$, and construct $\widetilde{F}_n(\cdot|\mathbf{y}^*)$, the smoothed version of the bootstrap EDF. (Where there is no ambiguity we shall write $F_n^*(\cdot)$ for $F_n(\cdot|\mathbf{y}^*)$ and $\widetilde{F}_n^*(\cdot)$ for $\widetilde{F}_n(\cdot|\mathbf{y}^*)$.) We can now calculate $\hat{q}_p^*$ as $\hat{q}_p^* = \widetilde{F}_n^{*-1}(p)$. The bootstrap parameter here is $\eta^* = \hat{q}_p(\mathbf{y})$.

In the parametric case, things are much easier. Quantiles are simply estimated from the fitted distribution $F(\cdot, \hat{\boldsymbol{\theta}})$.

3.2 Confidence intervals by direct bootstrapping

We now consider the construction of confidence intervals for a parameter of interest, η, and consider to what extent the corresponding bootstrap process can be used to supply a confidence interval for this original parameter η. The key requirement is that the distribution of the bootstrap difference $\hat{\eta}^* - \eta^*$ should be close to that of $\hat{\eta} - \eta$. Roughly speaking we need

$$\mathrm{P}^*\big(\sqrt{n}\big(\hat{\eta}^* - \eta^*\big) \leqslant y\big) - \mathrm{P}\big(\sqrt{n}(\hat{\eta} - \eta) \leqslant y\big) \to 0 \tag{6}$$

for any y as $n \to \infty$. (As before with E^*, P^* denotes the conditional probability evaluation treating $F_n(\cdot\mathbf{y})$ as the parent distribution.) More precise statements of this convergence (6) and other results are given in Section 4.

We can proceed as follows. We generate B bootstrap samples $\mathbf{y}_j^*$, $j = 1, 2, \ldots, B$, and corresponding bootstrap estimates $\hat{\eta}_j^*$, $j = 1, 2, \ldots, B$. Then we select an appropriate confidence level $(1-2\alpha)$ (we use 2α rather than α so that α corresponds to each of the two tail probabilities in two-sided intervals) and find values a^* and b^* for which

$$\mathrm{P}^*\big(a^* \leqslant \hat{\eta}^* - \eta^* \leqslant b^*\big) \simeq 1 - 2\alpha.$$

We then appeal to (6) to replace $\hat{\eta}^* - \eta^*$ by $\hat{\eta} - \eta$ and obtain

$$\mathrm{P}\big(a^* \leqslant \hat{\eta} - \eta \leqslant b^*\big) \simeq 1 - 2\alpha$$

which on inversion gives the approximate $(1 - 2\alpha)$ confidence interval

$$\hat{\eta} - b^* \leqslant \eta \leqslant \hat{\eta} - a^* \tag{7}$$

for η.

One way of obtaining a^* and b^* is as follows. Let the bootstrap estimates of the α and $(1 - \alpha)$ quantiles obtained from the smoothed EDF of the $\hat{\eta}_j^*$'s be $\hat{q}_\alpha(\hat{\eta}^*)$ and $\hat{q}_{1-\alpha}(\hat{\eta}^*)$. These are estimates of the quantiles of the distribution of $\hat{\eta}^*$; so the corresponding quantiles of $\hat{\eta}^* - \eta^*$ are

$$a^* = \hat{q}_\alpha\big(\hat{\eta}^*\big) - \eta^* \quad \text{and} \quad b^* = \hat{q}_{1-\alpha}\big(\hat{\eta}^*\big) - \eta^*. \tag{8}$$

Substituting these into (7) and noting that $\eta^* = \hat{\eta}$ gives what is normally called *the basic bootstrap confidence interval*

$$2\hat{\eta} - \hat{q}_{1-\alpha}\big(\hat{\eta}^*\big) \leqslant \eta \leqslant 2\hat{\eta} - \hat{q}_\alpha\big(\hat{\eta}^*\big). \tag{9}$$

This confidence interval is also known as the *hybrid bootstrap confidence interval* as we have essentially replaced percentage points of the unknown original EDF by those of the smoothed bootstrap EDF. There is substantial evidence that this basic bootstrap can be significantly improved (in terms of giving coverage that is closer to the nominal level) if we use a *pivotal* quantity, like a Studentized mean, rather than focusing just on the difference $\hat{\eta} - \eta$. The reason is that a pivotal quantity is less dependent (ideally not dependent at all) on the form of the original distribution. This would mean that the confidence interval is less influenced by any difference there may be in the distributions of $\hat{\eta}^* - \eta^*$ and $\hat{\eta} - \eta$. We consider this next.

3.3 Studentization

Suppose that $\hat{\sigma}^2(\mathbf{y})$, an estimate of the variance of $\hat{\eta}$, can be calculated from the sample $\mathbf{y}$. This will be the case for $\hat{\eta} = \bar{y}$ say, when we have $\hat{\sigma}^2(\mathbf{y}) = s^2$, the sample variance.

As before suppose we can find a^* and b^* such that

$$\mathrm{P}^*\left(a^* \leqslant \frac{\hat{\eta}^* - \eta^*}{\hat{\sigma}^*} \leqslant b^*\right) \simeq 1 - 2\alpha.$$

If we then appeal to a similar result to (6) and replace $(\hat{\eta}^* - \eta^*)/\hat{\sigma}^*$ by $(\hat{\eta} - \eta)/\hat{\sigma}$, we obtain

$$\mathrm{P}\big(\hat{\sigma}a^* \leqslant \hat{\eta} - \eta \leqslant \hat{\sigma}b^*\big) \simeq 1 - 2\alpha$$

which on inversion gives the approximate $(1 - 2\alpha)$ confidence interval

$$\hat{\eta} - \hat{\sigma}b^* \leqslant \eta \leqslant \hat{\eta} - \hat{\sigma}a^*.$$

We can now calculate a^* and b^* by drawing B bootstrap versions of $z = (\hat{\eta} - \eta)/\hat{\sigma}$: $z_j^* = (\hat{\eta}_j^* - \eta^*)/\hat{\sigma}_j^*$, $j = 1, 2, \ldots, B$. Let $\hat{q}_\alpha(\mathbf{z}^*)$ and $\hat{q}_{1-\alpha}(\mathbf{z}^*)$ be the quantiles obtained from the EDF of these z_j^*. The confidence interval for η is now

$$\big(\hat{\eta}_\alpha^{Stud}, \hat{\eta}_{1-\alpha}^{Stud}\big) = \big(\hat{\eta} - \hat{\sigma}\hat{q}_{1-\alpha}(\mathbf{z}^*), \hat{\eta} - \hat{\sigma}\hat{q}_\alpha(\mathbf{z}^*)\big). \tag{10}$$

This is usually called the *Studentized bootstrap confidence interval*.

Studentized bootstrap intervals can be readily used with the parametric model $F(y, \boldsymbol{\theta})$ of Section 2.3. Suppose that $\mathbf{y}$ is a random sample of size n drawn from the distribution $F(y, \boldsymbol{\theta}_0)$, and that $\eta_0 = \eta(\boldsymbol{\theta}_0)$ is the quantity of interest. We can estimate η_0 using $\hat{\eta} = \eta(\hat{\boldsymbol{\theta}})$ where $\hat{\boldsymbol{\theta}}$ is the MLE of $\boldsymbol{\theta}_0$. When n is large we can use the asymptotic approximation

$$n^{1/2}\frac{\hat{\eta} - \eta_0}{\sigma}(\hat{\boldsymbol{\theta}}) \sim \mathrm{N}(0, 1).$$

When n is not large it is worth employing bootstrapping. We draw B bootstrap samples $\mathbf{y}_j^*$, $j = 1, 2, \ldots, B$, from the fitted distribution $F(y, \hat{\boldsymbol{\theta}})$. From each sample we obtain the bootstrap ML estimator $\hat{\boldsymbol{\theta}}_j^*$ and bootstrap Studentized quantity $z_j^* = n^{1/2}(\hat{\eta}_j^* - \eta^*)/\sigma(\hat{\boldsymbol{\theta}}_j^*)$. The quantiles $\hat{q}_\alpha(\mathbf{z}^*)$ and $\hat{q}_{1-\alpha}(\mathbf{z}^*)$ can be obtained from the EDF of these z_j^* and the Studentized interval (10) constructed.

In the nonparametric case an estimate of the variance of $\hat{\eta}$ may not be immediately available. One possibility is to use what is called the *double bootstrap*.

```
The Double Bootstrap
Given a random sample $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ from $F(\cdot)$,
Let $\hat{\eta} = s(\mathbf{y})$,
$\eta^* = \eta(F_n(\cdot|\mathbf{y}_j))$
Outer Bootstrap:
For $j = 1$ to $B$
    For $i = 1$ to $n$
        Draw $y_{ij}^*$ from $F_n(\cdot|\mathbf{y})$
    Next $i$
    Let $\hat{\eta}_j^* = s(\mathbf{y}_j^*)$, $\eta_j^{**} = \eta(F_n(\cdot|\mathbf{y}_j^*))$
    Inner Bootstrap:
    For $k = 1$ to $B$
        For $i = 1$ to $n$
            Draw $y_{ijk}^{**}$ from $F_n(\cdot|\mathbf{y}_j^*)$
        Next $i$
        Let $s_{jk}^{**} = s(\mathbf{y}_{jk}^{**})$
    Next $k$
    Let $\bar{s}_{j\cdot}^{**} = \frac{1}{B}\sum_{k=1}^{B} s_{jk}^{**}$
    Let $v_j^{**} = \frac{1}{B-1}\sum_{k=1}^{B}(s_{jk}^{**} - \bar{s}_{j\cdot}^{**})^2$
    Let $z_j^* = n^{1/2}(\hat{\eta}_j^* - \eta^*)/(v_j^{**})^{1/2}$
    End of Inner Loop
Next $j$
Form the EDF of the $z_j^*$, $j = 1, \ldots, B$
Obtain the quantiles $\hat{q}_\alpha(\mathbf{z}^*)$ and $\hat{q}_{1-\alpha}(\mathbf{z}^*)$
Form the Studentized confidence interval
$(\hat{\eta}_\alpha^{Stud}, \hat{\eta}_{1-\alpha}^{Stud}) = (\hat{\eta} - \hat{\sigma}\hat{q}_{1-\alpha}(\mathbf{z}^*), \hat{\eta} - \hat{\sigma}\hat{q}_\alpha(\mathbf{z}^*))$
End of Outer Bootstrap
```

Fig. 3. The double bootstrap.

This comprises an outer and inner bootstrap loop. The outer loop produces, for each j, a bootstrap sample $\mathbf{y}_j^*$ and the resulting bootstrap quantity of interest $\hat{\eta}_j^*$. In addition, for each j, there is an inner loop which produces a set of samples $\{\mathbf{y}_{jk}^{**}, k = 1, \ldots, B\}$ by bootstrap sampling from $\widetilde{F}_n(\cdot|\mathbf{y}_j^*)$ (the EDF of the bootstrap sample $\mathbf{y}_j^*$). This allows a set of (double) bootstrap quantities of interest to be produced: $\hat{\eta}_{jk}^{**}$, $k = 1, \ldots, B$. The sample variance, v_j^{**} say, of these $\hat{\eta}_{jk}^{**}$, then provides an estimate of the variance of $\hat{\eta}_j^*$ thereby allowing the bootstrap Studentized z_j^* to be calculated. Precise details of this double bootstrap calculation is given in Figure 3.

If the double bootstrap is considered computationally too expensive, then an alternative using *influence functions* can be used provided our statistic is expressible as a functional of the EDF, i.e., $s(\mathbf{y}) = \eta[F_n(\cdot|\mathbf{y})]$. Such statistics are termed *statistical functions*, and were introduced by von Mises (1947). We then assume that the relevant functional $\eta = \eta[F(\cdot)]$ has a linearized form (akin to a first-order linear generalization of a Taylor series): $\eta(G(\cdot)) \simeq \eta(F(\cdot)) +$

$\int L_\eta(y|F(\cdot))\,\mathrm{d}G(y)$. Here

$$L_\eta\big(y|F(\cdot)\big) = \frac{\partial\eta\{(1-\varepsilon)F(\cdot)+\varepsilon H_y(\cdot)\}}{\partial\varepsilon} = \lim_{\varepsilon\to 0}\varepsilon^{-1}\big[\eta\big\{(1-\varepsilon)F(\cdot)+\varepsilon H_y(\cdot)\big\}-\eta\big(F(\cdot)\big)\big], \tag{11}$$

the derivative of η at F, is called the *influence function* of η, and $H_y(x)$ is the Heaviside unit step function with jump from 0 to 1 at $x = y$. The sample approximation

$$l(y|\mathbf{y}) = L_\eta\big(y|F_n(\cdot|\mathbf{y})\big) \tag{12}$$

is called the *empirical influence function*. An analogous argument to that used in the delta method yields the nonparametric analogue of (4) as

$$\mathrm{Var}\big[\eta\big\{F(\cdot)\big\}\big] = n^{-1}\int L_\eta^2\big\{y|F(\cdot)\big\}\,\mathrm{d}F(y),$$

with sample version

$$\mathrm{Var}\big[\eta\big\{F_n(\cdot|\mathbf{y})\big\}\big] = n^{-2}\sum_{j=1}^{n} l^2(y_j|\mathbf{y}),$$

where the $l(y_j|\mathbf{y})$ are the *empirical function values* evaluated at the observed values y_j. In practice these values have to be evaluated numerically using (11) with, typically, $\varepsilon = (100n)^{-1}$.

The ease with which the empirical influence function can be evaluated is rather problem dependent, and the potential complexity of this calculation detracts from this approach. One simple approximation is the jacknife approximation

$$l_{jack}(y_j|\mathbf{y}) = (n-1)\big(\hat\eta(\mathbf{y}) - \hat\eta(\mathbf{y}_{\backslash j})\big),$$

where $\hat\eta(\mathbf{y}_{\backslash j})$ is the estimate calculated from the sample $\mathbf{y}$ but with the observation y_j omitted. (See Davison and Hinkley, 1997, problem 2.18.)

3.4 *Quantile methods*

Suppose now that the distribution of $\hat\eta$ is symmetric, or more generally, there is some transformation $w = h(\hat\eta)$ for which the distribution is symmetric. Examination of the form of the bootstrap quantiles (8) (see, for example, Hjorth, 1994, Section 6.6.2) shows that they do not depend on the explicit form of $h(\cdot)$ at all. Instead we find that they can, with a change of sign, be swapped, i.e. a^* is replaced by $-b^*$ and b^* by $-a^*$. Then (7) becomes

$$\big(\hat\eta_\alpha^{\mathrm{Q}}, \hat\eta_{1-\alpha}^{\mathrm{Q}}\big) = \big(\hat q_\alpha(\hat\eta^*), \hat q_{1-\alpha}(\hat\eta^*)\big). \tag{13}$$

This $(1-2\alpha)$ confidence interval is known as the *bootstrap quantile interval*. This confidence interval is easy to implement but can be improved. We give two variants called the *bias corrected* (BC) and *accelerated bias corrected* (BCa) *quantile methods*.

The basic idea in the BCa method is that there is a Studentized transformation g for which

$$\frac{g(\hat{\eta}) - g(\eta)}{1 + ag(\eta)} + b \sim \mathrm{N}(0, 1). \tag{14}$$

The BC method is simply the special case where $a = 0$.

If we calculate a one-sided confidence interval for η under the assumption (14) we find that if we set

$$\beta_1 = \Phi\left(b + \frac{b + z_\alpha}{1 - a(b + z_\alpha)}\right),$$

where $\Phi(\cdot)$ is the standard normal distribution function, then with confidence $1 - \alpha$

$$\hat{q}_{\beta_1}(\hat{\eta}^*) < \eta,$$

where $\hat{q}_\beta(\hat{\eta}^*)$ is the bootstrap quantile of the $\hat{\eta}^*$ with probability β. In a similar way a two-sided $(1 - 2\alpha)$ confidence interval is

$$\hat{q}_{\beta_1}(\hat{\eta}^*) < \eta < \hat{q}_{\beta_2}(\hat{\eta}^*),$$

where

$$\beta_2 = \Phi\left(b + \frac{b + z_{1-\alpha}}{1 - a(b + z_{1-\alpha})}\right).$$

The bias parameter b is obtained as

$$b = \Phi^{-1}\{\widetilde{F}_n(\hat{\eta}|\hat{\eta}^*)\}$$

where $\widetilde{F}_n(\hat{\eta}|\hat{\eta}^*)$ is the smoothed EDF of the bootstrap sample $\hat{\eta}^*$ evaluated at the original estimated value $\hat{\eta}$. If we now set $a = 0$ this gives the BC method. We denote the resulting two-sided version of the confidence interval by $(\hat{\eta}_\alpha^{\mathrm{BC}}, \hat{\eta}_{1-\alpha}^{\mathrm{BC}})$.

The acceleration parameter a is arguably less easily calculated. Efron (1987) suggests the approximation (which is one-sixth the estimated standard skewness of the linear approximation to η)

$$a = \frac{1}{6}\frac{\sum_{i=1}^{n} l^3(y_i|\mathbf{y})}{[\sum_{i=1}^{n} l^2(y_i|\mathbf{y})]^{3/2}},$$

where $l(y_i|\mathbf{y})$ are the values of the empirical influence function (12) evaluated at the observations y_i. We denote the two-sided version of the confidence interval given by this BCa method by $(\hat{\eta}_\alpha^{\mathrm{BCa}}, \hat{\eta}_{1-\alpha}^{\mathrm{BCa}})$.

3.5 Comparison of methods

We have considered four types of confidence intervals: the *basic bootstrap method*, the *Studentized bootstrap method*, the *quantile method* and the *accelerated bias corrected method* (and its variant, the *bias corrected method*). The obvious question is when should each be used? Davison and Hinkley (1997) give several empirical examples which indicate that the basic and quantile methods can perform quite poorly especially when sample size is small ($n < 30$, say). In contrast the Studentized and accelerated bias corrected methods appear quite robust, giving coverages close to the nominal value. If an appropriate transformation of the observations can be applied, to make them more normally distributed, then this does improve the performance of the basic and quantile methods, but tends to improve the performance of the Studentized and accelerated methods even more. Hjorth (1994), Example 6.4, gives an interesting numerical example that corroborates these findings. The accelerated bias corrected method performed best, followed by the Studentized. The percentile came next, with the basic method faring noticeably the worst.

This empirical evidence is borne out by theory which is considered in the next section.

4 Theory

Like many statistical methods, understanding of the practical usefulness of the bootstrap as well as its limitations has been built up gradually with experience through applications. This practical experience has been underpinned by a growing asymptotic theory which provides a basis for understanding when bootstrapping will work and when it will not. A truly general theory rapidly becomes very technical and is still incomplete. We shall not attempt a detailed treatment but will summarize some of the more accessible and useful results. A weakness of the theory is that it is asymptotic, and so is not directly applicable to the small sample situation of real interest. Nevertheless, though indirect, asymptotic convergence properties provide some feel for when a method might perform well in small samples.

Bootstrapping relies on sampling from the EDF $F_n(\cdot|\mathbf{y})$ reproducing the behavior of sampling from the original distribution $F(\cdot)$. We therefore need convergence of $F_n(\cdot|\mathbf{y})$ to $F(\cdot)$. This is confirmed by the Glivenko–Cantelli theorem which guarantees strong convergence, i.e.,

$$\sup_y \left|F_n(y|\mathbf{y}) - F(y)\right| \to 0 \quad \text{with probability 1}$$

as $n \to \infty$. Though reassuring this does not throw any direct light on the bootstrap process.

To investigate the bootstrap process we look instead at the equally well-known result that the process

$$\{Z_n(y) = \sqrt{n}\big(F_n(y) - F(y)\big)\colon\ y \in \mathbb{R}\}$$

converges in probability to the Brownian bridge $\{B(F(y))\colon\ y \in \mathbb{R}\}$ as $n \to \infty$. (A Brownian bridge $B(t)$, $0 < t < 1$, is a Gaussian process for which $B(0) = B(1) = 0$, $\mathrm{E}[B(u)] = 0$ and $\mathrm{E}[B(u)B(v)] = u(1-v)$ for $0 < u < v < 1$.) Bickel and Freedman (1981) give the following bootstrap version of this.

Theorem. *Let $Y_1, Y_2, \ldots$ be a sequence of independent observations from $F(\cdot)$, with $\mathbf{Y}_n$ comprising the first n values. Then for almost all such sequences* (*i.e.*, *with probability* 1) *the bootstrap process*

$$\{Z_n^*(y) = \sqrt{n}\big(F_n^*(y|\mathbf{Y}_n) - F_n(y)\big)\colon\ y \in \mathbb{R}\}$$

converges in probability to the Brownian bridge $\{B(F(y))\colon\ y \in \mathbb{R}\}$ as $n \to \infty$.

An interesting immediate application of this result to the calculation of confidence bands for $F(y)$ is given by Bickel and Freedman. We set a confidence level $1 - \alpha$ and, from the bootstrap process, we then select $c_n(\alpha)$ so that, as $n \to \infty$,

$$\mathrm{P}^*\Big\{\sup_y \big|Z_n^*(y)\big| \leqslant c_n(\alpha)\Big\} \to 1 - \alpha.$$

Then, as $Z_n^*(y)$ and $Z_n(y)$ converge to the same process $B(F(y))$, we have also that

$$\mathrm{P}^*\Big\{\sup_y \big|Z_n(y)\big| \leqslant c_n(\alpha)\Big\} \to 1 - \alpha.$$

Thus an asymptotically correct $(1 - \alpha)$ confidence band for $F(y)$ is

$$F_n(y) \pm \frac{c_n(\alpha)}{\sqrt{n}}.$$

We now consider the distribution of an estimator $\hat{\eta}$ of a parameter η. We need the distributions of the bootstrap estimator $\hat{\eta}^*$ and of the difference $\hat{\eta}^* - \eta^*$ to reproduce respectively the distributions of $\hat{\eta}$ ($\equiv \eta^*$) and of $\hat{\eta} - \eta$. The most general results obtained to date assume that the statistic can be expressed as $s = \eta(F_n)$ where both the EDF F_n and the underlying distribution F belong to a space $\mathcal{F}$ of distributions. Moreover such results require s to be differentiable in such a way that a linear approximation can be formed with, as first-order terms, the previously described influence functions, i.e.,

$$\eta(F_n) = \eta(F) + n^{-1}\sum_{j=1}^{n} l(y_j|\mathbf{y}) + \mathrm{o}_p\big(n^{-1/2}\big).$$

Then it is often possible to establish the analogous result for the bootstrap version

$$\eta(F_n^*) = \eta(F) + n^{-1}\sum_{j=1}^{n} l(y_j^*|\mathbf{y}^*) + \mathrm{o}_p(n^{-1/2}),$$

and, provided $\mathrm{Var}[\eta(F)]$ is finite, then in probability, the sequence $y_1, y_2, \ldots$ is such that

$$\sup_z \bigl|\mathrm{P}^*\bigl(\eta(F_n^*) \leqslant z\bigr) - \mathrm{P}\bigl(\eta(F_n) \leqslant z\bigr)\bigr| \to 0$$

as $n \to \infty$. Here we have written $\mathrm{P}^*(\eta(F_n^*) \leqslant z)$ for $P(\eta(F_n^*) \leqslant z|F_n)$. If, in addition, s is *continuously* differentiable, then we can usually strengthen the result to almost sure convergence.

A detailed discussion is given by Shao and Tu (1995) who give examples of $\hat{\eta}(F)$ that satisfy such conditions. Convergence has been studied in detail for the particular case of means and quantiles. In this case a more accessible approach using the Berry–Esséen inequality is possible, and this we discuss in the next section.

4.1 Convergence rates

Efron (1979) considered the finite case where Y takes just a finite set of values which we can take to be $\{1, 2, \ldots, m\}$ with probabilities $\mathbf{p} = (p_1, p_2, \ldots, p_m)^{\mathrm{T}}$. We shall not discuss this particular case, a good summary for which is provided by Hjorth (1994).

We shall instead focus on the continuous case where $\eta = \mu \equiv \mathrm{E}(Y)$, with $\hat{\eta} = \overline{Y}_n = \sum_{i=1}^{n} Y_i/n$. We also write $\mathrm{Var}(Y) = \sigma^2$, $\rho = \mathrm{E}(|Y|^3)$. The well-known Berry–Esséen theorem (see Serfling, 1980, for example) then provides a powerful result for investigating bootstrap convergence. To avoid long formulas we shall write

$$H_n(y) = \mathrm{P}\bigl(\sqrt{n}(\overline{Y}_n - \mu) \leqslant y\bigr)$$

for the (unknown) CDF of interest and

$$H_n^*(y) = \mathrm{P}^*\bigl(\sqrt{n}\bigl(\overline{Y}_n^* - \overline{Y}_n\bigr) \leqslant y\bigr)$$

for the bootstrap version of this. We write

$$\widetilde{H}_n(y) = \mathrm{P}\biggl(\sqrt{n}\frac{\overline{Y}_n - \mu}{\sigma} \leqslant y\biggr)$$

for the (unknown) *standardized* CDF of interest,

$$\breve{H}_n(y) = \mathrm{P}\biggl(\sqrt{n}\frac{\overline{Y}_n - \mu}{\hat{\sigma}_n} \leqslant y\biggr)$$

for the Studentized CDF, and

$$\breve{H}_n^*(y) = \mathrm{P}^*\left(\sqrt{n}\frac{\overline{Y}_n^* - \overline{Y}_n}{\hat{\sigma}_n^*} \leqslant y\right)$$

for its bootstrap *Studentized* version. We write also $\Phi(y)$ for the CDF of the standard normal distribution.

The Berry–Esséen theorem states that if $Y_1, Y_2, \ldots$ are independent identically distributed random variables and $\rho = \mathrm{E}[|Y_i^3|] < \infty$ then for all n,

$$\sup_y \left| H_n(y) - \Phi\left(\frac{y}{\sigma}\right)\right| < K\frac{\rho}{\sigma^3\sqrt{n}}.$$

The value of the constant given by Berry and Esséen, $K = \frac{33}{4}$, has been improved and reduced to 0.7975 (van Beeck, 1972). Thus the theorem gives a bound on the effectiveness of the normal distribution in approximating $H_n(\cdot)$.

A typical application of the theorem is in proving the following one.

Theorem. *If* $\rho = \mathrm{E}[|Y_i^3|] < \infty$ *then, as* $n \to \infty$,

$$\sup_y \left| H_n^*(y) - H_n(y)\right| \to 0 \tag{15}$$

for almost all sequences $y_1, y_2, \ldots$ *of independent observations drawn from* $F(\cdot)$.

Proof outline. Write $\hat{\sigma}_n = \sqrt{n^{-1}\sum_{i=1}^n (Y_i - \overline{Y}_n)^2}$, $\hat{\rho}_n = n^{-1}\sum_{i=1}^n |Y_i - \overline{Y}_n|^3$. By the Berry–Esséen theorem we have

$$\left| H_n(y) - \Phi\left(\frac{y}{\sigma}\right)\right| < K\frac{\rho}{\sigma^3\sqrt{n}}$$

and

$$\left| H_n^*(y) - \Phi\left(\frac{y}{\hat{\sigma}_n}\right)\right| < K\frac{\hat{\rho}_n}{\hat{\sigma}_n^3\sqrt{n}}.$$

By the strong law of large numbers $\mu^* \equiv \bar{y}_n \to \mu$, $\hat{\sigma}_n \to \sigma$, $\hat{\rho}_n \to \rho$ all with probability 1. This shows that the two probabilities in (15) converge to the same normal distribution from which the result follows easily. □

In fact the result above does not depend on $\rho < \infty$; the condition $\sigma^2 < \infty$ is actually enough (see Bickel and Freedman (1981) or Singh (1981)). However, the condition $\rho < \infty$ and explicit use of the Berry–Esséen theorem is needed to establish the *rate* of convergence more precisely. In the remainder of this section we summarize results obtained by Singh (1981) and Bickel and Freedman (1981).

We need one further definition. A random variable has distribution $F(\cdot)$ that is *lattice* if there is zero probability of it taking values outside the discrete points $y_j = c \pm jd$, $j = 0, 1, 2, \ldots,$ where c and d are constants.

Theorem. *For almost all sequences $Y_1, Y_2, \ldots$ of independent observations drawn from $F(\cdot)$:*

(i) *If* $\mathrm{E}(Y^4) < \infty$, *then*

$$\limsup_{n\to\infty}\left[\frac{\sqrt{n}}{\sqrt{\log\log n}}\sup_y\left|H_n^*(y) - H_n(y)\right|\right] = \frac{\sqrt{\mathrm{Var}[(Y-\mu)^2]}}{2\sigma^2\sqrt{\pi e}}.$$

(ii) *If* $\mathrm{E}[|Y|^3] < \infty$ *and* $F(\cdot)$ *is lattice then*

$$\limsup_{n\to\infty}\left[\sqrt{n}\sup_y\left|\widetilde{H}_n^*(y) - \widetilde{H}_n(y)\right|\right] = \frac{d}{\sqrt{2\pi}\sigma}.$$

(iii) *If* $\mathrm{E}[|Y|^3] < \infty$ *and* $F(\cdot)$ *is nonlattice then*

$$\sqrt{n}\sup_y\left|\widetilde{H}_n^*(y) - \widetilde{H}_n(y)\right| \to 0.$$

The result (i) shows that, when $\mathrm{E}(Y^4) < \infty$, the convergence rate of H_n^* is $\mathrm{O}(\sqrt{\log\log n/n})$. This is actually the same as that for approximating $\sqrt{n}(\overline{Y}_n - \mu)/\hat{\sigma}_n$, the *Studentized* original $\overline{Y}$ by Z, a standard normal, i.e., approximating $H_n(y)$ by $\Phi(y/\hat{\sigma}_n)$. So in this case we do no better using the bootstrap than standard normal theory.

Results (ii) and (iii) show that when $\mathrm{E}[|Y|^3] < \infty$, the convergence rate of $\widetilde{H}_n^*(\cdot)$ to $\widetilde{H}_n(\cdot)$ is at least $\mathrm{O}(n^{-1/2})$ and is $\mathrm{o}(n^{-1/2})$ when F is nonlattice. Thus bootstrapping does better than the usual normal approximation in this case. *This is the key theoretical result that underpins much of bootstrapping.*

We consider now quantile estimation. Perhaps the most informative general result is that due to Singh (1981). Let $\eta = q(p) = F^{-1}(p)$; where the conditions of the theorem ensure that $q(p)$ is uniquely defined. We can estimate η from the EDF using $\hat{\eta} = F_n^{-1}(p) = \sup\{y\colon F_n(y) \leqslant p\}$ and define $\hat{\eta}^* = F_n^{*-1}(p)$ from $\mathbf{y}^*$ as usual. We take $H_n = \mathrm{P}(\sqrt{n}(\hat{\eta} - \eta) \leqslant y)$ and $H_n^* = \mathrm{P}^*(\sqrt{n}(\hat{\eta}^* - \hat{\eta}) \leqslant y)$.

Theorem. *If F has a bounded second derivative in a neighborhood of η and $f(\eta) = \frac{\mathrm{d}F}{\mathrm{d}y}|_\eta > 0$, then*

$$\limsup_{n\to\infty}\frac{n^{1/4}\sup_y|H_n^*(y) - H_n(y)|}{\sqrt{\log\log n}} = c_{p,F} \quad \textit{with probability } 1,$$

where $c_{p,F}$ is a constant depending on p and F only.

This result shows that $H_n^*(y)$ converges at rate $\mathrm{O}(n^{-1/4}\sqrt{\log\log n})$ to $H_n(y)$. Now for a quantile estimator we have from the Berry–Esséen theorem

$$\sup_y\left|H_n(y) - \Phi\left(\frac{y}{\tau}\right)\right| = \mathrm{O}(n^{-1/2}),$$

where $\tau = \sqrt{p(1-p)}/f(\eta)$. Thus if we were to use a normal approximation for $H_n(y)$ we would use $\Phi(y/\hat{\tau})$ where $\hat{\tau}$ is an estimate of τ. Whether this is better than $H_n^*(y)$ thus depends on the convergence rate of $\hat{\tau}$, and the matter is not a clear one.

4.2 Asymptotic accuracy of EDFs

Edgeworth expansions are asymptotic series aimed at improving the normal approximation by introducing additional terms that try to correct for effects of skewness, kurtosis and higher moments which slow the rate of convergence to normality.

For fixed n, asymptotic series usually diverge as more and more terms are included. However for a fixed number of terms, k say, the series converges as $n \to \infty$. The rate of convergence is usually of smaller order than the last included term. We shall only consider the special case $\hat{\eta} = \overline{Y}_n$. Here, the general Edgeworth expansion is a power series in $n^{-1/2}$ and has the form

$$\begin{aligned}\widetilde{H}_n(y) = \Phi(y) + n^{-1/2}p^{(1)}(y)\phi(y) + \cdots + n^{-k/2}p^{(k)}(y)\phi(y)\\ + \mathrm{o}(n^{-k/2}),\end{aligned} \tag{16}$$

where $\phi(y) = (2\pi)^{-1/2}\exp(-y^2/2)$ is the standard normal density, and $p^{(j)}$ is a polynomial of degree $3j-1$. We have explicitly

$$p^{(1)}(y) = -\frac{1}{6}\kappa_3(y^2-1)$$

and

$$p^{(2)}(y) = -\left\{\frac{1}{24}\kappa_4(y^2-3) + \frac{1}{72}\kappa_3^2(y^4-10y^2+15)\right\},$$

where κ_3 and κ_4 are the skewness and kurtosis of $F(\cdot)$. The term involving $p^{(1)}$ corrects for the main effect of skewness, the term involving $p^{(2)}$ corrects for the main effect of kurtosis and for the secondary effect of skewness.

Often the remainder term $\mathrm{o}(n^{-k/2})$ can be replaced by $\mathrm{O}(n^{-(k+1)/2})$ when the Edgeworth expansion is said to be $(k+1)$st order accurate. Usually inclusion of more than one or two correction terms becomes counter-productive as the coefficients associated with the powers of $n^{-1/2}$ rapidly become large with k.

When $\mathrm{E}(|Y|^3) < \infty$ and Y is nonlattice, one-term Edgeworth expansions for both $H_n(y)$ and $H_n^*(y)$ exist and a comparison (see Shao and Tu, 1995, Section 3.3.3) shows that $H_n^*(y)$ has smaller asymptotic mean square error than $\Phi(y/\hat{\sigma}_n)$ unless the skewness is zero. Comparison of $H_n^*(y)$ and the one-term Edgeworth expansion estimator

$$H_n^{\mathrm{EDG}}(y) = \Phi(z) + n^{-1/2}p_n^{(1)}(z|\mathbf{y}_n)\phi(z),$$

where $z = \hat{\sigma}_n^{-1} y$, and $p_n^{(1)}(z|\mathbf{y}_n)$ is the polynomial $p^{(1)}(z)$ with the moments of $F(\cdot)$ replaced by the sample moments calculated from $\mathbf{y}_n$, shows that both have the same asymptotic mean square error.

For Studentized versions the bootstrap does even better. Under appropriate moment conditions both $\breve{H}_n(y)$ and $\widetilde{H}_n^*(y)$ have two-term Edgeworth expansions and a comparison of these shows that $\widetilde{H}_n^*(y)$ has a smaller asymptotic mean square error than the corresponding one-term Edgeworth estimator, though they have the same convergence rate.

The normal, bootstrap and one-term Edgeworth expansion estimators of the standardized distribution have been compared by Hall (1988) using the criterion of asymptotic relative error.

When $\mathrm{E}(|Y|^3) < \infty$ then the bootstrap does better than the normal approximation, but it may or may not do better than the one term Edgeworth expansion (see Shao and Tu, 1995).

When $\mathrm{E}(|Y|^3) = \infty$, the situation is more complicated and depends on the tail behavior of $F(\cdot)$. When the tail is thin the bootstrap can be worse than the normal approximation.

In estimating tail behavior the bootstrap is comparable to the one term Edgeworth expansion except in the extreme of the tail.

4.3 Asymptotic accuracy of confidence intervals

The analysis above focuses on distribution functions, and does not give the whole picture. It is helpful to consider also the coverage accuracy of confidence intervals. We shall write the basic confidence limit that we seek as $\hat{\eta}_\alpha$ defined by

$$\Pr(\eta \leqslant \hat{\eta}_\alpha) = \alpha$$

and the normal, basic, quantile, Studentized bootstrap, BC and BCa approximations as $\hat{\eta}_\alpha^{Norm}$, $\hat{\eta}_\alpha^{Boot}$, $\hat{\eta}_{\alpha,}^{\mathrm{Q}}$ $\hat{\eta}_\alpha^{Stud}$, $\hat{\eta}_\alpha^{\mathrm{BC}}$ and $\hat{\eta}_\alpha^{\mathrm{BCa}}$, respectively. We summarize the results given in Shao and Tu (1995). These apply when the parameter of interest is a smooth function of the mean, $\eta = \eta(\bar{y}_n)$. Then an analysis analogous to that used for EDF's can be carried out, but now relying on an expansion of the quantile, that is the inverse of Edgeworth series, called the *Cornish–Fisher expansion*.

Let $\Phi(z_\alpha) = \alpha$. Under appropriate moment conditions q_α has an expansion of the form

$$q_\alpha = z_\alpha + n^{-1/2} q^{(1)}(z_\alpha) + n^{-1} q^{(2)}(z_\alpha) + \mathrm{o}(n^{-1}),$$

where comparison with the Edgeworth expansion shows that

$$q^{(1)}(y) = -p^{(1)}(y)$$

and

$$q^{(2)}(y) = p^{(1)}(y) p^{(1)\prime}(y) - \frac{1}{2} p^{(1)}(y)^2 - p^{(2)}(y).$$

Under appropriate conditions we find that analogous expansions exist for the quantile approximations listed above. We find in particular that

$$\hat{\eta}_\alpha^{Boot} - \hat{\eta}_\alpha = \mathrm{O}_p(n^{-1})$$

and in general

$$\Pr(\hat{\eta}_\alpha^{Boot} \leqslant \eta) = 1 - \alpha + \mathrm{O}(n^{-1/2}).$$

However, the two tailed version is second-order accurate:

$$\Pr(\hat{\eta}_\alpha^{Boot} \leqslant \eta \leqslant \hat{\eta}_{1-\alpha}^{Boot}) = 1 - 2\alpha + \mathrm{O}(n^{-1}).$$

For symmetric distributions, these results apply to the quantile approximation as well, for example:

$$\hat{\eta}_\alpha^{Q} - \hat{\eta}_\alpha = \mathrm{O}_p(n^{-1}).$$

The bootstrap BC method turns out to perform no better than the quantile limit, in terms of convergence rate, but the constant factor is smaller so it is marginally to be preferred.

Studentization is definitely better with

$$\hat{\eta}_\alpha^{Stud} - \hat{\eta}_\alpha = \mathrm{O}_p(n^{-3/2}) \quad \text{and} \quad \hat{\eta}_\alpha^{\mathrm{BCa}} - \hat{\eta}_\alpha = \mathrm{O}_p(n^{-3/2})$$

and

$$\Pr(\hat{\eta}_\alpha^{Stud} \leqslant \eta) = 1 - \alpha + \mathrm{O}(n^{-1}) \quad \text{and}$$
$$\Pr(\hat{\eta}_\alpha^{\mathrm{BCa}} \leqslant \eta) = 1 - \alpha + \mathrm{O}(n^{-1}).$$

It follows that the two-sided intervals for both limits are also both second-order accurate.

Finally we note that

$$\hat{\eta}_\alpha^{Boot} - \hat{\eta}_\alpha^{Norm} = \mathrm{O}_p(n^{-1})$$

so that the usual normal approximation and the basic bootstrap behave similarly.

4.4 Failure of bootstrapping

It should be clear from the previous three subsections that, for bootstrapping to work well, regularity conditions are required on both the distribution F and also on the statistic of interest.

More explicitly, bootstrapping is sensitive to the tail behavior of F; convergence of H_n^* usually requires moment conditions on F that are more stringent than those needed for convergence of H_n.

Also the statistic $s(\mathbf{Y})$ has to be suitably smooth in an appropriate sense.

Finally it is possible for convergence of the bootstrap to be sensitive to the method used in carrying out the bootstrapping.

An example of the first situation is inconsistency of $s(\mathbf{Y})$, when it is simply the variance, even when the asymptotic variance is finite (see Ghosh et al., 1984). This can occur if $F(\cdot)$ is fat-tailed and does not have appropriate moments. The problem then arises because the bootstrap $s(\mathbf{Y}^*)$ can take exceptionally large values.

An example of the second situation is where $\mathbf{y}$ is a random sample, from $\mathrm{U}(0, b)$ say, and we wish to consider $y_{(n)}$, the largest-order statistic. Then a natural statistic to consider is $s(\mathbf{y}) = n(b - y_{(n)})/b$, which has a limiting standard exponential distribution as $n \to \infty$. The bootstrap version is then $s(\mathbf{y}^*|\mathbf{y}) = n(y_{(n)} - y^*_{(n)})/y_{(n)}$. But

$$\mathrm{P}^*\bigl(s(\mathbf{y}^*|\mathbf{y}) = 0\bigr) = \mathrm{P}^*\bigl(y^*_{(n)} = y_{(n)}\bigr) = 1 - \bigl(1 - n^{-1}\bigr)^n \to 1 - \mathrm{e}^{-1}.$$

Thus H^*_n does not tend to H_n as $n \to \infty$. This result applies to any given order statistic $y_{(n-k)}$ where k is fixed. However the problem does not arise for a given quantile $y_{(pn)}$ where p is fixed with $0 < p < 1$.

We shall see an example of the last situation, where convergence depends not on the problem but on the bootstrapping method employed, when we consider model selection in Section 5.6.

5 Simulation models

5.1 Direct models

Simulation models are *direct models* in the sense that they attempt to mimic the behavior of an actual system by simulating the different objects of a system and the (typically) dynamic relationships between them. For example in a single server queue we try to capture the actual arrival patterns of customers and how they are then processed by the server. The quantities that we try to analyze are summary characteristics such as average queue length and waiting times. We can therefore think of a set of n simulation runs as yielding observations

$$y_j = y\bigl(\mathbf{u}_j, \mathbf{v}_j, \hat{\boldsymbol{\theta}}(\mathbf{w}), \mathbf{x}_j\bigr), \quad j = 1, 2, \ldots, n, \tag{17}$$

where the y_j depend on a number of quantities that we now explain.

Firstly $\mathbf{u}_j$ denotes the stream of uniformly distributed $\mathrm{U}(0, 1)$ random numbers used in the jth run. Typically the uniforms are not used directly, but are transformed into random variables drawn from distributions other than the uniform.

Next comes $\mathbf{v}_j$. This represents a sequence of inputs that are random, but that has been generated independently of the simulation model. A typical instance is where $\mathbf{v}_j$ comprises sampled real observations taken from some real system, separate from the system being modeled, but on which y_j depends. Such a sampled real process is sometimes called a *trace*. We shall simply think of $\mathbf{v}_j$ as being just a sample of observations.

In addition there may be further quantities which may affect y. These are denoted by $\mathbf{x}$ and $\boldsymbol{\theta}$. There is no essential difference between $\mathbf{x}$ and $\boldsymbol{\theta}$ in the way that they influence y. They are simply variables on which y depends. However we make a distinction in supposing that $\mathbf{x}$ are *decision variables*, i.e., quantities that can be selected by the simulator. However we also include in $\mathbf{x}$ parameters whose values are known, and that might affect y, but which are not selectable by the simulator. The reason for adopting this convention is that it then allows us to assume $\boldsymbol{\theta}$ to be those parameters whose values are not known, and so have to be estimated. We will therefore assume that in addition to $\mathbf{v}$, there exists input data $\mathbf{w}$ that is used exclusively to estimate $\boldsymbol{\theta}$ and it is the estimated values, $\hat{\boldsymbol{\theta}}(\mathbf{w})$, that are used in the simulation runs.

A simple example is a simulation model of a multiserver queue, where y is the observed average queue length over a given period, $\mathbf{v}$ might be a previously observed set of interarrival times, $\mathbf{x}$ might be the (scalar) number of servers and $\boldsymbol{\theta}$ the service rates of servers. Here we treat $\mathbf{x}$ as being selectable by the simulator so that it is a design variable, and $\boldsymbol{\theta}$ may not be known and have to be estimated from available sampled real service times $\mathbf{w}$.

Resampling might already feature in carrying out the runs yielding the y_j of (17). For example variability in the $\mathbf{v}$ will contribute to the variability of y. We can therefore automatically build this effect into the simulation runs simply by using resampling from an initial given sequence $\mathbf{v}$ to construct resampled sequences $\mathbf{v}_j^*$ for use in the actual runs.

Similarly in the case (17), because $\hat{\boldsymbol{\theta}}(\mathbf{w})$ is estimated, we can allow for its randomness affecting the distribution of y by using *independent values* of $\boldsymbol{\theta}_j$ in each simulation run. When $\hat{\boldsymbol{\theta}}(\mathbf{w})$ is the ML estimate we can draw sample values of $\boldsymbol{\theta}$

$$\boldsymbol{\theta}_j^* \sim \mathrm{N}\bigl(\hat{\boldsymbol{\theta}}, I^{-1}(\hat{\boldsymbol{\theta}})\bigr) \tag{18}$$

from the asymptotic normal distribution (1) with $I(\hat{\boldsymbol{\theta}})$ calculated from (2). An alternative is to produce bootstrap samples $\mathbf{w}_j^*$ from $\mathbf{w}$, and to use the bootstrap estimate

$$\boldsymbol{\theta}_j^* = \hat{\boldsymbol{\theta}}\bigl(\mathbf{w}_j^*\bigr) \tag{19}$$

for $\boldsymbol{\theta}$ in the jth run.

5.2 Metamodels

We now consider a very different approach where the behavior of a system is represented by a *metamodel*. This is simply where we regard the behavior of y as representable by a *statistical regression model*. The runs are then not represented as (17) but instead as

$$y_j = \eta(\mathbf{x}_j; \boldsymbol{\beta}) + \varepsilon_j, \quad j = 1, 2, \ldots, n.$$

where $\eta(\mathbf{x}_j; \boldsymbol{\beta})$ is called the *regression function*. Its form is usually known, though in the problem of *model selection*, one has to choose from a number of possible competing models. The errors ε_j are assumed to be independent with $\mathrm{E}(\varepsilon_i) = 0$. We shall consider mainly the case where all the error variances are equal $\mathrm{Var}(\varepsilon_i) = \sigma^2$, but will consider the heteroscedastic case too. The $\boldsymbol{\beta}$ are unknown coefficients that have to be estimated.

An immediate question is why one would wish to use a metamodel to represent the output, as opposed to having a direct model representation. The main reason is that a metamodel provides a very succinct representation of the output response and its dependence on concomitant factors. Therefore, when the representation is a good one, a metamodel enables the nature of the variation in simulation output to be accurately characterized in a simple and transparent way. Partly as a consequence of this, a metamodel provides an excellent and efficient vehicle for extracting, from the data, information about the system under investigation.

Use of a metamodel is very advantageous when implementing a bootstrap analysis. We simply treat the fitted metamodel as the surrogate for the real simulation model and analyze (using bootstrapping) the fitted metamodel as if it were the simulation model.

We shall discuss the some of the uses of metamodels in Section 5.5. However, we shall first discuss the mechanics of obtaining bootstrap samples from metamodels. This is done in the next two subsections. We consider linear metamodels first.

5.3 Linear metamodels

In the linear metamodel, observations are assumed to take the form

$$Y_j = \mathbf{x}_j^{\mathrm{T}} \boldsymbol{\beta} + \varepsilon_j, \quad j = 1, 2, \ldots, n,$$

where the errors are assumed to be independent and identically distributed with $\mathrm{E}(\varepsilon_i) = 0$ and $\mathrm{Var}(\varepsilon_i) = \sigma^2$, and the $\boldsymbol{\beta}$ are unknown and have to be estimated. Let

$$\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^{\mathrm{T}}, \qquad \mathbf{Y} = (Y_1, \ldots, Y_n)^{\mathrm{T}}$$

and

$$\hat{\beta} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}$$

be the usual least squares estimate (equivalent to maximum likelihood when the ε_i are normally distributed). The fitted values are

$$\widehat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}, \tag{20}$$

where

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}} \tag{21}$$

is the well-known ‘hat’ matrix.

The raw residuals are $\mathbf{r} = (\mathbf{I}-\mathbf{H})\mathbf{Y}$. One way of resampling is to sample from these residuals as if they were the unknown errors. This gives the bootstrap sample

$$Y_j^* = \mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}} + r_j^*, \quad j = 1, 2, \ldots, n. \tag{22}$$

We shall call this *residual sampling*. However we have that $\mathrm{E}(\mathbf{r}) = \mathbf{0}$ and $\mathrm{Var}(\mathbf{r}) = \sigma^2(\mathbf{I} - \mathbf{H})$. Thus a better bootstrapping method is to sample from the *adjusted residuals*

$$e_j = \frac{r_j}{(1 - h_{jj})^{1/2}}, \tag{23}$$

where h_{jj} is the jth diagonal entry of $\mathbf{H}$ (h_{jj} is commonly known as the *leverage* of the jth point). Bootstrap sampling from these e_j yields bootstrap samples of the form

$$Y_j^* = \mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}} + e_j^*, \quad j = 1, 2, \ldots, n. \tag{24}$$

We shall call this *adjusted residual sampling*.

An alternative is to use *case sampling* and to sample from the $(Y_j, \mathbf{x}_j)$ pairs directly. The main problem with this method is that it introduces extra variation into the data. In the conventional situation $\mathbf{X}$ is regarded as fixed, but using case sampling replaces this by a bootstrap $\mathbf{X}^*$ which will be variable. This problem is particularly pronounced when the number of parameters, p say, is large compared with n. There may then be a nonnegligible probability that the case sampled matrix $\mathbf{X}^*$ results in a singular design matrix, with not all parameters estimable.

A partial correction can often be made. For example suppose we are interested in the distribution of $\mathbf{c}^{\mathrm{T}}\hat{\boldsymbol{\beta}}$ where $\mathbf{c}$ is a constant vector. Then $\mathrm{Var}[\mathbf{c}^{\mathrm{T}}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})] = \sigma^2\mathbf{c}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{c}$. We can then study the behavior of

$$\frac{\mathbf{c}^{\mathrm{T}}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})}{\mathbf{c}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{c}}$$

indirectly by considering the bootstrap version

$$\frac{\mathbf{c}^{\mathrm{T}}(\hat{\boldsymbol{\beta}}^* - \boldsymbol{\beta}^*)}{\mathbf{c}^{\mathrm{T}}(\mathbf{X}^{*\mathrm{T}}\mathbf{X}^*)^{-1}\mathbf{c}}$$

(where $\boldsymbol{\beta}^* = \hat{\boldsymbol{\beta}}$).

A common variation is when the errors are heteroscedastic so that their variances are unequal. In this case resampling can be carried out as

$$Y_j^* = \mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}} + e_j t_j^*, \quad j = 1, 2, \ldots, n,$$

with e_i as defined in (23) and with t_j^* independent and random with $\mathrm{E}(t_j^*) = 0$, $\mathrm{Var}(t_j^*) = 1$. Wu (1986) suggests sampling the t_j^* with replacement from the set

$$(r_j - \bar{r}) \Big/ \left[n^{-1} \sum_{i=1}^{n} (r_i - \bar{r})^2 \right]^{1/2}, \quad j = 1, 2, \ldots, n.$$

5.4 Nonlinear metamodels

The previous ideas apply with little change to nonlinear models. For additive errors we have

$$Y_j = \eta(\mathbf{x}_j; \boldsymbol{\beta}) + \varepsilon_j, \quad j = 1, 2, \ldots, n, \tag{25}$$

where $\eta(\mathbf{x}_j; \boldsymbol{\beta})$ is not necessarily linear in $\boldsymbol{\beta}$. Again $\hat{\boldsymbol{\beta}}$ can be obtained by ML say and residuals formed: $r_j = Y_j - \eta(\mathbf{x}_j; \hat{\boldsymbol{\beta}})$, $j = 1, 2, \ldots, n$. This time there is no obvious equivalent of standardization so the bootstrap sample is simply

$$Y_j^* = \eta(\mathbf{x}_j; \hat{\boldsymbol{\beta}}) + r_j^*, \quad j = 1, 2, \ldots, n.$$

However the method extends easily to more complex nonlinear situations. For example it may be that we have observations $(y_j, \mathbf{x}_j)$ where

$$y_j \sim \mathrm{Poisson}(\mu_j), \quad \mu_j = \exp\bigl(\mathbf{x}_j^{\mathrm{T}} \boldsymbol{\beta}\bigr).$$

Again $\hat{\boldsymbol{\beta}}$ can be obtained by ML and a bootstrap sample can then be formed by drawing a sample as

$$y_j^* \sim \mathrm{Poisson}\bigl(\mu_j^*\bigr), \quad \mu_j^* = \exp\bigl(\mathbf{x}_j^{\mathrm{T}} \hat{\boldsymbol{\beta}}\bigr).$$

A clear introduction to resampling using *generalized linear models* is given by Hjorth (1994).

5.5 Uses of metamodels

We end by discussing some illustrations of *when* we might wish to use a metamodel and where bootstrap resampling would be useful. We give three examples.

The first is when the regression function $\eta(\mathbf{x}; \boldsymbol{\beta})$ represents a performance index and we might wish to construct a confidence interval for it at some given value of $\mathbf{x}$, $\mathbf{x}_0$ say. Suppose that we draw B sets of bootstrap observations of the form (24)

$$\bigl\{Y_j^{(i)*} = \eta(\mathbf{x}_j; \hat{\boldsymbol{\beta}}) + e_j^{(i)*}, j = 1, 2, \ldots, n\bigr\}, \quad i = 1, 2, \ldots, B,$$

and from each set we calculate a bootstrap estimator $\hat{\boldsymbol{\beta}}^{(i)*}$ of $\hat{\boldsymbol{\beta}}$ using the ML method say. Then the corresponding set of bootstrap regression function values $\eta(\mathbf{x}_0, \hat{\boldsymbol{\beta}}^{(i)*})$ can be used to calculate a bootstrap confidence interval. For instance we could use (13) with $\hat{\boldsymbol{\eta}}^*$ in that formula having components $\hat{\eta}_i^* = \eta(\mathbf{x}_0, \hat{\boldsymbol{\beta}}^{(i)*})$.

The second example is when we wish to find an optimal setting for $\mathbf{x}$. We consider the simplest case where

$$Y = \eta(x; \boldsymbol{\beta}) + \varepsilon,$$

with x scalar and where

$$\eta(x; \boldsymbol{\beta}) = \beta_0 + \beta_1 x + \beta_2 x^2.$$

Then the optimal setting is given by $\mathrm{d}\eta/\mathrm{d}x = 0$, i.e., at $x_{opt} = -\beta_1/(2\beta_2)$. Using ML to estimate this yields

$$\hat{x}_{opt} = -\frac{\hat{\beta}_1}{2\hat{\beta}_2}.$$

Again a simple confidence interval for the unknown x_{opt} is given by (13), only this time, with $\hat{\boldsymbol{\eta}}^*$ in that formula having components $\hat{\eta}_i^* = -\hat{\beta}_1^{(i)*}/(2\hat{\beta}_2^{(i)*})$. This is an example where bootstrapping furnishes a relatively easy answer to what in classical statistics is a problem that is not straightforward.

The final example concerns the identification of important factors. Suppose we are interested in identifying those coefficients in the regression function $\eta(\mathbf{x}; \boldsymbol{\beta})$ for which $|\beta_i| > \beta_{0i}$ where $\beta_{0i} > 0$ are given constants. Let $S = \{\beta_i\colon |\beta_i| > \beta_{0i}\}$ denote the *important set of coefficients*. The obvious estimate is to select those coefficients β_i for which $|\hat{\beta}_i| > \beta_{0i}$, i.e., $\widehat{S} = \{\beta_i\colon |\hat{\beta}_i| > \beta_{0i}\}$. Bootstrapping is a simple way of assessing the stability of this choice. We generate B bootstrap samples $(y_j^{(k)*}, \mathbf{x}_j^{(k)*})$, $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, B$, using either residual sampling or case sampling say and fit the regression model to each bootstrap sample to give bootstrap estimates $\hat{\boldsymbol{\beta}}^{(k)*}$, $k = 1, 2, \ldots, B$. We then calculate $\widehat{S}^{(k)*} = \{\beta_i\colon |\hat{\beta}_i^{(k)*}| > \beta_{0i}\}$, $k = 1, 2, \ldots, K$. Now assume that B is sufficiently large so that each distinct bootstrap important set that has been obtained occurs a reasonable number of times. Then a $(1-\alpha)$ confidence region for the unknown true important set can be constructed by selecting bootstrap important sets in decreasing order of their observed frequency of occurrence until a proportion $(1-\alpha)$ of the $\widehat{S}^{(k)*}$ have been chosen.

Related to identifying important factors in metamodels is the rather more difficult problem of metamodel selection. We consider this in the next section.

5.6 Metamodel comparison and selection

Metamodel comparison and selection is a difficult subject. The main trouble is that models that we wish to compare may be of different functional complexity. The statistical properties associated with quantities derived from different models may therefore be hard to put on a common scale on which they can be compared.

A reasonably satisfactory approach is based on the use of *cross-validation*. We suppose that the initial data has the form

$$S = \{(y_j, \mathbf{x}_j), j = 1, 2, \ldots, n\}$$

with

$$y_j = \eta(\mathbf{x}_j, \boldsymbol{\beta}) + \varepsilon_j, \quad j = 1, 2, \ldots, n.$$

Suppose that our fitted regression is

$$\hat{y}(\mathbf{x}) = \eta(\mathbf{x}, \hat{\boldsymbol{\beta}}).$$

In its basic form cross-validation is used to assess how effective the fitted regression is for predicting the response at some new design point $\mathbf{x}_{new}$. Rather than explicitly choosing such a new point, a simple idea is the *leave-one-out* method where we drop an observed point $(y_j, \mathbf{x}_j)$ from the set of all observed points, fit the metamodel to the remaining points $S_{-j} = S \setminus (y_j, \mathbf{x}_j)$ giving the fitted regression as

$$\hat{y}_{-j}(\mathbf{x}) = \eta(\mathbf{x}, \hat{\boldsymbol{\beta}}_{-j}),$$

and then look at the squared difference between the omitted y_j and the value of y at $\mathbf{x}_j$, as predicted by the fitted model; i.e.,

$$\bigl(y_j - \hat{y}_{-j}(\mathbf{x}_j)\bigr)^2.$$

If we do this even handedly by leaving out each observation in turn we have as an *estimate of cross-validation prediction error*

$$\widehat{L}_{\mathrm{CV}} = n^{-1} \sum_{j=1}^{n} \bigl(y_j - \hat{y}_{-j}(\mathbf{x}_j)\bigr)^2. \tag{26}$$

It turns out that, for the linear regression model, this formula simplifies elegantly to one where we only need comparisons with the one model fitted to all the original observations

$$\widehat{L}_{\mathrm{CV}} = n^{-1} \sum_{j=1}^{n} \frac{(y_j - \hat{y}(\mathbf{x}_j))^2}{1 - h_{jj}},$$

where

$$\hat{y}(\mathbf{x}_j) = \mathbf{x}_j^{\mathrm{T}} \hat{\boldsymbol{\beta}},$$

and h_{jj} is the jth main diagonal entry in the hat-matrix (21), as before.

The distributional property of $\widehat{L}_{\mathrm{CV}}$ can be obtained in the usual way by bootstrapping to get B bootstrap samples. If we use case resampling say then

$$S^{(i)*} = \bigl\{\bigl(y_j^{(i)*}, \mathbf{x}_j^{(i)*}\bigr), j = 1, 2, \ldots, n\bigr\},$$

where each $(y_j^{(i)*}, \mathbf{x}_j^{(i)*})$ is an observation drawn at random from the set S.

We turn now to model selection. For simplicity we shall only consider the linear case where $y = \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} + \varepsilon$, though with an obvious adjustment, the discussion does generalize. Let $\boldsymbol{\beta}$ be of dimension p. If we are uncertain as to which design variables are really needed in the model, then we would have in principle up to 2^p models to choose from with $\binom{p}{q}$ (sub)models where there were precisely q covariates selected. We denote a typical submodel by M. The estimate of prediction error using (26) is

$$\widehat{L}_{\mathrm{CV}}(M) = n^{-1}\sum_{j=1}^{n}\frac{(y_j - \hat{y}_M(\mathbf{x}_j))^2}{1 - h_{Mjj}}.$$

It turns out that this measure does not work as well as it might even when n is large. (This is an example previously referred to in Section 4.4 where inconsistency arises not in the problem itself but in the bootstrap sampling method.) A much better variant is not to leave out just one observation at a time but instead to split the observations into two: a *training* set, and an *assessment* set with respectively $n_t = n - m$ and $n_a = m$ observations in each. We shall moreover select not one but K such pairs and denote the sets as $S_{t,k}$ and $S_{a,k}$, $k = 1, 2, \ldots, K$. We shall write $\hat{\boldsymbol{\beta}}_{Mk}$ for the coefficients of model M fitted to the kth training set, and write $\hat{y}_{Mjk} = \mathbf{x}_{Mj}^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{Mk}$ for the value of $\mathbf{x}_j^{\mathrm{T}}\boldsymbol{\beta}$ predicted by this model M at $\mathbf{x}_j$. Then $\widehat{L}_{\mathrm{CV}}(M)$ becomes

$$\widehat{L}_{\mathrm{CV}}(M) = K^{-1}\sum_{k=1}^{K} m^{-1}\sum_{j\in S_{a,k}}(y_j - \hat{y}_{Mjk})^2. \tag{27}$$

We use the same set of K pairs for all models being compared. Provided $n - m \to \infty$ and $m/n \to 1$ as $n \to \infty$ then selecting M to minimize (27) will yield a consistent estimator for the correct model. When n is small it may not be possible to select m large enough in which case Davison and Hinkley (1997) suggest taking $m/n \simeq 2/3$.

The variability of $\widehat{L}_{\mathrm{CV}}(M)$ can be estimated by bootstrapping in the usual way.

6 Bootstrap comparisons

In this section we briefly consider problems where there are a number of different samples to compare. A more comprehensive discussion focusing on the issue of selecting between systems is given in Chapter 17: Selecting the Best System.

We suppose that we have m data sets

$$\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{i,n_i}), \quad i = 1, 2, \ldots, m, \tag{28}$$

the ith being of size n_i runs. This situation covers two different scenarios in particular.

The first is where we are considering the question: Does the output of the simulation model accurately represent the output of the system being modeled? Here we have just two data sets, so that $m = 2$, and one data set comprises observations of a real system whilst the other comprises the observed output from a simulation model. Thus this is a question of validation.

The second is when we have a number of different simulation models to assess. Here validation against real data is not the issue. The different models simply represent different systems, and their validity is not in doubt. Instead we are just interested in assessing how differently the systems behave, as represented by the output of their simulation models. Typically we might be interested in which system produces the greatest average output, or least average delay. Alternatively we might wish to compare output variability.

A similar methodology can be applied to either problem. We discuss this in the remainder of this section.

6.1 Goodness-of-fit and validation

A discussion of validation involving trace-driven simulation is given in Kleijnen et al. (2000, 2001). The basic idea used there can be generalized. We treat the problem as essentially one of goodness-of-fit.

A simple starting point is where we have n simulation outputs y_j, $j = 1, 2, \dots, n$, to compare with m observations y_j^{Real}, $j = 1, 2, \dots, m$, of a real system and we wish to know if the two samples are identically distributed. A good goodness-of-fit statistic is the two sample Cramér–von Mises statistic (see Anderson, 1962)

$$W^2 = \int \left[F_n(y) - F_m^{Real}(y)\right]^2 \mathrm{d}H_{n+m}(y), \tag{29}$$

where F_n, F_m^{Real} and H_{n+m} are respectively the EDFs of the simulated, real and combined samples. The asymptotic distribution of W^2 is known when the samples are drawn from continuous distributions, but it is easy to set up a bootstrap version of the test that can handle discrete distributions just as well.

We can use a bootstrapping method as follows. We obtain B pairs of bootstrap samples $(\mathbf{y}^{(i)*}, \mathbf{y}^{Real(i)*})$, $i = 1, 2, \dots, B$, each sample in the pair being obtained by resampling with replacement from $\mathbf{y}$ and $\mathbf{y}^{Real}$ combined but with $\mathbf{y}^{(i)*}$ the same sample size as $\mathbf{y}$, and $\mathbf{y}^{Real(i)*}$ the same sample size as $\mathbf{y}^{Real}$. Denote the EDFs of the samples of each pair by $F_n^{(i)*}$ and $F_n^{Real(i)*}$, $i = 1, 2, \dots, B$. We can now calculate B bootstrap two sample Cramér–von Mises statistics from each pair

$$W^{(i)*2} = \int \left[F_n^{(i)*}(y) - F_m^{Real(i)*}(y)\right]^2 \mathrm{d}H_{n+m}^{(i)*}(y). \tag{30}$$

Under the null assumption that $\mathbf{y}$ and $\mathbf{y}^{Real}$ are drawn from the same distribution it follows that each $W^{(i)*2}$ is just a bootstrap version of W^2. We can thus calculate a critical p-value from the EDF of the $W^{(i)*2}$, $i = 1, 2, \ldots, B$. The null hypothesis that $\mathbf{y}$ and $\mathbf{y}^{Real}$ are drawn from the same distribution can be tested simply by checking if the original W^2 exceeds this p-value or not.

The validation method just described is easily extended to allow the representational accuracy of a number of different competing simulation models to be compared against (the same) real data. We order the Cramér–von Mises goodness-of-fit statistics (29) obtained from comparing the real data with the simulated output of each of the models. We can assess the reliability of the comparison by bootstrapping as in (30) to obtain bootstrap distributions of each such Cramér–von Mises statistic and looking at the degree of overlap among these distributions.

When the W^2 statistic shows the data samples $\mathbf{y}_j$ to be significantly different, there is an interesting decomposition of W^2 that allows the nature of the differences to be more clearly identified. Durbin and Knott (1972) show that in the one sample case where $F_n(\cdot)$ is being compared with a uniform null distribution, so that $W^2 = n\int_0^1 [F_n(y) - y]^2\,\mathrm{d}y$, then W^2 has the orthogonal representation

$$W^2 = \sum_{j=1}^{\infty} (j\pi)^{-2} z_{nj}^2,$$

where the z_{nj} are scaled versions of the coefficients in the Fourier decomposition of the process $\sqrt{n}(F_n(y) - y)$. The z_{nj} are stochastic of course, depending on $F_n(\cdot)$, but they are independent and, under the null, have identical distributions. It can be shown that z_{n1}, z_{n2} and z_{n3} are dependent essentially exclusively on deviations respectively of the mean, variance and skewness of $F(\cdot)$ from that of the null. In other words these components z_{n1}, z_{n2} and z_{n3} can be used to provide convenient statistical tests for these differences. Cheng and Jones (2000, 2004) describe a generalization of the results of Durbin and Knott (1972), which they term *EDFIT statistics*, suitable for comparison of data of the form (28) arising in the simulation context. Their formulation uses ranks rather than the original observations. Though this leads to some loss of power the method does then have the advantage of enabling critical values of tests to be easily obtained by bootstrapping, and moreover in an exact way.

6.2 *Comparison of different systems*

We now consider the situation of (28) where the data sets are outputs from different simulation models only, and no comparison is made with real data. Here we are interested simply in making comparisons between different models. The discussion of the previous subsection goes through with no essential change. The only difference is that all samples have the same status; there is no sample being singled out as special by being drawn from a real system. The EDFIT approach can therefore be used directly, with bootstrapping providing critical null values.

A more direct approach, not using goodness-of-fit ideas, is possible. We focus on examining differences between the means, $\bar{y}_i$, $i = 1, 2, \dots, m$, of the m samples of (28). Comparison of other sample statistics can be handled in the same way. If we suppose that largest is best we can rank the means as $\bar{y}_{(1)} > \bar{y}_{(2)} > \cdots > \bar{y}_{(m)}$. The question then is how stable is this order? Bootstrapping provides an easy answer. We generate B bootstrap sets of observations

$$\mathbf{y}_i^{(k)*} = \big(y_{i1}^{(k)*}, y_{i2}^{(k)*}, \dots, y_{i,n_i}^{(k)*}\big), \quad i = 1, 2, \dots, m, k = 1, 2, \dots, B, \tag{31}$$

where each sample $\mathbf{y}_i^{(k)*}$, $k = 1, 2, \dots, B$, is obtained by sampling with replacement from $\mathbf{y}_i$. We then order the means in each bootstrapped set: $\bar{y}_{(1)}^{(k)*} > \bar{y}_{(2)}^{(k)*} > \cdots > \bar{y}_{(m)}^{(k)*}$, $k = 1, 2, \dots, B$. The frequency count of how many times the mean of a given sample comes out on top is a measure of its relative merit. A more comprehensive picture is provided by setting up a two-way table showing the number of times $\bar{y}_i^{(k)*} > \bar{y}_j^{(k)*}$ out of the B bootstrapped sets, for each possible pair $1 \leqslant i, j \leqslant m$.

The parametric form of the bootstrapping procedure is equally easy to implement, though it is not clear there is much advantage to be had in doing so. Suppose for example that the ith sample $\mathbf{y}_i$ is drawn from the distribution $F^{(i)}(\cdot, \boldsymbol{\theta}_i)$. Usually the $F^{(i)}(\cdot, \boldsymbol{\theta}_i)$ will have the same functional form, for example all normal distributions, but the procedure works equally well when the $F^{(i)}$ are functionally different. Let the mean of the $F^{(i)}$ distribution be $\mu^{(i)}(\boldsymbol{\theta}_i)$. Then we can estimate, by ML say, the parameters of each distribution from their corresponding sample. Let these estimates be $\hat{\boldsymbol{\theta}}_i$. We can then carry out parametric bootstrapping by forming (31) but this time with each sample $\mathbf{y}_i^{(k)*}$, $k = 1, 2, \dots, B$, obtained by sampling from the fitted distribution $F^{(i)}(\cdot, \hat{\boldsymbol{\theta}}_i)$. The analysis then proceeds as before.

7 Bayesian models

A full discussion of Bayesian techniques is given in Chapter 9. Much of the current popularity of Bayesian methods comes about because increased computing power makes possible Bayesian analysis by numerical procedures, most notably Markov chain Monte Carlo (MCMC). See Gilks et al. (1996). This allows numerical sampling to be carried out in much the same way as is done in bootstrap and other resampling procedures. We indicate the commonality by considering just one or two situations where Bayesian methods and resampling overlap.

We again focus on a random sample $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ drawn from a distribution $F(w, \boldsymbol{\theta})$. As in the situation where we considered ML estimation, the form of F is known but it depends on a vector $\boldsymbol{\theta}$ of parameters. In the ML estimation situation $\boldsymbol{\theta}_0$, the true value of $\boldsymbol{\theta}$, was assumed unknown. In the Bayesian case a prior distribution is assumed known. We consider just the continuous case and denote the p.d.f. of the prior by $\pi(\boldsymbol{\theta})$. The main step in Bayesian analysis is to construct the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{w})$ which shows how the sample $\mathbf{w}$, which depends on $\boldsymbol{\theta}$, has modified the prior. The Bayesian formula is

$$\pi(\boldsymbol{\theta}|\mathbf{w}) = \frac{p(\mathbf{w}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{\int p(\mathbf{w}|\boldsymbol{\varphi})\pi(\boldsymbol{\varphi})\,\mathrm{d}\boldsymbol{\varphi}}. \tag{32}$$

The difficult part is the evaluation of the normalizing integral $\int p(\mathbf{w}|\boldsymbol{\varphi}) \times \pi(\boldsymbol{\varphi})\,\mathrm{d}\boldsymbol{\varphi}$. MCMC has proved remarkably successful in providing a powerful numerical means of doing this.

In the context of simulation a Bayesian approach is useful in assessing uncertainty concerning input parameter values used in a simulation model. We can adopt the viewpoint given in (17) and regard the runs as taking the form

$$y_j = y(\mathbf{u}_j, \mathbf{v}_j, \boldsymbol{\theta}, \mathbf{x}_j), \quad j = 1, 2, \ldots, n, \tag{33}$$

only now $\boldsymbol{\theta}$ is assumed to have a Bayesian prior distribution $\pi(\boldsymbol{\theta})$. Note that in this situation we do *not* have the equivalent of data $\mathbf{w}$ from which to calculate a posterior distribution for $\boldsymbol{\theta}$. What we can do is to treat the prior $\pi(\boldsymbol{\theta})$ as *inducing a prior distribution* on $y = y(\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}, \mathbf{x})$. In this sense we can think of y itself as having a prior which can then be estimated by sampling $\boldsymbol{\theta}$ from its prior $\pi(\boldsymbol{\theta})$, yielding values $\boldsymbol{\theta}_j$, $j = 1, 2, \ldots, n$, and then running the model to produce a set of observations

$$y_j = y(\mathbf{u}_j, \mathbf{v}_j, \boldsymbol{\theta}_j, \mathbf{x}_j), \quad j = 1, 2, \ldots, n.$$

The EDF of these y_j then estimates this prior distribution of y. This process clearly is the analogue to the classical situation where $\boldsymbol{\theta}$ is estimated from data $\mathbf{w}$, as is supposed in (17), and we then allowed for this variability in the y_j by replacing $\hat{\boldsymbol{\theta}}(\mathbf{w})$ by $\boldsymbol{\theta}_j^*$ calculated from (18) or (19).

More interestingly we can take this a step further if there are real observations available of the y process itself. We do not attempt a fully general formulation but give an example to indicate the possibilities. Suppose that

$$y_j^{Real} = y^{Real}(\mathbf{x}_j), \quad j = 1, 2, \ldots, n,$$

comprise n observations on the real system being modeled by the simulation. The precise relationship between the simulation output $y = y(\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}, \mathbf{x})$ and $y^{Real}(\mathbf{x})$ is not known. However, we might assume that

$$y(\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}, \mathbf{x}) \sim \mathrm{N}\big(y^{Real}(\mathbf{x}), \sigma^2\big),$$

where σ^2 is an additional parameter that will also be treated as Bayesian in the sense of having a prior. We can express our great prior uncertainty about σ by assuming a *reference prior distribution*, $\rho(\sigma)$, for it. Then, by (32), the posterior distribution of $(\boldsymbol{\theta}, \sigma)$ is proportional to

$$\sigma^{-n}\pi(\boldsymbol{\theta})\rho(\sigma)\exp\left\{-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left[y(\mathbf{u}, \mathbf{v}, \boldsymbol{\theta}, \mathbf{x}_j) - y^{Real}(\mathbf{x}_j)\right]^2\right\}.$$

The posterior distribution can thus be obtained by MCMC methods for example. An interesting application of this problem occurs in epidemiological modeling. Suppose that y is a measure of the progress of an epidemic that is dependent on factors $\boldsymbol{\theta}$ for which there are previous measurements or for which there is expert information. A Bayesian approach is then a natural way of incorporating this prior information. We need also to have a good epidemic model for producing a simulated y. Thus reverse use of simulation as indicated above has allowed this prior information to be updated.

8 Time series output

Bootstrapping of time series is a well-studied problem. In simulation the most likely use of such procedures is to generate correlated input for a model. As usual the parametric form is relatively easy to explain and implement and we discuss this first.

8.1 Residual sampling

We consider the case where the time-series is an autoregressive model. Here the residual sampling method used to construct a bootstrap metamodel applies with little change. Suppose we have the autoregressive model

$$Y_t = \sum_{j=1}^{p} a_j Y_{t-j} + \varepsilon_t,$$

where the ε_t are independently and identically distributed, commonly called the *innovations*. Suppose that $y_1, y_2, \ldots, y_n$ are a series drawn from this model. Then we can estimate the a_j by least squares say, yielding the estimates $\hat{a}_j$, $j = 1, 2, \ldots, p$, and form the residuals

$$r_t = y_t - \sum_{j=1}^{p} \hat{a}_j y_{t-j}, \quad t = p+1, p+2, \ldots, n.$$

We can then form the bootstrap time-series as

$$y_t^* = \sum_{j=1}^{p} \hat{a}_j y_{t-j}^* + r_t^*, \quad t = 1, 2, \ldots, n,$$

by sampling the r_t^* from the EDF of the residuals $\{r_t\}$. We need $y_1^*, y_2^*, \dots, y_p^*$ to initiate the process, but if we assume that the observed series is stationary, it is probably easiest to simply initiate the series with some arbitrary starting values, possibly the original $y_1, y_2, \dots, y_p$, then run the bootstrapping sufficiently long to make the initial effects negligible and collect the actual y_t^* from that point on. Freedman (1984) gives conditions for the asymptotic validity of this procedure, Basawa et al. (1989) extending these results to the case of nonlinear time-series.

8.2 Block sampling

For time-series the analogue of case sampling is *block sampling*. We cannot sample individual observations y_t because this loses the correlation between observations. If the series is long enough then we can take $n = bl$ and think of the series as comprising b blocks each of length l. We write the ith block as $\mathbf{y}_i = (y_{l(i-1)+1}, y_{l(i-1)+2}, \dots, y_{li})$, $i = 1, 2, \dots, b$. Bootstrapping is done by sampling blocks with replacement from this set of b blocks, retaining the order of the observations in each block when writing down the individual observations of the bootstrapped series. A balance needs to be struck between having block lengths long enough to retain the correlation properties between neighboring observations, and having enough blocks to measure the inherent variation of the series. A typical compromise is to use say $b = l = n^{1/2}$ so that both quantities tend to infinity as $n \to \infty$. The methodology is related to the batch means method for obtaining independent observations, see Chapter 15.

A major weakness of block sampling is the loss of correlation incurred by the random sampling of blocks. This loss of correlation is called *whitening*. It is especially serious when the statistic of interest involves correlations of high lag. The crude block sampling just described may be quite ineffective if the size of blocks is not large enough, because calculation involves quantities which straddle blocks and which are then not sufficiently correlated because of the whitening. There are many variants of block sampling aimed at reducing the effect of whitening in specific situations. A good example is estimation of the lag m covariance

$$c_m = \frac{1}{n-m}\sum_{t=1}^{n-m}(y_t - \bar{y})(y_{t+m} - \bar{y}).$$

Here we can define a two-dimensional process

$$\mathbf{z}_t = \begin{pmatrix} z_{1t} \\ z_{2t} \end{pmatrix} = \begin{pmatrix} y_t \\ y_{t+m} \end{pmatrix}, \quad t = 1, 2, \dots, n-m,$$

with $\bar{z}_i = (n-m)^{-1}\sum_{t=1}^{n-m} z_{it}$ and think of c_1 as a statistic of this process

$$c_1 = \frac{1}{n-1}\sum_{t=1}^{n-1}(z_{1t} - \bar{z}_1)(z_{2t} - \bar{z}_2).$$

We can then obtain bootstrap $\mathbf{z}_t^*$ by sampling with replacement from the set $\{\mathbf{z}_t\}$. The bootstrap lag m covariance then clearly substantially retains the covariance of the original series as we have, in effect, bootstrap sampled the terms $(y_t - \bar{y})(y_{t+m} - \bar{y})$ appearing in the formula giving c_m. Generalizations of this technique are known as *block of blocks sampling*.

8.3 Spectral resampling

Residual and block sampling are time domain techniques. An alternative approach is to sample in the frequency domain. A big advantage is that spectral increments are uncorrelated, and for Gaussian processes this strengthens to independent increments.

Suppose we have $n = 2m + 1$ observations

$$y_t, \quad t = -m, -m+1, \ldots, m-1, m,$$

for which there is a continuous spectral density $S(\omega)$ and define the frequencies $\omega_k = 2\pi k/n$, $-m \leqslant k \leqslant m$. Then the observations have the spectral representation

$$y_t = \sum_{k=-m}^{m} a_k \mathrm{e}^{\mathrm{i}\omega_k t},$$

where

$$a_k = n^{-1} \sum_{t=-m}^{m} y_t \mathrm{e}^{-\mathrm{i}\omega_k t}.$$

In this section $\mathrm{i} = \sqrt{-1}$. For a Gaussian process the real and purely imaginary components of the a_k are independent, and the a_k are asymptotically so.

Norgaard (1992) gives two possible ways of obtaining bootstrap samples of the a_k. The simplest version is to draw a_k^* at random from one of the twelve elements

$$(\pm a_{k-1}, \pm a_k, \pm a_{k+1}, \pm \mathrm{i}a_{k-1}, \pm \mathrm{i}a_k, \pm \mathrm{i}a_{k+1})$$

when $0 < k < n$, and to draw a_n^* at random from one of the twelve elements

$$(\pm a_{n-1}, \pm a_n, \pm a_n, \pm \mathrm{i}a_{n-1}, \pm \mathrm{i}a_n, \pm \mathrm{i}a_n)$$

(note the repetition of some elements in this latter case). In both cases we set

$$a_{-k}^* = \overline{a_k^*},$$

i.e., a_{-k}^* is the complex conjugate of a_k^*. The value of a_0^* needs to be real. The simplest case is when $\mathrm{E}(Y_t) = 0$ when we can select a_0^* at random as one of the four elements $(\pm a_0, \pm |a_1|)$.

A variant which seems to be significantly superior is called the *extended circle* (EC) method by Norgaard (1992). Here we select $|a|_k^*$ as a random element of the set $(|a_{k-1}|, |a_k|, |a_{k+1}|)$ and then give this a random spin in both the real and imaginary directions:

$$a_k^* = |a|_k^* (\cos \phi_{1k} + \mathrm{i} \sin \phi_{2k}), \quad k = 1, 2, \ldots, n,$$

where ϕ_{1k} and ϕ_{2k} are independent $\mathrm{U}(-\pi, \pi)$ angles. We have $a_{-k}^* = \overline{a_k^*}$ as before.

9 Final comments

It is hoped that this chapter has given a hint of the richness of the topic and its practical potential.

We have not discussed implementation issues at all. It should be clear however that most of the procedures are quite elementary to set up. Efron and Tibshirani (1993, Appendix) gives good advice on how best to implement bootstrapping, in a general statistical context. In the simulation context this is still perhaps an issue that deserves more attention. When these are cheap, runs of the model are to be preferred. However in complex models, simulation runs can be expensive to make. It would be convenient to have a toggle that switches from simulation model sampling to bootstrap sampling. A framework that allows this to be done, and its actual implementation in simulation packages would allow bootstrapping to be much more available on tap.

There are many topics not covered in this chapter that could well have been included. We have for instance, not touched on the closely related technique of jackknifing. Other topics we have made only cursory mention, such as cross-validation and when bootstrapping fails.

The main thing that distinguishes simulation experiments from other statistical experiments is the control exercised in the random sampling. Variance reduction is perhaps special to simulation experiments. There has been some consideration of how bootstrapping and variance reduction might be combined (see Hall, 1989). A particular example is the use of balanced resampling is discussed by Davison et al. (1986), who also consider use of control variates. Importance resampling has been suggested by Johns (1988). This perhaps is an area that still needs further work.

References

Anderson, T.W. (1962). On the distribution of the two-sample Cramér–von Mises criterion. *Annals of Mathematical Statistics* 33, 1148–1159.

Barton, R.R., Schruben, L.W. (1993). Uniform and bootstrap resampling of empirical distributions. In: Evans, G.W., Mollaghasemi, M., Russell, W.E., Biles, E.C. (Eds.), *Proceedings of the 1993 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 503–508.

Basawa, I.V., Mallik, A.K., McCormick, W.P., Taylor, R.L. (1989). Bootstrapping explosive autoregressive processes. *The Annals of Statistics* 17, 1479–1486.

Bickel, P.J., Freedman, D.A. (1981). Some asymptotic theory for the bootstrap. *The Annals of Statistics* 9, 1196–1217.

Cheng, R.C.H., Jones, O.D. (2000). Analysis of simulation factorial experiments by EDF resample statistics. In: Joines, J.A., Barton, R., Fishwick, P., Kang, K. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 697–703.

Cheng, R.C.H. (1995). Bootstrap methods in computer simulation experiments. In: Lilegdon, W.R., Goldsman, D., Alexopoulos, K., Kang, C. (Eds.), *Proceedings of 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 171–177.

Cheng, R.C.H., Jones, O.D. (2004). Analysis of distributions in factorial experiments. *Statistica Sinica* 14, 1085–1103.

Chernick, M.R. (1999). *Bootstrap Methods*. Wiley, New York.

Davison, A.C., Hinkley, D.V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge, England.

Davison, A.C., Hinkley, D.V., Schechtman, E. (1986). Efficient bootstrap simulations. *Biometrika* 73, 555–566.

Durbin, J., Knott, M. (1972). Components of Cramér–von Mises statistics I. *Journal of the Royal Statistical Society* 34, 290–307.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics* 7, 1–26.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American Statistical Association* 82, 171–185.

Efron, B., Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, New York–London.

Freedman, D. (1984). On bootstrapping two-stage least-squares estimates in stationary linear models. *The Annals of Statistics* 12, 827–842.

Ghosh, M., Parr, W.C., Singh, K., Babu, G.J. (1984). A note on bootstrapping the sample median. *The Annals of Statistics* 12, 1130–1135.

Gilks, W.R., Richardson, S., Spiegelhalter, D.J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London.

Hall, P. (1988). Rate of convergence in bootstrap approximations. *The Annals of Probability* 16, 1165–1685.

Hall, P. (1989). Antithetic resampling for the bootstrap. *Biometrika* 76, 713–724.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*. Springer-Verlag, New York.

Hjorth, J.S.U. (1994). *Computer Intensive Statistical Methods*. Chapman and Hall, London.

Johns, M.V. (1988). Importance sampling for bootstrap confidence itervals. *Journal of the American Statistical Association* 83, 709–714.

Kim, Y.B., Willemain, T.R., Haddock, J. (1993). The threshold bootstrap: A new approach to simulation output analysis. In: Evans, G.W., Mollaghasemi, M., Russell, W.E., Biles, E.C. (Eds.), *Proceedings of the 1993 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 498–502.

Kleijnen, J.P.C., Cheng, R.C.H., Bettonvil, B. (2000). Validation of trace-driven simulation models: More on bootrstrap tests. In: Joines, J.A., Barton, R.R., Kang, P.A., Fishwick, K. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 882–892.

Kleijnen, J.P.C., Cheng, R.C.H., Bettonvil, B. (2001). Validation of trace driven simulation models: Bootstrap tests. *Management Science* 47, 1533–1538.

Norgaard, A. (1992). Resampling stochastic processes using a bootstrap approach. In: *Bootstrapping and Related Topics*. *Lecture Notes in Economics and Mathematical Systems*, vol. 376. Springer-Verlag, New York, pp. 181–185.

Serfling, R.J. (1980). *Approximation Theorems of Mathematical Statistics*. Wiley, New York.

Shao, J., Tu, D. (1995). *The Jackknife and the Bootstrap*. Springer-Verlag, New York.

Shiue, W.-K., Xu, C.-W., Rea, C.B. (1993). Bootstrap confidence intervals for simulation outputs. *Journal of Statistical Computation and Simulation* 45, 249–255.

Singh, K. (1981). On the asymptotic accuracy of Efron's bootstrap. *The Annals of Statistics* 9, 1187–1195.

van Beeck, P. (1972). An application of Fourier methods to the problem of sharpening the Berry–Esséen inequality. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 15, 279–290.

von Mises, R. (1947). On the asymptotic distribution of differentiable statistical functions. *Annals of Mathematical Statistics* 18, 309–348.

Wu, C.F.J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics* 14, 1261–1295.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13015-7

Chapter 15

Correlation-Based Methods for Output Analysis

David Goldsman

School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
E-mail: sman@isye.gatech.edu

Barry L. Nelson

Department of Industrial Engineering and Management Sciences, Northwestern University, USA
E-mail: nelsonb@northwestern.edu

Abstract

We describe methods for estimating the variance of the sample mean of a steady-state simulation output process using correlation-based methods, all involving some form of "batching". Since no single method dominates all others across all measures of performance, we provide a framework for asymptotic comparison of such estimators. Research challenges are also identified.

1 Introduction

The input processes driving a simulation are usually random variables – examples include interarrival times of customers, resource service times, and machine breakdown times. Random input means that the output from the simulation is also apt to be random – for instance, customer waiting times, resource utilizations, and product cycle times. This in turn implies that runs of the simulation only yield *estimates* of the true measures of system performance – e.g., the mean customer waiting time, the long-run utilization, or the 80th percentile of product cycle time. Since these estimators are themselves random variables, they are therefore subject to sampling error which must be taken into account in a rigorous way if we are to make valid inferences or decisions concerning the performance of the underlying system. Thus, as part of a complete simulation study, we should always carry out a careful statistical analysis of the simulation's output.

The fundamental problem is that simulations almost never produce raw output that is independent and identically distributed (i.i.d.), much less normally distributed. For example, consecutive customer waiting times from a complicated queueing system pose a number of difficulties that hinder analysis via elementary statistical methods:

- Simulation data are not independent – typically, they are serially correlated. If one customer at the post office waits in line a long time, then the next customer is also likely to wait a long time.
- Simulation data are not identically distributed. Customers showing up early in the morning might have a much shorter wait than those who show up just before closing time.
- Simulation data are not normally distributed – waiting-time distributions are usually skewed to the right (and are certainly never less than zero).

These facts of life render it challenging to apply "classical" statistical techniques to the analysis of simulation output. So our purpose in this chapter is to present methods for statistical analysis of output from discrete-event computer simulations. We focus our discussion on data arising from *steady-state* (nonterminating) simulations, where the interest lies in the long-run behavior of the system. Presumedly this steady-state behavior is independent of the simulation's initial conditions. An example is a continuously running production line for which the experimenter is interested in some long-run performance measure. See Chapters 1 and 2 for examples and a precise definition of the "steady-state simulation" problem.

The main concern of the present chapter lies in studying methods for evaluating the variance of estimators derived from steady-state simulation output data. In particular, we concentrate on so-called "correlation-based" techniques, where we have knowledge of – and take advantage of – the correlation structure of certain stochastic processes. For a general introduction to estimation methods for simulation see Chapter 8.

Of course, many simulations are not run in steady state, e.g., the simulation of the ebb and flow of customers in a bank that opens and closes at certain times every day. Such *transient* (terminating) simulations use output analysis techniques that are primarily based on the method of independent replications (see Law and Kelton, 2000 and Chapter 8).

We henceforth assume access to steady-state data $Y_1, Y_2, \ldots, Y_n$, where Y_i might represent, e.g., the ith customer's waiting time in a specified queue. In this case, a good analysis might start off with, at the very least, an estimate of the unknown mean μ of the steady-state output process. Of course, the sample mean $\overline{Y}_n$ is the usual estimator for μ; but since the sample mean is a random variable, the experimenter should estimate the sample mean's variability as well. One such measure is simply $\sigma_n^2 \equiv n\,\mathrm{Var}(\overline{Y}_n)$, or almost equivalently, the *variance parameter*, $\sigma^2 \equiv \lim_{n\to\infty} \sigma_n^2$. Owing to the fact that outputs from steady-state simulations are almost never i.i.d., the "standard" sample-variance estimator for σ^2 is almost guaranteed to be biased (usually on the low side for queueing simulations), and should therefore never be used.

The remainder of this chapter addresses a number of valid methods for estimating σ_n^2 or σ^2. There are many techniques in the literature concerning this key task. We begin in Section 2 with some additional motivational comments and basic material to keep the discussion self-contained. In Section 3, we devote much our time to a tutorial on the well-known methods of nonoverlapping batch means (NBM) and standardized time series (STS); afterwards, Section 4 introduces overlapping batch means (OBM) estimators and overlapping versions of the STS estimators. These, along with other popular techniques such as spectral analysis, regeneration, and autoregressive modeling, are prescribed in Chapter 16 and standard simulation texts such as Bratley et al. (1987) and Law and Kelton (2000). Finally, Section 5 gives a summary, conclusions, and some suggestions for future research areas.

2 Motivation

In this chapter, we consider a stationary stochastic process $Y_1, Y_2, \ldots, Y_n$, e.g., a simulation in steady state. In fact, we assume that the stochastic process satisfies a Functional Central Limit Theorem (FCLT). This assumption applies to a general class of stochastic processes, and will help give us the limiting properties of the various variance estimators considered herein.

Assumption 1 (FCLT). There exist constants μ and positive σ such that as $n \to \infty$,

$$X_n \Rightarrow \sigma\mathcal{W},$$

where $\mathcal{W}$ is a standard Brownian motion process, "⇒" denotes weak convergence as $n \to \infty$ (see Billingsley, 1968) and

$$X_n(t) \equiv \frac{\lfloor nt\rfloor(\overline{Y}_{\lfloor nt\rfloor} - \mu)}{\sqrt{n}} \quad \text{for } t \geqslant 0,$$

where $\overline{Y}_j \equiv \sum_{k=1}^{j} Y_k/j$, $j = 1, 2, \ldots,$ and $\lfloor\cdot\rfloor$ is the greatest integer function.

Glynn and Iglehart (1990) list several different sets of sufficient conditions – usually in the form of moment and mixing conditions – for Assumption 1 to hold; see also Chapter 2. The constants μ and σ^2 in the assumption can be identified with the steady-state mean and variance parameters, respectively.

Let $R_k \equiv \mathrm{Cov}(Y_1, Y_{1+k})$, $k = 0, \pm 1, \pm 2, \ldots,$ denote the covariance function of the stationary stochastic process, and define the "center of gravity" quantity $\gamma \equiv -2\sum_{k=1}^{\infty} kR_k$ (cf. Song and Schmeiser, 1995). In addition, the notation $p(n) = \mathrm{o}(q(n))$ means that $p(n)/q(n) \to 0$ as $n \to \infty$. This is notation we will need later.

The rest of this chapter is devoted to motivating, defining and comparing classes of estimators of σ_n^2 and σ^2. To facilitate discussing what makes an estimator good, let $\hat{\sigma}_n^2$ and $\hat{\sigma}^2$ denote generic estimators of σ_n^2 and σ^2, respectively. In this section we focus on $\hat{\sigma}^2$ for convenience, but nearly everything applies to $\hat{\sigma}_n^2$ as well.

The most critical properties of $\hat{\sigma}^2$ are its variance and bias: $\mathrm{Var}(\hat{\sigma}^2) = \mathrm{E}\{(\hat{\sigma}^2 - \mathrm{E}[\hat{\sigma}^2])^2\}$ and $\mathrm{Bias}(\hat{\sigma}^2) = \mathrm{E}[\hat{\sigma}^2] - \sigma^2$. Clearly an estimator with low bias – which implies that its distribution is centered at the desired value σ^2 – and low variance – so that it also tends to be close to the desired value – is what we want. A single measure that combines both measures is the mean squared error

$$\mathrm{MSE}(\hat{\sigma}^2) = \mathrm{Var}(\hat{\sigma}^2) + \mathrm{Bias}^2(\hat{\sigma}^2).$$

As will become apparent when we look at different classes of estimators, there is often a bias-variance tradeoff so that doing something that decreases one of the measures increases the other. So why not simply use the variance estimator $\hat{\sigma}^2$ offering the lowest MSE? The problem is that it is not possible to derive a useful expression for the MSE of any reasonable variance estimator applied to all stationary output process that we might encounter in real life. As a result, the analysis and comparison of variance estimators typically takes one or more of the following forms:

Empirical evaluation: Variance estimators are applied to simulated processes with *known* σ^2's, allowing their bias, variance and MSE to be estimated. Since we "know the answer", we can see how the competing variance estimators compare against each other. Simple Markovian queueing models are often used for this type of comparison (e.g., Sargent et al., 1992).

Surrogate-processes evaluation: Simple output-process models, under which the bias, variance, and MSE of $\hat{\sigma}^2$ can be explicitly derived, are used as surrogates (stand-ins) for the more-general simulation output processes we might encounter. Time-series processes, such as low-order autoregressive and moving average, are often chosen for this purpose (cf. Sargent et al., 1992).

Asymptotic analysis: The bias, variance, and MSE measures of appropriately scaled versions of $\hat{\sigma}^2$ can sometimes be derived as the sample size n goes to infinity, and these limiting results are often free of the fine details of the specific simulation output process. We focus on this approach here by relying on the FCLT as our basic building block.

Unfortunately, it has never been proven, nor is it likely ever to be proven, that any one estimator has uniformly the smallest MSE across the space of all stationary output processes to which it could be applied.

The bias and variance of $\hat{\sigma}^2$ are the most critical distributional properties when determining its usefulness as an estimator. Secondarily, the distribution itself also matters, particularly when the goal is to use $\hat{\sigma}^2$ to form a confidence interval for μ. The usual normal-theory $(1-\alpha)100\%$ confidence interval for μ takes the form

$$\overline{Y}_n \pm t_{1-\alpha/2,\nu}\sqrt{\frac{\hat{\sigma}^2}{n}}, \tag{1}$$

where $t_{1-\alpha/2,\nu}$ is the $1-\alpha/2$ quantile of the t distribution with ν degrees of freedom. The validity of this confidence interval depends on $\overline{Y}_n$ being approximately normally distributed and independent of $\hat{\sigma}^2$, which has a scaled chi-squared distribution with ν degrees of freedom. Therefore, if $\hat{\sigma}^2$ is to be incorporated into a confidence-interval procedure it is important to have some assurance that its distribution is approximately chi-squared and an appropriate degrees of freedom can be associated with it. For all practical purposes this property *never* precisely holds for steady-state simulation output processes, so the that best we can hope is that it is true in some appropriate asymptotic sense (cf. Alexopoulos et al., 2005a, who present histograms of the distributions of various variance estimators).

This chapter emphasizes asymptotic analysis, which has been exceptionally useful for deriving and comparing classes of estimators. But the estimator with the best asymptotic properties is not necessarily the best estimator in practice, because "practice" can include problems with sample sizes too small for the asymptotic properties of all available estimators to hold. Certain estimators with inferior asymptotic properties have been shown to be robust in practical problems. Nevertheless, asymptotic comparisons provide a level playing field at least for initial evaluation, as well as providing hints for how to improve estimator performance.

3 Estimators using nonoverlapping batches

This section examines a number of different estimators for σ^2 resulting from nonoverlapping batches of observations. Loosely speaking, this section examines estimators arising from the following recipe:

- Divide the run into contiguous, nonoverlapping *batches*,
- form an estimator from each batch, and
- take the average of the estimators.

In particular, we discuss the NBM, STS batched area, and STS batched Cramér–von Mises (CvM) estimators for σ^2 in Sections 3.1, 3.3 and 3.4, respectively; Section 3.2 gives a short primer on standardized time series. In each case, we give results on the expected value and variance of the estimator under consideration.

Throughout the entire section we will work with b contiguous, nonoverlapping batches of observations, each of length m, from the simulation output, $Y_1, Y_2, \ldots, Y_n$, where $n = bm$. Thus, the observations $Y_{(i-1)m+1}$, $Y_{(i-1)m+2}, \ldots, Y_{im}$ constitute batch i, for $i = 1, 2, \ldots, b$.

3.1 NBM estimator

The batches of observations can be depicted as follows.

$$\begin{aligned}
&\text{Batch 1: } Y_1, Y_2, \ldots, Y_m,\\
&\text{Batch 2: } Y_{m+1}, Y_{m+2}, \ldots, Y_{2m},\\
&\quad\vdots\\
&\text{Batch } b\text{: } Y_{(b-1)m+1}, Y_{(b-1)m+2}, \ldots, Y_n.
\end{aligned}$$

For each of these batches, we calculate the *batch mean*, $\overline{Y}_{i,m} \equiv m^{-1} \times \sum_{k=1}^{m} Y_{(i-1)m+k}$, for $i = 1, 2, \ldots, b$, which is the genesis of the method's name.

The NBM estimator for μ is simply the grand sample mean from the b batch means, $\overline{Y}_n \equiv b^{-1} \sum_{i=1}^{b} \overline{Y}_{i,m} = n^{-1} \sum_{\ell=1}^{n} Y_\ell$. Stationarity implies that $\mathrm{E}[\overline{Y}_n] = \mu$, so the grand mean is unbiased for μ; and the variance of the grand mean is, by definition, $\mathrm{Var}(\overline{Y}_n) = \sigma_n^2/n$.

The batch means $\overline{Y}_{i,m}$, $i = 1, 2, \ldots, b$, are often assumed to be i.i.d. normal random variables, at least for large enough batch size m; this is borne out by Equation (8) below. The i.i.d. assumption immediately suggests the NBM estimator for σ^2,

$$\mathcal{N}(b, m) \equiv \frac{m}{b-1} \sum_{i=1}^{b} (\overline{Y}_{i,m} - \overline{Y}_n)^2 \xrightarrow{\mathcal{D}} \frac{\sigma^2 \chi^2_{b-1}}{b-1} \tag{2}$$

as $m \to \infty$ with b fixed, and where χ^2_ν denotes a chi-squared random variable with ν degrees of freedom and "$\xrightarrow{\mathcal{D}}$" denotes convergence in distribution as $m \to \infty$ (see, e.g., Glynn and Whitt, 1991; Schmeiser, 1982; Steiger and Wilson, 2001). The NBM estimator, which is the sample variance of the batch means, is one of the most popular for σ^2, and is a benchmark for comparison with other estimators.

Under mild conditions, Chien et al. (1997), Goldsman and Meketon (1986) and Song and Schmeiser (1995) show that

$$\mathrm{E}\bigl[\mathcal{N}(b, m)\bigr] = \sigma^2 + \frac{\gamma(b+1)}{bm} + \mathrm{o}\left(\frac{1}{m}\right). \tag{3}$$

So as the batch size m increases, the bias of $\mathcal{N}(b, m)$ as an estimator of σ^2 goes to zero.

How does one prove a result such as Equation (3) for the expected value? It amounts to careful bookkeeping of covariance terms. First of all, assuming that the underlying process $\{Y_i\}$ is stationary and that all of the following sums are well defined, we have

$$\begin{aligned}
\sigma_m^2 &\equiv m\,\mathrm{Var}(\overline{Y}_m)\\
&= \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m}\mathrm{Cov}(Y_i, Y_j)\\
&= R_0 + 2\sum_{i=1}^{m-1}\left(1-\frac{i}{m}\right)R_i \quad \text{(after collecting like covariance terms)}\\
&= R_0 + 2\sum_{i=1}^{m-1}R_i - \frac{2}{m}\sum_{i=1}^{m-1}iR_i\\
&= R_0 + 2\sum_{i=1}^{\infty}R_i - 2\sum_{i=m}^{\infty}R_i - \frac{2}{m}\sum_{i=1}^{\infty}iR_i + \frac{2}{m}\sum_{i=m}^{\infty}iR_i\\
&= \sigma^2 + \frac{\gamma}{m} - 2\sum_{i=m}^{\infty}\left(1-\frac{i}{m}\right)R_i\\
&= \sigma^2 + \frac{\gamma}{m} + \mathrm{o}\left(\frac{1}{m}\right), \qquad (4)
\end{aligned}$$

where, as in Chien et al. (1997), we have implicitly assumed that $\sum_{j=1}^{\infty} j|R_j| < \infty$. Turning to the matter at hand, we have

$$\begin{aligned}
\mathrm{E}\big[\mathcal{N}(b,m)\big] &= \mathrm{E}\left[\frac{m}{b-1}\sum_{i=1}^{b}(\overline{Y}_{i,m}-\overline{Y}_n)^2\right]\\
&= \frac{m}{b-1}\mathrm{E}\left[\sum_{i=1}^{b}\overline{Y}_{i,m}^2 - b\overline{Y}_n^2\right]\\
&= \frac{m}{b-1}\left[\sum_{i=1}^{b}\mathrm{E}\big[\overline{Y}_{i,m}^2\big] - b\mathrm{E}\big[\overline{Y}_n^2\big]\right]\\
&= \frac{bm}{b-1}\big[\mathrm{E}\big[\overline{Y}_{i,m}^2\big] - \mathrm{E}\big[\overline{Y}_n^2\big]\big]\\
&\qquad \text{(by stationarity of the batch means)}\\
&= \frac{bm}{b-1}\big[\mathrm{Var}(\overline{Y}_{i,m}) - \mathrm{Var}(\overline{Y}_n)\big]\\
&\qquad \text{(since the common mean } \mu \text{ cancels)}\\
&= \frac{1}{b-1}\left[b\sigma^2 + \frac{b\gamma}{m} - \sigma^2 - \frac{\gamma}{n}\right] + \mathrm{o}\left(\frac{1}{m}\right)\\
&\qquad \text{(by Equation (4))},
\end{aligned}$$

which gives (3) after a little algebra.

Obtaining the NBM estimator's variance takes more work. Chien et al. (1997), Glynn and Whitt (1991), Goldsman and Meketon (1986) and Song and Schmeiser (1995) find that, for fixed b,

$$\lim_{m\to\infty}(b-1)\operatorname{Var}\bigl(\mathcal{N}(b,m)\bigr)=2\sigma^4 \tag{5}$$

which certainly makes sense in light of the distributional result given by Equation (2).

By combining Equations (3) and (5), we can obtain a simple expression for the mean squared error of the NBM estimator for σ^2,

$$\mathrm{MSE}\bigl[\mathcal{N}(b,m)\bigr]\doteq\left(\frac{\gamma(b+1)}{mb}\right)^2+\frac{2\sigma^4}{b}, \tag{6}$$

where we have ignored some small-order terms. For large b and m, this expression can be minimized by taking the number of batches $b=cn^{2/3}$ and the batch size $m=n/b=n^{1/3}/c$, where we need to determine the constant c. Thus, expression (6) becomes

$$\mathrm{MSE}\bigl[\mathcal{N}(b,m)\bigr]\doteq\left(c^2\gamma^2+\frac{2\sigma^4}{c}\right)\frac{1}{n^{2/3}}.$$

Minimizing this expression with respect to c, we obtain $c^\star=(\sigma^4/\gamma^2)^{1/3}$, and hence an (asymptotically) optimal mean squared error of

$$\mathrm{MSE}^\star\bigl[\mathcal{N}(b,m)\bigr]\doteq 3\left(\frac{\sigma^4\gamma}{n}\right)^{2/3}. \tag{7}$$

Of course, σ^2 and γ are not known in general, so that it will be problematic to actually obtain this minimal MSE in practice; however, Equation (7) still has practical value, since it can be used as a basis for comparison among other variance estimators. For instance, if another estimator has a minimal MSE of the same form as (7), but with a leading constant smaller than 3, then one can claim that it is a "better" estimator than NBM.

3.2 STS primer

Before presenting additional estimators for σ^2, we give a mini-tutorial on standardized time series, which will provide the necessary background. Schruben (1983) defines the *standardized time series* from batch i as

$$T_{i,m}(t)\equiv\frac{\lfloor mt\rfloor(\overline{Y}_{i,\lfloor mt\rfloor}-\overline{Y}_{i,m})}{\sigma\sqrt{m}}$$

for $0\leqslant t\leqslant 1$ and $i=1,2,\ldots,b$, where

$$\overline{Y}_{i,j}\equiv\frac{1}{j}\sum_{k=1}^{j}Y_{(i-1)m+k}$$

for $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, b$. Then we have the following theorem.

Theorem 1 (see Schruben (1983), Glynn and Iglehart (1990), Foley and Goldsman (1999), or Alexopoulos et al. (2005b), among others). *Define* $Z_i(m) \equiv \sqrt{m}(\overline{Y}_{i,m} - \mu)$, $i = 1, 2, \ldots, b$. *Then under Assumption* 1,

$$\begin{aligned}&\bigl(Z_1(m), \ldots, Z_b(m); \sigma T_{1,m}, \ldots, \sigma T_{b,m}\bigr)\\&\quad\Rightarrow \bigl(\sigma Z_1, \ldots, \sigma Z_b; \sigma \mathcal{B}_0, \ldots, \sigma \mathcal{B}_{b-1}\bigr),\end{aligned}\tag{8}$$

where the Z_i*'s are i.i.d. standard normal random variables, and* $\mathcal{B}_u$ *denotes a Brownian bridge process on* $[u, u+1]$, *i.e., for* $t \in [0, 1]$,

$$\mathcal{B}_u(t) = \mathcal{W}(u+t) - \mathcal{W}(u) - t\bigl[\mathcal{W}(u+1) - \mathcal{W}(u)\bigr].$$

Notice that the scaling factor σ actually cancels out in the denominator of $T_{i,m}$, thus eliminating any unknowns and so allowing us to explicitly calculate all of the quantities on the left-hand side of Equation (8). Further, for those unfamiliar with Brownian bridges, it turns out that all finite-dimensional joint distributions of a Brownian bridge $\mathcal{B}_u$ are normal with $\mathrm{E}[\mathcal{B}_u(t)] = 0$ and $\mathrm{Cov}(\mathcal{B}_u(s), \mathcal{B}_u(t)) = \min(s, t) - st$, $0 < s, t < 1$. In addition, since Brownian motion has independent increments, it is easy to see that $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_{b-1}$ are independent Brownian bridges. These facts will allow us to evaluate properties of the random variables on the left-hand side of (8) using the analogous asymptotic distributions from the right-hand side of the equation.

3.3 Batched area estimator

This subsection deals with the (nonoverlapping) batched area estimator for σ^2 (Goldsman et al., 1990; Goldsman and Schruben, 1990).

We will work with the square of the weighted area under the standardized time series from the ith batch,

$$A_i(f; m) \equiv \left[\frac{1}{m}\sum_{k=1}^{m} f\left(\frac{k}{m}\right)\sigma T_{i,m}\left(\frac{k}{m}\right)\right]^2,$$

and its limiting functional

$$A_i(f) \equiv \left[\int_0^1 f(t)\sigma \mathcal{B}_{i-1}(t)\,\mathrm{d}t\right]^2,$$

for $i = 1, 2, \ldots, b$, where $f(t)$ is continuous on the interval $[0, 1]$ and normalized so that $\mathrm{Var}(\int_0^1 f(t)\mathcal{B}_0(t)\,\mathrm{d}t) = 1$. Under mild conditions (see Alexopoulos et al., 2005b), one can use the continuous mapping theorem (see Billingsley, 1968) to show that $A_i(f; m) \xrightarrow{\mathcal{D}} A_i(f)$, $i = 1, 2, \ldots, b$; and further, $A_1(f), \ldots, A_b(f)$ are i.i.d. $\sigma^2\chi_1^2$. This result motivates construction of the *batched area estimator* for σ^2,

$$\mathcal{A}(f;b,m) \equiv \frac{1}{b}\sum_{i=1}^{b} A_i(f;m). \tag{9}$$

As the batch size $m \to \infty$ with a fixed number of batches b, the distribution of the batched area estimator converges to that of the average of the corresponding $A_i(f)$ functionals, i.e.,

$$\mathcal{A}(f;b,m) \xrightarrow{\mathcal{D}} \mathcal{A}(f;b) \equiv \frac{1}{b}\sum_{i=1}^{b} A_i(f) \sim \sigma^2 \frac{\chi_b^2}{b}.$$

The next theorem gives the expected value and variance of the area estimator.

Theorem 2 (see, e.g., Foley and Goldsman, 1999). *Suppose that* $\{Y_i, i \geqslant 1\}$ *is a stationary process for which Assumption 1 holds,* $\sum_{k=1}^{\infty} k^2|R_k| < \infty$, *and* $\sigma^2 > 0$. *Further, suppose that* $\mathcal{A}^2(f;b,m)$ *is uniformly integrable* (*cf.* Billingsley, 1968). *If we define the quantities* $F \equiv \int_0^1 f(t)\,\mathrm{d}t$, $\overline{F} \equiv \int_0^1 \int_0^t f(s)\,\mathrm{d}s\,\mathrm{d}t$ *and* $F^\star \equiv [(F-\overline{F})^2 + \overline{F}^2]/2$, *then*

$$\mathrm{E}\big[\mathcal{A}(f;b,m)\big] = \sigma^2 + \frac{F^\star\gamma}{m} + \mathrm{o}\left(\frac{1}{m}\right)$$

and

$$\mathrm{Var}\big(\mathcal{A}(f;b,m)\big) \to \mathrm{Var}\left(\sigma^2\frac{\chi_b^2}{b}\right) = 2\frac{\sigma^4}{b}$$

as $m \to \infty$. *Note that the limiting variance does not depend on the choice of the weighting function (as long as it is legal).*

Example 1. Schruben (1983) first considered the area estimator with constant weighting function $f_0(t) \equiv \sqrt{12}$ for $0 \leqslant t \leqslant 1$. For this choice, Theorem 2 implies that $\mathrm{E}[\mathcal{A}(f_0;b,m)] = \sigma^2 + 3\gamma/m + \mathrm{o}(1/m)$.

Example 2. If the selection of $f(t)$ turns out to give $F = \overline{F} = 0$, the resulting estimator is *first-order unbiased* for σ^2, i.e., its bias is $\mathrm{o}(1/m)$. An example of such a weighting function is the quadratic $f_2(t) \equiv \sqrt{840}(3t^2 - 3t + 1/2)$ (Goldsman et al., 1990; Goldsman and Schruben, 1990).

Example 3. Foley and Goldsman (1999) give an "orthonormal" sequence of first-order unbiased weights, $f_{\cos,j}(t) = \sqrt{8}\pi j\cos(2\pi jt)$, $j = 1, 2, \ldots$. It can be shown that the orthonormal estimators' limiting functionals $A_i(f_{\cos,1})$, $A_i(f_{\cos,2}), \ldots$ are i.i.d. $\sigma^2\chi_1^2$. Thus, we have a new estimator for σ^2, since as the batch size m becomes large,

$$\frac{1}{k}\sum_{j=1}^{k} A_i(f_{\cos,j}; m) \xrightarrow{\mathcal{D}} \sigma^2 \frac{\chi_k^2}{k}, \tag{10}$$

for some "reasonable" number of orthonormal weights k. This estimator is actually the average of different weighted area estimators from the *same batch*, whereas the estimator given in Equation (9) uses observations from *all batches*. Of course, one could also average estimators of the form given in Equation (10) over all batches to obtain even more degrees of freedom, but that estimator's performance properties have not yet been thoroughly evaluated.

Remark 1. Since the Z_i's and $\mathcal{B}_i$'s in Equation (8) turn out to be uncorrelated normal random variables, the standardized sample mean is asymptotically (as $m \to \infty$) independent of the standardized time series (Schruben, 1983). One can use this fact to argue that STS estimators such as $\mathcal{A}(f; b, m)$ are asymptotically independent of the NBM estimator $\mathcal{N}(b, m)$; and then we can take an appropriate linear combination of the STS and NBM estimators, e.g.,

$$\frac{b\mathcal{A}(f; b, m) + (b-1)\mathcal{N}(b, m)}{2b-1},$$

with the intent that the combined estimator will have more degrees of freedom – and so reduce variance without affecting bias too much.

3.4 Batched CvM estimator

This subsection discusses the weighted Cramér–von Mises estimator for σ^2 (see Goldsman et al., 1999). To begin with, the weighted area under the square of the STS from the ith batch and its limiting functional are given by

$$C_i(g; m) \equiv \frac{1}{m}\sum_{k=1}^{m} g\left(\frac{k}{m}\right)\sigma^2 T_{i,m}^2\left(\frac{k}{m}\right)$$

and

$$C_i(g) \equiv \int_0^1 g(t)\sigma^2 \mathcal{B}_{i-1}^2(t)\,\mathrm{d}t,$$

respectively. Here $g(t)$ is a weighting function normalized so that $\mathrm{E}[C_i(g)] = \sigma^2$ and possessing a continuous and bounded second derivative on [0, 1].

Under mild assumptions, the continuous mapping theorem implies that $C_i(g; m) \xrightarrow{\mathcal{D}} C_i(g)$, $i = 1, 2, \ldots, b$. This leads to the *batched CvM* estimator for σ^2

$$\mathcal{C}(g; b, m) \equiv \frac{1}{b}\sum_{i=1}^{b} C_i(g; m).$$

As the batch size $m \to \infty$ with a fixed number of batches b, the distribution of the batched CvM estimator converges to that of the average of the corresponding $C_i(g)$ functionals, i.e.,

$$\mathcal{C}(g; b, m) \xrightarrow{\mathcal{D}} \mathcal{C}(g; b) \equiv \frac{1}{b}\sum_{i=1}^{b} C_i(g).$$

An immediate question to ask is: Why should we bother with yet another estimator $\mathcal{C}(g; b, m)$ for σ^2? Theorem 3 prepares us for an answer by presenting results on the expected value and variance of the weighted CvM estimator.

Theorem 3 (Goldsman et al., 1999). *Define* $G \equiv \int_0^1 g(t)\,\mathrm{d}t$. *Under conditions similar to those of Theorem 2,*

$$\mathrm{E}\big[\mathcal{C}(g; b, m)\big] = \sigma^2 + \frac{\gamma}{m}(G - 1) + \mathrm{o}\left(\frac{1}{m}\right)$$

and, for fixed b,

$$\begin{aligned}\lim_{m\to\infty} b\,\mathrm{Var}\big(\mathcal{C}(g; b, m)\big) &= b\,\mathrm{Var}\big(\mathcal{C}(g; b)\big)\\ &= \mathrm{Var}\big(C_1(g)\big)\\ &= 4\sigma^4 \int_0^1 g(t)(1-t)^2 \int_0^t g(s)s^2\,\mathrm{d}s\,\mathrm{d}t. \qquad (11)\end{aligned}$$

Example 4. Consider the constant weighting function $g_0(t) \equiv 6$ for $0 \leqslant t \leqslant 1$. Theorem 3 shows that the resulting CvM estimator has expected value $\mathrm{E}[\mathcal{C}(g_0; b, m)] = \sigma^2 + 5\gamma/m + \mathrm{o}(1/m)$.

Example 5. Suppose we select a weighting function having $G = 1$ (in addition to the normalizing and derivative constraints). Then the theorem implies that the CvM estimator $\mathcal{C}(g; b, m)$ has bias $\mathrm{o}(1/m)$. For example, the quadratic weighting function $g_2^\star(t) \equiv -24+150t-150t^2$ has this first-order unbiasedness property.

Now, back to the question concerning our interest in yet another estimator for σ^2. We see from Theorem 3 that the choice of weighting function $g(t)$ affects the variances of $\mathcal{C}(g; b, m)$ and $\mathcal{C}(g; b)$. This was not the case for the area estimator of Section 3.3, where the weighting function $f(t)$ affects the variance of $\mathcal{A}(f; b, m)$, but not that of the limiting functional $\mathcal{A}(f; b)$, which is always $\mathrm{Var}(\mathcal{A}(f; b)) = 2\sigma^4/b$. Thus, the CvM estimator gives us a tool to find reduced-variance estimators for σ^2.

Example 6. Theorem 3 shows that $\mathrm{Var}(\mathcal{C}(g_0; b)) = 0.8\sigma^4/b$ and $\mathrm{Var}(\mathcal{C}(g_2^\star; b)) = 1.73\sigma^4/b$, which are both smaller than the limiting ($m \to \infty$) variance of any batched area estimator, as discussed immediately above. Even though $\mathrm{Var}(\mathcal{C}(g_2^\star; b)) > \mathrm{Var}(\mathcal{C}(g_0; b))$, the estimator $\mathcal{C}(g_2^\star; b, m)$ is first-order unbiased for σ^2, while $\mathcal{C}(g_0; b, m)$ does not have this nice property.

3.5 Comparison

We have seen that as the batch size $m \to \infty$, the NBM, batched area, and batched CvM estimators are all asymptotically unbiased for σ^2. Furthermore, the variances of these estimators are all more-or-less inversely proportional to the number of batches – though one has to be careful not to decrease the variance too much at the expense of relatively high bias or MSE (cf. Song and Schmeiser, 1995).

What happens if m and b *both* become large? Such a scenario might occur in sequential estimation procedures – for example, Steiger et al.'s (2005) ASAP3 procedure – which occasionally take additional observations to achieve user-specified estimator precision requirements. As m and $b \to \infty$, it has been shown in, e.g., Alexopoulos et al. (2000) and Chien et al. (1997) that the estimators under discussion herein are consistent in mean square.

Since NBM is regarded as the benchmark method, one could informally regard its bias and variance as straw men. The interesting result is that STS area and CvM estimators with certain well-chosen weighting functions can beat NBM in terms of large-sample bias; in addition, the CvM estimators studied in this article have smaller variance than does NBM. Better yet, we will see in the next section that the use of overlapping batches with respect to any particular estimator preserves its expected value, while reducing its variance – sometimes substantially. See Table 1 for an early preview.

Obviously, for fixed sample size $n = mb$, some estimators will tend to do better than others in terms of bias and variance. But for fixed n, decreasing one performance measure usually comes at the expense of increasing the other – the well-known trade-off that we have already mentioned. Thus, we ought to be interested in small-sample performance of the various estimators in addition to asymptotic performance. For various "toy" processes, it is actually possible to calculate exact results for finite sample sizes.

Example 7. Suppose that $Y_1, Y_2, \ldots, Y_n$ arise from a first-order moving average (MA(1)) process, that is, $Y_{i+1} = \theta\varepsilon_i + \varepsilon_{i+1}$, where the ε_i's are i.i.d. standard normal and $-1 < \theta < 1$. The MA(1) has covariance function $R_0 = 1 + \theta^2$, $R_{\pm 1} = \theta$, and $R_k = 0$ elsewhere, from which we can easily derive $\sigma^2 = \sum_{j=-\infty}^{\infty} R_j = (1 + \theta)^2$ and $\gamma = -2\sum_{j=1}^{\infty} jR_j = -2\theta$.

After some tedious algebra (see, e.g., Goldsman et al., 2003), we have the following *exact* results:

$$\mathrm{E}\big[\mathcal{N}(b,m)\big] = \sigma^2 + \frac{\gamma(b+1)}{mb},$$
$$\mathrm{E}\big[\mathcal{A}(f_2;b,m)\big] = \sigma^2 + \frac{7(\sigma^2+6\gamma)}{2m^2} + \mathrm{o}\big(m^{-2}\big),$$
$$\mathrm{E}\big[\mathcal{C}\big(g_2^\star;b,m\big)\big] = \sigma^2 + \frac{4(\sigma^2+6\gamma)}{m^2} + \mathrm{o}\big(m^{-2}\big).$$

We see that these particular weighted area and CvM estimators are less biased than NBM as the batch size m becomes large; and the area estimator beats CvM for moderate m.

4 Estimators from overlapping batches

We now discuss the use of estimators based on *overlapping* batches, à la Meketon and Schmeiser (1984). Here we implement a slightly different recipe than that for nonoverlapping batches:

- Divide the run into a number of overlapping batches,
- form an estimator from each batch, and
- take the average of the estimators.

The presentation roughly follows that of the previous section. Section 4.1 discusses some necessary basics and provides the notation that we will use. Sections 4.2–4.4 give the OBM, STS overlapping area, and STS overlapping CvM estimators, respectively. As before, we are concerned with results on the expected values and variances of the various estimators.

4.1 Overlapping fundamentals

Suppose we have n observations $Y_1, Y_2, \ldots, Y_n$ on hand and that we form $n-m+1$ overlapping batches, each of size m. The overlapping batches can be depicted as follows.

$$\begin{array}{ll}
\text{Overlapping batch 1:} & Y_1, Y_2, \ldots, Y_m, \\
\text{Overlapping batch 2:} & Y_2, Y_3, \ldots, Y_{m+1}, \\
\quad\vdots & \\
\text{Overlapping batch } n-m+1\text{:} & Y_{n-m+1}, Y_{n-m+2}, \ldots, Y_n
\end{array}$$

Specifically, the observations $Y_i, Y_{i+1}, \ldots, Y_{i+m-1}$ comprise batch i, $i = 1, 2, \ldots, n-m+1$. Further, we use the notation $b \equiv n/m$ as before, though b is no longer "the number of batches".

As in Section 3.2, the standardized time series from overlapping batch i is

$$T^{\mathrm{O}}_{i,m}(t) \equiv \frac{\lfloor mt \rfloor (\overline{Y}^{\mathrm{O}}_{i,\lfloor mt \rfloor} - \overline{Y}^{\mathrm{O}}_{i,m})}{\sigma\sqrt{m}}$$

for $0 \leqslant t \leqslant 1$ and $i = 1, 2, \ldots, n-m+1$, where

$$\overline{Y}^{\mathrm{O}}_{i,j} \equiv \frac{1}{j}\sum_{k=0}^{j-1} Y_{i+k}$$

for $i = 1, 2, \ldots, n-m+1$ and $j = 1, 2, \ldots, m$. Under the same mild conditions as before,

$$\sigma T^{\mathrm{O}}_{\lfloor um \rfloor, m} \Rightarrow \sigma \mathcal{B}_u, \quad 0 \leqslant u \leqslant b-1, \ u \text{ fixed.}$$

Roughly speaking, we will try to glean more information about σ^2 by considering the variance estimators from each of the $n-m+1$ overlapping batches rather than from just the b nonoverlapping batches of the earlier discussion. The hope is that the additional information will somehow compensate for the fact that estimators arising from overlapping batches might be highly correlated.

4.2 OBM estimator

The ith overlapping batch mean is given by $\overline{Y}^{\mathrm{O}}_{i,m}$, $i = 1, 2, \ldots, n-m+1$. The OBM estimator for σ^2 was originally studied by Meketon and Schmeiser (1984) (using a slightly different scaling coefficient), and is given by

$$\mathcal{O}(b,m) \equiv \frac{nm}{(n-m+1)(n-m)} \sum_{i=1}^{n-m+1} \left(\overline{Y}^{\mathrm{O}}_{i,m} - \overline{Y}_n\right)^2.$$

Theorem 4. *Under mild conditions*, Goldsman and Meketon (1986) *and* Song and Schmeiser (1995) *show that*, *for large* b,

$$\mathrm{E}\big[\mathcal{O}(b,m)\big] \doteq \sigma^2 + \frac{\gamma}{m} + \mathrm{o}\left(\frac{1}{m}\right).$$

Further, Meketon and Schmeiser (1984), Damerdji (1995) *and* Alexopoulos et al. (2005b) *find that for large* b, *as* $m \to \infty$,

$$\mathrm{Var}\big(\mathcal{O}(b,m)\big) \doteq \frac{4\sigma^4}{3b}.$$

4.3 Overlapping area estimator

The square of the weighted area under the standardized time series from the ith overlapping batch is

$$A_i^{\mathrm{O}}(f;m) \equiv \left[\frac{1}{m}\sum_{k=1}^{m} f\left(\frac{k}{m}\right)\sigma T_{i,m}^{\mathrm{O}}\left(\frac{k}{m}\right)\right]^2,$$

$i = 1, 2, \dots, n-m+1$. The *overlapping area estimator* for σ^2 is

$$\mathcal{A}^{\mathrm{O}}(f;b,m) \equiv \frac{1}{n-m+1}\sum_{i=1}^{n-m+1} A_i^{\mathrm{O}}(f;m).$$

Alexopoulos et al. (2005b) use the continuous mapping theorem to show that as $m \to \infty$,

$$\mathcal{A}^{\mathrm{O}}(f;b,m) \xrightarrow{\mathcal{D}} \mathcal{A}^{\mathrm{O}}(f;b) \equiv \frac{\sigma^2}{b-1}\int_0^{b-1}\left[\int_0^1 f(t)\mathcal{B}_u(t)\,\mathrm{d}t\right]^2 \mathrm{d}u. \tag{12}$$

It is easy to see that the expected value of the overlapping area estimator equals that of the corresponding batched area estimator. Thus, Theorem 2 gives the following.

Theorem 5. *Under mild conditions*, Alexopoulos et al. (2005a, 2005b) *show that*

$$\mathrm{E}\big[\mathcal{A}^{\mathrm{O}}(f;b,m)\big] = \sigma^2 + \frac{F^\star\gamma}{m} + \mathrm{o}\left(\frac{1}{m}\right).$$

Calculation of the variance of the overlapping area estimator can be undertaken using the right-hand side of Equation (12) along with some algebraic elbow grease. Some examples from Alexopoulos et al. (2005b) reveal that the limiting ($m \to \infty$) variance of the overlapping area estimator depends on the choice of weighting function.

Example 8. Consider the overlapping constant-weighted area estimator from Example 1. We have after some algebra that as $m \to \infty$,

$$\mathrm{Var}\big(\mathcal{A}^{\mathrm{O}}(f_0;b,m)\big) \to \mathrm{Var}\big(\mathcal{A}^{\mathrm{O}}(f_0;b)\big) = \frac{24b-31}{35(b-1)^2}\sigma^4.$$

This compares very nicely to the generic batched area estimator's asymptotic ($m \to \infty$) variance, $\mathrm{Var}(\mathcal{A}(f;b)) = 2\sigma^4/b$ (see Theorem 2).

Example 9. Consider the overlapping area estimator with first-order unbiased quadratic weighting function $f_2(t)$ from Example 2. This estimator has a limiting variance of

$$\operatorname{Var}\bigl(\mathcal{A}^{\mathrm{O}}(f_2; b)\bigr) = \frac{3514b - 4359}{4290(b-1)^2}\sigma^4.$$

Example 10. Consider the overlapping area estimators from the family of orthonormal first-order unbiased weights $f_{\cos,j}(t)$, $j = 1, 2, \ldots$. From Example 3, we have a limiting variance of about

$$\operatorname{Var}\bigl(\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b)\bigr) \doteq \frac{8\pi^2 j^2 + 15}{12\pi^2 j^2 b}\sigma^4.$$

Remark 2. One can average the orthonormal estimators $\mathcal{A}^{\mathrm{O}}(f_{\cos,j}; b, m)$, $j = 1, 2, \ldots$, and use knowledge of the covariances of these estimators to obtain estimators with even smaller variance (cf. Alexopoulos et al., 2005b).

4.4 Overlapping CvM estimator

We define the overlapping CvM estimator from overlapping batch i by

$$C_i^{\mathrm{O}}(g; m) \equiv \frac{1}{m}\sum_{k=1}^{m} g\left(\frac{k}{m}\right)\left[\sigma T_{i,m}^{\mathrm{O}}\left(\frac{k}{m}\right)\right]^2,$$

$i = 1, 2, \ldots, n - m + 1$. The *overlapping CvM* estimator for σ^2 is

$$\mathcal{C}^{\mathrm{O}}(g; b, m) \equiv \frac{1}{n - m + 1}\sum_{i=1}^{n-m+1} C_i^{\mathrm{O}}(g; m).$$

Then it can be shown using the continuous mapping theorem that as $m \to \infty$,

$$\mathcal{C}^{\mathrm{O}}(g; b, m) \xrightarrow{\mathcal{D}} \mathcal{C}^{\mathrm{O}}(g; b) \equiv \frac{1}{b-1}\int_0^{b-1}\int_0^1 g(u)\sigma^2 \mathcal{B}_s^2(u)\,\mathrm{d}u\,\mathrm{d}s. \tag{13}$$

Meanwhile, Theorem 3 implies

$$\mathrm{E}\bigl[\mathcal{C}^{\mathrm{O}}(g; b, m)\bigr] = \sigma^2 + \frac{\gamma(G-1)}{m} + \mathrm{o}\left(\frac{1}{m}\right).$$

So the expected value of the overlapping CvM estimator is the same as that of the batched CvM estimator.

Let us turn to the variance of the overlapping CvM estimator. After a great deal of algebra involving Equation (13), we have the following results.

Example 11. For the overlapping constant-weighted CvM estimator, we have

$$\mathrm{Var}\big(\mathcal{C}^{\mathrm{O}}(g_0; b, m)\big) \to \mathrm{Var}\big(\mathcal{C}^{\mathrm{O}}(g_0; b)\big) = \frac{88b - 115}{210(b-1)^2}\sigma^4.$$

This compares nicely to the batched constant-weighted CvM estimator's asymptotic ($m \to \infty$) variance, $\mathrm{Var}(\mathcal{C}(g_0; b)) = 4\sigma^4/(5b)$.

Example 12. For the overlapping CvM estimator with quadratic weight $g_2^\star(t)$ from Example 5, we have

$$\mathrm{Var}\big(\mathcal{C}^{\mathrm{O}}(g_2^\star; b)\big) = \frac{10768b - 13605}{13860(b-1)^2}\sigma^4 \doteq \frac{0.777}{b}\sigma^4.$$

This compares to the batched quadratic CvM estimator's asymptotic variance, $\mathrm{Var}(\mathcal{C}(g_2^\star; b)) = 121\sigma^4/(70b)$. In other words, $\mathrm{Var}(\mathcal{C}^{\mathrm{O}}(g_2^\star; b))/\mathrm{Var}(\mathcal{C}(g_2^\star; b)) \doteq 0.450$.

Example 13. We can even work with the quartic weighting function

$$g_4^\star(t) \equiv -\frac{1310}{21} + \frac{19270t}{21} - \frac{25230t^2}{7} + \frac{16120t^3}{3} - \frac{8060t^4}{3}.$$

Goldsman et al. (1999) show that this weighting function yields the first-order unbiased CvM estimator for σ^2 having the minimum variance over all quartic weights. After still more algebra involving Equation (13), we find that the corresponding overlapping quartic CvM estimator has

$$\mathrm{Var}\big(\mathcal{C}^{\mathrm{O}}(g_4^\star; b)\big) \doteq \frac{0.477}{b}\sigma^4,$$

which is quite competitive compared to the other estimators examined so far.

4.5 Comparison

Paralleling the discussion in Section 3.5, we see that as $m \to \infty$, the overlapping area, overlapping CvM, and OBM estimators are all asymptotically unbiased for σ^2. In addition, the variances of these estimators are all inversely proportional to the ratio $b = n/m$ (for sufficiently large batch size).

All of the overlapping estimators preserve the bias properties of their nonoverlapping counterparts. Thus, we found that the overlapping area and overlapping CvM estimators with certain "unbiased" weighting functions can beat OBM in terms of large-sample bias. The overlapping STS estimators also defeat their nonoverlapped counterparts as well as OBM in terms of variance. Table 1, abstracted from Alexopoulos et al. (2005a, 2005b), gives a more-complete synopsis of all of the asymptotic results.

We can carry out small-sample analysis for the various overlapping estimators, similar to what we did in Example 7.

Table 1.
Approximate asymptotic bias and variance for different estimators (abstracted from Alexopoulos et al., 2005a, 2005b)

Nonoverlapping	(m/γ) Bias	(b/σ^4) Var	Overlapping	(m/γ) Bias	(b/σ^4) Var
$\mathcal{A}(f;b,m)$	$F^\star$	2	$\mathcal{A}^{\mathrm{O}}(f;b,m)$	$F^\star$	various
$\mathcal{A}(f_0;b,m)$	3	2	$\mathcal{A}^{\mathrm{O}}(f_0;b,m)$	3	0.686
$\mathcal{A}(f_2;b,m)$	o(1)	2	$\mathcal{A}^{\mathrm{O}}(f_2;b,m)$	o(1)	0.819
$\mathcal{A}(f_{\cos,j};b,m)$	o(1)	2	$\mathcal{A}^{\mathrm{O}}(f_{\cos,j};b,m)$	o(1)	$(8\pi^2 j^2+15)/(12\pi^2 j^2)$
$\mathcal{C}(g;b,m)$	$G-1$	Eq. (11)	$\mathcal{C}^{\mathrm{O}}(g;b,m)$	$G-1$	various
$\mathcal{C}(g_0;b,m)$	5	0.8	$\mathcal{C}^{\mathrm{O}}(g_0;b,m)$	5	0.419
$\mathcal{C}(g_2^\star;b,m)$	o(1)	1.729	$\mathcal{C}^{\mathrm{O}}(g_2^\star;b,m)$	o(1)	0.777
$\mathcal{C}(g_4^\star;b,m)$	o(1)	1.042	$\mathcal{C}^{\mathrm{O}}(g_4^\star;b,m)$	o(1)	0.477
$\mathcal{N}(b,m)$	1	2	$\mathcal{O}(b,m)$	1	1.333

Example 14. Suppose that $Y_1, Y_2, \ldots, Y_n$ arise from the MA(1) process described in Example 7. Goldsman et al. (2003) derive the following *exact* results:

$$\mathrm{E}\big[\mathcal{O}(b,m)\big] = \sigma^2 + \frac{\gamma}{m(b-1)}\left[\frac{b^2+1}{b} - \frac{2}{mb-m+1}\right],$$

$$\mathrm{E}\big[\mathcal{A}^{\mathrm{O}}(f_2;b,m)\big] = \mathrm{E}\big[\mathcal{A}(f_2;b,m)\big] = \sigma^2 + \frac{7(\sigma^2+6\gamma)}{2m^2} + \mathrm{o}\big(m^{-2}\big)$$

and

$$\mathrm{E}\big[\mathcal{C}^{\mathrm{O}}\big(g_2^\star;b,m\big)\big] = \mathrm{E}\big[\mathcal{C}\big(g_2^\star;b,m\big)\big] = \sigma^2 + \frac{4(\sigma^2+6\gamma)}{m^2} + \mathrm{o}\big(m^{-2}\big).$$

First of all, we see that the bias results here reflect their counterparts from Example 7. Like that example, the overlapping area and CvM estimators are less biased than OBM as the batch size m becomes large; and the overlapping area estimator beats overlapping CvM for moderate m.

5 Summary and conclusions

This chapter has focused on the need to estimate the variance parameter σ^2 to give users information about the precision of the sample mean $\overline{Y}_n$ as an estimator for the steady-state mean μ. We presented a small selection of ways to go about estimating σ^2, illustrating various general tricks of the trade on certain variance estimators of interest along the way.

Certainly, mean squared error is an excellent criterion for comparing the performance of different estimators; but we could have also evaluated variance estimator performance when used within confidence interval estimators of the general form given by Equation (1) – it usually turns out that good point estimators yield good confidence intervals (cf. Sargent et al., 1992).

In terms of specifics, we found that

- overlapping estimators for σ^2 always seem to preserve bias and decrease variance vs. their nonoverlapping counterparts – sometimes by a great deal, especially in the case of the STS estimators;
- overlapping performs as advertised on the empirical MA(1) example.

So even though the overlapping estimators can be quite correlated from batch to batch, their overall performance is superior to that of the asymptotically i.i.d. estimators from nonoverlapping batches.

In the future, we can envision a focus on other variance estimators that reuse data over and over again, in the spirit of overlapping estimators. Examples of this ilk that immediately come to mind are the "orthonormal" estimators from Foley and Goldsman (1999) and the "permuted" estimators studied by Calvin and Nakayama (2002). Such estimators are apt to be computationally more complicated, but this should not present a problem, as a number of authors are addressing that particular issue, e.g., Damerdji et al. (1997).

Perhaps more-ambitious, long-term goals include those of enhancing automatic run-control procedures (e.g., Steiger et al., 2005), multivariate point and confidence interval estimation, and applications of variance estimation techniques to quality control, ranking-and-selection, optimization, and financial analysis problems.

Acknowledgement

Partial support for our research was provided by National Science Foundation Grant DMI-0400260.

References

Alexopoulos, C., Goldsman, D., Tokol, G. (2000). Properties of batched quadratic-form variance parameter estimators for simulations. *INFORMS Journal on Computing* 13, 149–156.

Alexopoulos, C., Argon, N.T., Goldsman, D., Steiger, N.M., Tokol, G., Wilson, J.R. (2005a). Overlapping variance estimators for simulation: Implementation and evaluation. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Alexopoulos, C., Argon, N.T., Goldsman, D., Tokol, G., Wilson, J.R. (2005b). Overlapping variance estimators for simulation. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Billingsley, P. (1968). *Convergence of Probability Measures*. Wiley, New York.

Bratley, P., Fox, B.L., Schrage, L.E. (1987). *A Guide to Simulation*, 2nd edition. Springer-Verlag, New York.

Calvin, J.M., Nakayama, M.K. (2002). A comparison of output-analysis methods for simulations of processes with multiple regeneration sequences. In: Yücesan, E., Chen, H., Snowdon, J.L. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 328–335.

Chien, C., Goldsman, D., Melamed, B. (1997). Large-sample results for batch means. *Management Science* 43, 1288–1295.

Damerdji, H. (1995). Mean-square consistency of the variance estimator in steady-state simulation output analysis. *Operations Research* 43, 282–291.

Damerdji, H., Henderson, S.G., Glynn, P.W. (1997). Computational efficiency evaluation in output analysis. In: Andradóttir, S., Healy, K.J., Withers, D.H. (Eds.), *Proceedings of the 1997 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 208–215.

Foley, R.D., Goldsman, D. (1999). Confidence intervals using orthonormally weighted standardized times series. *ACM Transactions on Modeling and Computer Simulation* 9, 297–325.

Glynn, P.W., Iglehart, D.L. (1990). Simulation output analysis using standardized time series. *Mathematics of Operations Research* 15, 1–16.

Glynn, P.W., Whitt, W. (1991). Estimating the asymptotic variance with batch means. *Operations Research Letters* 10, 431–435.

Goldsman, D., Meketon, M.S. (1986). A comparison of several variance estimators. Technical Report J-85-12, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Goldsman, D., Schruben, L.W. (1990). New confidence interval estimators using standardized time series. *Management Science* 36, 393–397.

Goldsman, D., Meketon, M.S., Schruben, L.W. (1990). Properties of standardized time series weighted area variance estimators. *Management Science* 36, 602–612.

Goldsman, D., Kang, K., Seila, A.F. (1999). Cramér–von Mises variance estimators for simulations. *Operations Research* 47, 299–309.

Goldsman, D., Kang, K., Kim, S.-H., Seila, A.F., Tokol, G. (2003). Combining standardized time series area and Cramér–von Mises variance estimators. Technical report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Meketon, M.S., Schmeiser, B.W. (1984). Overlapping batch means: Something for nothing? In: Sheppard, S., Pooch, U.W., Pegden, C.D. (Eds.), *Proceedings of the 1984 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 227–230.

Sargent, R.G., Kang, K., Goldsman, D. (1992). An investigation of finite-sample behavior of confidence interval estimators. *Operations Research* 40, 898–913.

Schmeiser, B.W. (1982). Batch size effects in the analysis of simulation output. *Operations Research* 30, 556–568.

Schruben, L.W. (1983). Confidence interval estimation using standardized time series. *Operations Research* 31, 1090–1108.

Song, W.-M.T., Schmeiser, B.W. (1995). Optimal mean-squared-error batch sizes. *Management Science* 41, 110–123.

Steiger, N.M., Wilson, J.R. (2001). Convergence properties of the batch-means method for simulation output analysis. *INFORMS Journal on Computing* 13, 277–293.

Steiger, N.M., Lada, E.K., Wilson, J.R., Joines, J.A., Alexopoulos, C., Goldsman, D. (2005). ASAP3: A batch means procedure for steady-state simulation analysis. *ACM Transactions on Modeling and Computer Simulation* 15, 39–73.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13016-9

Chapter 16

Simulation Algorithms for Regenerative Processes

Peter W. Glynn

Department of Management Science and Engineering, Stanford University, USA
E-mail: glynn@stanford.edu

Abstract

This chapter is concerned with reviewing the basic ideas and concepts underlying the use of regenerative structure in the development of efficient simulation algorithms. While it has long been known that discrete state space Markov chains exhibit regenerative structure, we argue that well-behaved discrete-event simulations typically also contain an embedded sequence of regeneration times. However, algorithmic identification of the corresponding regeneration times turns out to be a nontrivial problem. We discuss the theoretical and implementation issues involved in identifying the corresponding regeneration times, and describe how regenerative methodology supplies effective solutions to several difficult simulation problems. In particular, we discuss the use of regeneration in the context of steady-state simulation as a means of efficiently computing confidence intervals and correcting for initial bias. We also point out that regeneration has the potential to offer significant algorithmic efficiency improvements to the simulationist, and illustrate this idea via discussion of steady-state gradient estimation and computation of infinite horizon expected discounted reward.

1 Introduction

Let $V = (V(t)\colon\ t \geqslant 0)$ be a real-valued stochastic process in which $V(t)$ represents the simulation output collected at time t. Roughly speaking, the process V is said to be (classically) regenerative if there exist random times $T(0) < T(1) < \cdots$ at which the process "regenerates" (in the sense that V probabilistically starts afresh at each time $T(i)$, $i \geqslant 0$, and evolves independently of the process prior to time $T(i)$). Such regenerative structure implies that V can be viewed conceptually as a sequence of independent "cycles" $(V(s)\colon\ T(i-1) \leqslant s < T(i))$ that are "pasted together" (where we adopt the convention that $T(-1) = 0$). Thus, the infinite time behavior of V over $[0, \infty)$ is implicitly captured in the behavior of V over a cycle. Hence, in principle, virtually any expectation of V over $[0, \infty)$ can be alternatively described as an expectation involving cycle-related quantities. This observation is the key insight that underlies regenerative simulation.

The use of regenerative structure as an algorithmic tool in the simulation setting has primarily focused on its use in the context of steady-state simulation. The first suggestion that regenerative cycles could play a useful role in steady-state simulation output analysis came from Cox and Smith (1961), and the idea was further developed in Kabak (1968). However, the first comprehensive development of the regenerative method for steady-state simulation output analysis came in a series of papers of Crane and Iglehart (1974a, 1974b, 1975), as well as concurrent work by Fishman (1973, 1974). The great majority of subsequent work on algorithmic exploitation of regeneration has followed the historic tradition of focusing on its application to steady-state simulation output analysis.

In this chapter we focus our discussion on the key theoretical and algorithmic issues underlying the use of regeneration in the steady-state simulation context. We start, in Section 2, by describing the key challenges that confront a simulationist in analyzing steady-state simulation output, while Section 3 discusses the basic regenerative approach to forming an estimator for the so-called "time-average variance constant". Section 4 offers some discussion of how the particular choice of regenerative structure influences the efficiency of the method, and Section 5 describes the regenerative solution to the initial transient problem and the construction of low-bias steady-state estimators. In Sections 6–8 we discuss the theoretical issue of when a simulation is regenerative, with a particular focus on when a discrete-event simulation contains algorithmically identifiable regenerative structure. Section 9 then discusses steady-state regenerative analysis from the perspective of martingale theory.

The last two sections of the chapter are intended to give the reader a hint of the role that regeneration can play in the development of computationally efficient algorithms for other simulation problems. In particular, we show that in computing either steady-state gradients or infinite-horizon expected discounted reward that regeneration offers the simulationist the opportunity to not only construct asymptotically valid confidence statements but to also improve computational efficiency. While regeneration is primarily understood within the simulation community as offering a vehicle for analysis of simulation output, our two examples are intended to argue that regeneration has the potential to also play a significant role in the variance reduction context.

2 The steady-state simulation problem

Let $V = (V(t)\colon\ t \geqslant 0)$ be a real-valued stochastic process in which $V(t)$ represents the value of the simulation output process at (simulated) time t. For example, $V(t)$ could represent the total work-in-process at time t in a production context or the inventory position at time t in a supply chain setting. Throughout this chapter, we use a continuous time formulation to describe the relevant theory. (Note that any discrete-time sequence $(V_n\colon\ n \geqslant 0)$ can be embedded into continuous time via the definition $V(t) = V_{\lfloor t\rfloor}$ for $t \geqslant 0$, where $\lfloor x\rfloor$ is the greatest integer less than or equal to x.)

Many simulation applications demand that the simulationist compute a steady-state performance measure. To be specific, suppose that V satisfies a law of large numbers (LLN), so that there exists a (deterministic) constant α for which

$$\frac{1}{t}\int_0^t V(s)\,\mathrm{d}s \Rightarrow \alpha \tag{1}$$

as $t \to \infty$, where "$\Rightarrow$" denotes weak convergence. The constant α appearing in (1) is known as the steady-state mean of V. Computing α is the central problem in steady-state simulation.

Given the LLN (1), the time-average $\overline{V}(t) \triangleq t^{-1}\int_0^t V(s)\,\mathrm{d}s$ is the natural estimator for α. However, in view of the fact that the simulation of V will usually be initialized in a state that is atypical of equilibrium behavior, the process V will at best exhibit the stationarity associated with steady-state dynamics only in an approximate sense. As a consequence, such a simulation of V over $[0, t]$ will necessarily include some "initial transient period" over which the simulation outputs will be biased as estimators of steady-state performance. This, in turn, induces bias in the estimator $\overline{V}(t)$ (known as "initial bias"). While the effect of the initial transient can be expected to dissipate as $t \to \infty$, it can have a significant "small sample" impact on the quality of the estimator $\overline{V}(t)$.

To reduce the effect of the initial transient on the steady-state estimation algorithm, it is commonly recommended that the simulationist expend his computer budget on one (long) replication of V (for which the time horizon t can be made large), rather than multiple short replications. Because of the fact that only one realization of the process V is then simulated, estimating the variance of the associated estimator can then be challenging.

In particular, it is usually the case that a process V satisfying the LLN (1) will also satisfy a central limit theorem (CLT). Specifically, there exists a (deterministic) constant $\sigma \in (0, \infty)$ for which

$$t^{1/2}\bigl(\overline{V}(t) - \alpha\bigr) \Rightarrow \sigma\mathrm{N}(0, 1) \tag{2}$$

as $t \to \infty$, where $\mathrm{N}(0, 1)$ denotes a normal random variable (r.v.) having mean zero and unit variance. The constant σ^2 is called the time-average variance constant (TAVC) of V. In view of (2), it is easily verified that

$$\left[\overline{V}(t) - \frac{z\sigma}{\sqrt{t}}, \overline{V}(t) + \frac{z\sigma}{\sqrt{t}}\right] \tag{3}$$

is an (asymptotic) $100(1-\delta)\%$ confidence interval for α, provided that z is chosen so that $\mathrm{P}(-z \leqslant \mathrm{N}(0, 1) \leqslant z) = 1 - \delta$. Of course, (3) can be computed at a practical level only if the TAVC σ^2 is known.

Since knowledge of σ^2 is virtually never available, the simulationist must instead estimate σ^2 from the observed simulation up to time t. If V is stationary,

σ^2 can (in great generality) be represented in terms of the spectral density $\tilde{f}(\cdot)$. Specifically, $\sigma^2 = 2\pi\tilde{f}(0)$, where

$$\tilde{f}(\lambda) = \frac{1}{\pi}\int_0^\infty \cos(\lambda u)\,\mathrm{cov}\big(V(0), V(u)\big)\,\mathrm{d}u. \tag{4}$$

Spectral density estimation for stationary processes has been well studied in the literature; see, for example, Chapter 9 of Anderson (1971). Such estimators converge at a rate of $t^{-1/3}$ or $t^{-2/5}$, with the specific rate depending on the degree of differentiability of $\tilde{f}$. While virtually all steady-state simulations involve simulating nonstationary stochastic processes that contain an initial transient period, one would expect that the best possible convergence rate for an estimator of the TAVC σ^2 will be no faster than that which is achievable in the stationary setting. Hence, more problem structure must be assumed in order to obtain a TAVC that converges (for example) at rate $t^{-1/2}$.

In the next section, we show how regenerative structure can be exploited to obtain an estimator for the TAVC σ^2 that converges at rate $t^{-1/2}$ in the simulation time horizon t. Given the substantial body of theory establishing that $t^{-1/2}$ is typically an optimal rate of convergence for statistical estimators (see, for example, Chapter 2 of Ibragimov and Has'minskii, 1981), this suggests that regenerative structure permits the simulationist to obtain TAVC estimators that converge at the best possible rate.

3 The regenerative estimator for the TAVC

To obtain a TAVC estimator that converges to σ^2 at rate $t^{-1/2}$, one needs to assume additional structure about the process V. To illustrate this idea, suppose that the simulation output process V is a (continuous-time) autoregressive process satisfying

$$\mathrm{d}V(s) = -\gamma V(s)\,\mathrm{d}s + \mathrm{d}W(s), \tag{5}$$

where $\gamma > 0$ and $W = (W(s)\colon s \geqslant 0)$ is a square integrable process with stationary independent increments for which $\mathrm{E}W(s) = \mu s$ and $\mathrm{var}\,W(s) = \eta^2 s$ for $s \geqslant 0$. It is easily verified that

$$V(t) = \mathrm{e}^{-\gamma t}V(0) + \int_0^t \mathrm{e}^{-\gamma(t-s)}\,\mathrm{d}W(s)$$

and that V satisfies (2) with $\alpha = \mu/\gamma$ and $\sigma^2 = \eta^2/\gamma$. Hence, if the simulation output process can be assumed to evolve according to (5), we can estimate σ^2 via $\hat{\eta}^2/\hat{\gamma}$, where $\hat{\eta}^2$ and $\hat{\gamma}$ are appropriately chosen estimators for the parameters η^2 and γ underlying (5). If V satisfies (5), it can be shown (in great generality) that the resulting TAVC estimator converges at a rate $t^{-1/2}$. The problem, of course, is that it is rarely (if ever) the case that the simulation output process V evolves precisely according to (5). As a consequence, a TAVC estimation algorithm based on assuming that V's dynamics are governed by (5) can, at best, provide only an approximation to the true σ^2. Nevertheless, this autoregressive approach to estimating σ^2 can, when appropriately applied, offer an effective means of estimating σ^2.

The key idea exploited above is the fact that the TAVC σ^2 can be easily and exactly computed for the class of processes described by (5). One then uses a "plug-in" estimator to estimate the unknown quantities appearing in the corresponding expression for the TAVC.

The importance of the regenerative method lies in the fact that a large class of interesting and useful steady-state simulations fall into the class of regenerative processes, and that a simple expression for the TAVC of such processes can be derived. For example, suppose that $V(t) = g(X(t))$, where $X = (X(t)\colon\ t \geqslant 0)$ is an irreducible positive recurrent continuous-time Markov chain (CTMC) living on discrete state space S, and where $g : S \to \mathbb{R}$ is a given performance measure. Fix a state $z \in S$. Then, V is regenerative with cycles defined by the consecutive times $(T(n)\colon\ n \geqslant 0)$ at which X enters z. The class of CTMCs (and its discrete-time cousin, the class of discrete-time Markov chains (DTMCs)) form an important class of models that are commonly simulated and that enjoy regenerative structure.

Simple expressions for the steady-state mean α and TAVC σ^2 can be derived in the regenerative setting. For α, note that (1) suggests that

$$\frac{1}{T(n)} \int_0^{T(n)} V(s)\,\mathrm{d}s \Rightarrow \alpha \tag{6}$$

as $n \to \infty$. But the left-hand side of (6) equals

$$\frac{n^{-1}\sum_{i=1}^{n} Y_i}{n^{-1}\sum_{i=1}^{n} \tau_i},$$

where

$$Y_i \triangleq \int_{T(i-1)}^{T(i)} V(s)\,\mathrm{d}s \quad \text{and}$$

$$\tau_i \triangleq T(i) - T(i-1).$$

Since $(Y_i\colon\ i \geqslant 1)$ is a sequence of independent and identically distributed (i.i.d.) r.v.'s, $\overline{Y}_n \triangleq n^{-1}\sum_{i=1}^{n} Y_i \Rightarrow \mathrm{E}Y_1$ as $n \to \infty$ (provided $\mathrm{E}|Y_1| < \infty$). Similarly, we expect that $\bar{\tau}_n \triangleq n^{-1}\sum_{i=1}^{n} \tau_i \Rightarrow \mathrm{E}\tau_1$ as $n \to \infty$, so that the identity

$$\alpha = \frac{\mathrm{E}Y_1}{\mathrm{E}\tau_1} \tag{7}$$

must hold.

To heuristically derive a corresponding expression for σ^2, note that a regenerative process is (in great generality) asymptotically stationary. Given (4), we expect to be able to represent σ^2 as

$$\sigma^2 = 2\int_0^\infty \operatorname{cov}\bigl(V^*(0), V^*(r)\bigr)\,\mathrm{d}r, \tag{8}$$

where $V^* = (V^*(r){:}\ r \geqslant 0)$ is a stationary version of V. Put $V_c^*(r) = V^*(r) - \alpha$ and rewrite (8) as

$$\sigma^2 = 2\int_0^\infty \mathrm{E}V_c^*(0)V_c^*(r)\,\mathrm{d}r. \tag{9}$$

The stationary version V^* is itself regenerative, with regeneration times $0 < T^*(0) < T^*(1) < \cdots$. In view of the independence across cycles and the fact that $\mathrm{E}V_c^*(r) = 0$, we might hope that the right-hand side of (9) simplifies to

$$2\,\mathrm{E}\int_0^{T^*(0)} V_c^*(0)V_c^*(r)\,\mathrm{d}r. \tag{10}$$

Given the approximate stationarity of V, (10) should be approximately equal to

$$2\,\mathrm{E}\int_s^{T(N(s)+1)} V_c(s)V_c(r)\,\mathrm{d}r$$

when s is large, where $N(s) = \max\{n \geqslant -1{:}\ T(n) \leqslant s\}$. Averaging over $s \in [0, T(n)]$, this suggests that

$$\mathrm{E}\frac{2}{T(n)}\int_0^{T(n)}\int_s^{T(N(s)+1)} V_c(s)V_c(r)\,\mathrm{d}r\,\mathrm{d}s$$

should be close to σ^2 when t is large. But

$$\begin{aligned}
&2\int_0^{T(n)}\int_s^{T(N(s)+1)} V_c(s)V_c(r)\,\mathrm{d}r\,\mathrm{d}s \\
&\quad = 2\sum_{i=0}^{n}\int_{T(i-1)}^{T(i)} V_c(s)\int_s^{T(i)} V_c(r)\,\mathrm{d}r\,\mathrm{d}s \\
&\quad = \sum_{i=0}^{n} Z_i^2,
\end{aligned}$$

where $Z_i \triangleq Y_i - \alpha\tau_i$. The i.i.d. cycle structure implies that $n^{-1}\sum_{i=1}^{n} Z_i^2 \Rightarrow \mathrm{E}Z_1^2$ as $n \to \infty$. Equation (10) therefore suggests that the equality

$$\sigma^2 = \frac{\mathrm{E}Z_1^2}{\mathrm{E}\tau_1} \tag{11}$$

should hold.

Given the equality (11), the obvious “plug-in” estimator for the TAVC σ^2 based on simulating V over $[0, t]$ is

$$\frac{\sum_{i=1}^{N(t)}(Y_1 - \overline{V}(t)\tau_i)^2/N(t)}{\sum_{i=1}^{N(t)}\tau_i/N(t)}$$

or its asymptotically equivalent variant

$$\hat{\sigma}^2(t) = \frac{1}{t}\sum_{i=1}^{N(t)}\bigl(Y_1 - \overline{V}(t)\tau_i\bigr)^2.$$

The following theorem makes rigorous the validity of the regenerative TAVC estimator $\hat{\sigma}^2(t)$. For the proof, see Glynn and Whitt (1993, 2002) and Glynn and Iglehart (1993).

Theorem 1. *Suppose that V is regenerative with respect to the regeneration time sequence $(T(n)\colon n \geqslant 0)$. Assume that $\mathrm{E}\tau_1 < \infty$. Then, there exist (deterministic) constants α and σ^2 such that*

$$t^{1/2}\bigl(\overline{V}(t) - \alpha\bigr) \Rightarrow \sigma\mathrm{N}(0,1) \tag{12}$$

as $t \to \infty$ if and only if $\mathrm{E}|Y_1| < \infty$ and $\mathrm{E}Z_1^2 < \infty$, in which case

$$\alpha = \frac{\mathrm{E}Y_1}{\mathrm{E}\tau_1}, \qquad \sigma^2 = \frac{\mathrm{E}Z_1^2}{\mathrm{E}\tau_1}.$$

Furthermore, if (12) holds, then

$$\hat{\sigma}^2(t) \Rightarrow \sigma^2$$

as $t \to \infty$.

Theorem 1 shows that $\hat{\sigma}^2(t)$ is consistent as an estimator for σ^2 precisely when the CLT (12) holds. When (12) holds with $\sigma^2 > 0$, then

$$\left[\overline{V}(t) - z\sqrt{\frac{\hat{\sigma}^2(t)}{t}}, \overline{V}(t) + z\sqrt{\frac{\hat{\sigma}^2(t)}{t}}\right]$$

is an approximate $100(1-\delta)\%$ confidence interval for α.

4 Choice of the optimal regeneration state

Given a simulation of V over the time interval $[0, t]$, the natural point estimator for the steady-state mean α is, of course, the time-average $\overline{V}(t)$. While it may be desirable to modify $\overline{V}(t)$ to deal with initial transient or initial bias effects, one would expect such modifications to be of small order asymptotically.

Hence, any reasonable point estimator for α will either be exactly equal to $\overline{V}(t)$ or asymptotically equivalent to $\overline{V}(t)$. Of course, the r.v. $\overline{V}(t)$ is not influenced in any way by the choice of the regeneration or return state z.

On the other hand, the TAVC estimator $\hat{\sigma}^2(t)$ is defined relative to a specific choice of the regeneration or return state $z \in S$. A natural question that then arises is the determination of the state z^* that is the "best" choice of return state for estimating σ^2. This question can be resolved by studying the rate of convergence of $\sqrt{\hat{\sigma}^2(t)}$ to $\sqrt{\sigma^2}$.

Theorem 2. *Suppose that* $\mathrm{E}[Y_1^4 + \tau_1^4] < \infty$. *Then*

$$t^{1/2}\bigl(\overline{V}(t) - \alpha, \hat{\sigma}(t) - \sigma\bigr) \Rightarrow \mathrm{N}(0, D)$$

as $t \to \infty$, *where* $\mathrm{N}(0, D)$ *is a bivariate normal r.v. with mean* 0 *and covariance matrix* D *given by*

$$D = \frac{1}{\mathrm{E}\tau_1}\begin{pmatrix} \mathrm{E}Z_1^2 & \frac{\mathrm{E}A_1Z_1 - \lambda \mathrm{E}Z_1^2}{2\sigma} \\ \frac{\mathrm{E}A_1Z_1 - \lambda \mathrm{E}Z_1^2}{2\sigma} & \frac{\mathrm{E}A_1^2 - 2\lambda \mathrm{E}A_1Z_1 + \lambda^2 \mathrm{E}Z_1^2}{4\sigma^2} \end{pmatrix},$$

where $A_i = Z_i^2 - \sigma^2\tau_i$ *and* $\lambda = 2\,\mathrm{E}Z_1\tau_1/\mathrm{E}\tau_1$.

See Glynn and Iglehart (1987) for the proof. Theorem 2 establishes that the TAVC estimator does indeed converge at rate $t^{-1/2}$. It further describes the asymptotic variance of $\hat{\sigma}^2(t)$ in terms of the given regenerative cycle structure. The asymptotic variance can be explicitly computed for certain CTMC models; see Glynn and Iglehart (1986). These examples make clear that there is, unfortunately, no simple guidance available for how to choose the best possible regeneration state. In particular, the examples make clear that choosing the regeneration state $\tilde{z}$ that minimizes the mean return time is not necessarily the choice that minimizes the asymptotic variance of $\hat{\sigma}^2(t)$.

One odd characteristic of Theorem 2 is that the covariance entry D_{12} ($= D_{21}$) of the matrix D appearing there turns out to be independent of the choice of regeneration state. This result, due to Calvin (1994), has no obvious and apparent simple explanation, and is a consequence of a direct computation. By contrast, the entry D_{11} must clearly be independent of the choice of the regeneration state z, since it is the asymptotic variance of the r.v. $\overline{V}(t)$ that is defined independently of z.

5 The regenerative approach to the initial transient and initial bias problems

As discussed in Section 2, one of the major challenges in steady-state simulation is the mitigation of effects due to the initial transient and initial bias. We deal first with the better understood issue of how to reduce biasing effects due to a nonstationary initialization.

It is usual, in the presence of (1), that there exists $\nu > 0$ such that

$$\mathrm{E}V(t) = \alpha + \mathrm{O}\bigl(\mathrm{e}^{-\nu t}\bigr) \tag{13}$$

as $t \to \infty$, where $\mathrm{O}(h(t))$ represents a function that is bounded by a constant multiple of $|h(t)|$ as $t \to \infty$. For example, (13) is known to typically hold for geometrically ergodic Markov processes; see Meyn and Tweedie (1993). Given (13),

$$\int_0^t \mathrm{E}\bigl(V(s) - \alpha\bigr)\,\mathrm{d}s = b + \mathrm{O}\bigl(\mathrm{e}^{-\nu t}\bigr)$$

as $t \to \infty$, so that

$$\mathrm{E}\overline{V}(t) = \alpha + t^{-1}b + \mathrm{O}\bigl(\mathrm{e}^{-\nu t}\bigr) \tag{14}$$

as $t \to \infty$, where

$$b = \int_0^\infty \mathrm{E}\bigl(V(s) - \alpha\bigr)\,\mathrm{d}s.$$

An estimator with lower initialization bias can therefore be constructed if one can find an asymptotically unbiased estimator for b. Deriving such an estimator without imposing additional structure is an impossibility in the single replication context, because only one realization of the process from which to estimate b is available. On the other hand, if the process V is assumed to be regenerative, estimating b should (in principle) be possible, because the i.i.d. cycle structure suggests that the effect of initialization is now implicitly replicated (for example, by permuting the simulated cycles).

Appealing to renewal theory yields the following expression for b; for the proof see Glynn (1994).

Proposition 1. *Suppose that V is regenerative with respect to the regeneration times $0 = T(0) < T(1) < \cdots$. If τ_1 has a density and satisfies*

$$\mathrm{E}\tau_1\left(1 + \int_0^{\tau_1} \bigl|V(s)\bigr|\,\mathrm{d}s\right) < \infty,$$

then (14) holds with

$$b = -\frac{1}{\mathrm{E}\tau_1}\left(\mathrm{E}\int_0^{\tau_1} sV(s)\,\mathrm{d}s - \alpha\,\mathrm{E}\frac{\tau_1^2}{2}\right).$$

In view of Proposition 1, it is now clear how one can potentially reduce the effects of initial bias. In particular, consider the "plug-in" estimator for b defined by

$$\hat{b}(t) = -\frac{1}{T(N(t))}\sum_{i=1}^{N(t)}\left(\int_0^{\tau_i} sV\bigl(T(i-1) + s\bigr)\,\mathrm{d}s - \overline{V}(t)\frac{\tau_i^2}{2}\right).$$

We expect that

$$\mathrm{E}\big(\overline{V}(t) - t^{-1}\hat{b}(t)\big) = \alpha + \mathrm{o}\left(\frac{1}{t}\right) \tag{15}$$

as $t \to \infty$, where $\mathrm{o}(h(t))$ denotes a function that, when divided by $|h(t)|$, tends to zero as $t \to \infty$. Given that $\alpha_1(t) \triangleq \overline{V}(t) - t^{-1}\hat{b}(t)$ generally satisfies

$$\frac{\operatorname{var} \alpha_1(t)}{\operatorname{var} \overline{V}(t)} \to 1 \tag{16}$$

as $t \to \infty$, (15) establishes that $\alpha_1(t)$ has lower asymptotic bias while suffering no increase in asymptotic variance. For additional discussion, see Glynn (1994).

Consider the estimators:

$$\begin{aligned}
\alpha_2(t) &= \overline{V}\big(T\big(N(t)+1\big)\big), \\
\alpha_3(t) &= \overline{V}\big(T\big(N(t)\big)\big) + \frac{1}{t^2}\sum_{i=1}^{N(t)}\big(Y_i - \overline{V}(t)\tau_i\big)\tau_i, \\
\alpha_4(t) &= N(t)\overline{V}\big(T\big(N(t)\big)\big) - \frac{N(t)-1}{N(t)}\sum_{i=1}^{N(t)}\frac{\sum_{j\neq i} Y_j}{\sum_{j\neq i}\tau_j}.
\end{aligned}$$

Each of the above estimators duplicates the performance of $\alpha_1(t)$, in the sense that each satisfies both (15) and (16); see Meketon and Heidelberger (1982) and Glynn and Heidelberger (1990, 1992).

Turning next to the question of identifying the duration of the initial transient period, recall that the sequence of cycle variables $((Y_i, \tau_i)\colon\ i \geqslant 1)$ is i.i.d. Hence, on the time-scale of regenerative cycles, the initial transient disappears entirely. Furthermore, recall that the steady-state mean α can be expressed as the ratio of expectations (7) defined in terms of cycle-related quantities Y_i and τ_i. Hence, if one simulates V over n regenerative cycles (to time $T(n)$), the natural estimator for α is the ratio estimator $\overline{V}(T(n)) = \overline{Y}_n/\bar{\tau}_n$. The bias of this estimator arises as a consequence of the fact that $\overline{Y}_n/\bar{\tau}_n = h(\overline{Y}_n, \bar{\tau}_n)$, where h is the nonlinear function defined by $h(x_1, x_2) = x_1/x_2$. Thus, on the time-scale of regenerative cycles, initialization bias manifests itself as estimator bias due to nonlinearity. Such nonlinearity bias has long been studied within the statistical literature, and a large number of remedies for dealing with bias of this kind have been developed over the years: Taylor expansion methods (Cramér, 1946), the jack-knife (Miller, 1974), the bootstrap (Efron and Tibshirani, 1993), and sectioning (Lewis and Orav, 1989).

Thus, regenerative structure suggests a variety of different mechanisms for dealing with initial bias (and, on the regenerative cycle time scale, the initial transient).

6 When is a simulation regenerative?

As has been seen in preceding sections, regenerative structure turns out to be algorithmically useful in developing solutions to various aspects of the steady-state simulation problem. Furthermore, regenerative structure can be easily identified in the setting of discrete state space Markov chains, in either discrete or continuous time.

Of course, most real-world discrete-event simulations do not involve simulating a discrete state space Markov processes. Much more complicated models are typically simulated. However, one can persuasively argue that the great majority of discrete event simulations can, from a mathematical standpoint, be viewed as simulations of Markov processes (living on a continuous state space rather than a discrete state space). In particular, by adding supplementary variables to the "physical state" (e.g., the location of each customer in a network) of the system, one typically ends up with a state descriptor that evolves according to a Markov process. For example, one can supplement the physical state by adding the remaining time to completion of each currently scheduled event associated with the currently occupied physical state. Thus, one may abstractly view the typical discrete-event simulation as corresponding to the simulation of a Markov process $X = (X(t)\colon\ t \geqslant 0)$ living on a continuous state space S, where the continuous component of the state records the remaining time to event completion for each of the active event "clocks".

We assume throughout the reminder of the paper that the state space S is a separable metric space (so that, for example, open and closed subsets of $\mathbb{R}^d$ are covered by our theory).

Now that we have established that the typical discrete-event simulation can be viewed as a Markov process, we next argue that any Markov process for which the steady-state simulation problem is well defined necessarily exhibits regenerative structure. This, in turn, will show that any discrete-event simulation for which the steady-state simulation problem makes sense must contain regenerative structure.

We start with a precise definition of "well-posedness" for the steady-state simulation problem. For $x \in S$, let $\mathrm{P}_x(\cdot)$ and $\mathrm{E}_x(\cdot)$ be the probability and expectation operator, respectively, under which $X(0) = x$.

Definition 1. We say that the steady-state simulation problem is *well-posed* for the S-valued Markov process $X = (X(t)\colon\ t \geqslant 0)$ if for each bounded function $g : S \to \mathbb{R}$, there exists $\alpha(g)$ such that for $x \in S$,

$$\frac{1}{t}\int_0^t \mathrm{E}_x g\bigl(X(s)\bigr)\,\mathrm{d}s \to \alpha(g)$$

as $t \to \infty$.

According to the definition, the expectation of $V(s) = g(X(s))$ must converge (at least in an average sense) to a number $\alpha = \alpha(g)$ that is independent of the initial state x. This seems a reasonable definition of well-posedness, for otherwise, there exists a bounded function g for which $t^{-1}\int_0^t \mathrm{E}_x g(X(s))\,\mathrm{d}s$ either does not converge or converges to a limit that depends on the initial state x. In either case, this precludes what is normally understood by the notion of steady-state.

Recall that a sequence $(W_i\colon i \geqslant 0)$ is said to be one-dependent if $(W_j\colon j < i)$ is independent of $(W_j\colon j > i)$ for each $i \geqslant 1$.

Theorem 3. *Let $X = (X(t)\colon t \geqslant 0)$ be an S-valued Markov process for which the steady-state simulation problem is well-posed. Then, there exist random times $0 \leqslant T(0) < T(1) < \cdots$ such that:*

(i) *The probability distribution $\mathrm{P}_x((X(T(i-1)+s)\colon 0 \leqslant s \leqslant \tau_i) \in \cdot)$ is independent of both $x \in S$ and $i \geqslant 1$;*
(ii) *The sequence of cycles $((X(T(i-1)+s)\colon 0 \leqslant s \leqslant \tau_i)\colon i \geqslant 0)$ is one-dependent;*
(iii) $\mathrm{E}_x\tau_1 < \infty$.

For the proof, see Glynn (1994).

This theorem asserts that any simulation for which the steady-state simulation problem is well-posed necessarily possesses regenerative structure. However, the regenerative structure identified by this result only guarantees the existence of one-dependent identically distributed cycles. Fortunately, much of the theory developed in Sections 3–5 generalizes from the classical regenerative structure (of i.i.d. cycles) to the setting of one-dependent regenerative processes. For example, the one-dependent analog to $\hat{\sigma}^2(t)$ continues to converge to the TAVC σ^2 at rate $t^{-1/2}$ in the one-dependent setting; see Henderson and Glynn (2001).

However, an alternative approach exists that permits one to use conventional regenerative methodology based on i.i.d. cycle structure. For one-dependent regenerative processes, the ratio formula for the steady-state mean $\alpha(g)$ continues to hold:

$$\alpha(g) = \frac{\mathrm{E}\int_0^{\tau_1} g(X(T(0)+s))\,\mathrm{d}s}{\mathrm{E}\tau_1}. \tag{17}$$

To estimate $\alpha(g)$, we simulate X over the cycle corresponding to the interval $[T(0), T(1)]$. At time $T(1)$, rather than continuing the simulation of X, one now terminates the simulation. One then independently draws a new initial condition from $\mathrm{P}(X(T(0)) \in \cdot)$ and simulates a second independent trajectory of X up to completion of its corresponding first cycle. By repeating this process, we are simulating independent realizations of X over its first cycle. By "pasting" these i.i.d. cycles back to back, one is generating a new process $\widetilde{X}$ that is regenerative in the classical sense (with i.i.d. cycles). Given the ratio formula (17), the steady-state of $\widetilde{X}$ exactly coincides with that of the one-dependent process X. Hence, if one simulates $\widetilde{X}$ rather than X, all the methods of Sections 3–5 apply without change.

7 When is a GSMP regenerative?

Section 6 makes clear that regeneration is the rule rather than the exception for well-behaved steady-state simulations. This, however, leaves open the question of when a specific simulation model has the structure necessary to guarantee that the associated steady-state simulation is well behaved.

We shall focus exclusively, in this section, on conditions under which discrete-event simulations possess the required structure. We take the point of view here that a discrete-event simulation is tantamount to simulation of a class of processes known as generalized semi-Markov processes (GSMPs). To describe a GSMP, we make concrete our discussion of Section 6, in which we argued that a discrete-event simulation can be viewed as a Markov process. Let $\mathcal{P}$ be a finite or countably infinite set of physical states and let $\mathcal{E}$ be a finite set corresponding to those events that can trigger physical state transitions. For each of the events $e \in \mathcal{E}$ that are active in a physical state $s \in \mathcal{P}$, we can conceptually imagine that there is an associated clock. When a clock e^* runs down to zero in state s, it triggers a physical state transition to s' with probability $p(s'; s, e^*)$. The clocks e' active in s' correspond to events that were already scheduled in the previous state s (but had not yet run down to zero), in which case the clocks continue running down to zero in s' at rate $r(s', e')$, or correspond to new events that must be scheduled in s'. The clocks associated with such new events in e' are independently scheduled according to distributions $F(\cdot; e', s', e^*, s)$, where e^* is the trigger event that initiated the transition. Experienced simulationists will recognize that the state of the clocks effectively describes the "future event schedule" of the associated discrete-event simulation.

Given that the physical state and future event schedule is precisely the information necessary to evolve a discrete-event simulation forward in time, it is clear that $X(t) = (S(t), C(t))$ is Markov, where $S(t)$ is the physical state occupied at time t (known as the GSMP corresponding to X), and $C(t)$ is the vector of clock readings corresponding to active events.

To develop a sufficient condition under which the steady-state simulation problem for the Markov process $X = (X(t)\colon\ t \geqslant 0)$ is well posed, one clearly needs to invoke an assumption that ensures that there is a unique stationary distribution for X. This, of course, requires an irreducibility hypothesis of some kind.

Definition 2. The GSMP corresponding to X is said to be *irreducible* if for each pair $(s, s') \in \mathcal{P} \times \mathcal{P}$, there exists a finite sequence of states $s_1, \ldots, s_n$ and events $e_1, \ldots, e_n$ such that for $0 \leqslant i \leqslant n$, e_i is active in s_i ($s_0 \triangleq s$, $s_{n+1} = s'$) and

$$\prod_{i=0}^{n} p(s_{i+1}; s_i, e_i) r(s_i, e_i) > 0.$$

The above assumption induces irreducibility over the physical states of the GSMP. For irreducibility over the clock readings of X, consider the following definition.

Definition 3. The GSMP corresponding to X satisfies the *positive density assumption* if each distribution $F(\cdot; e', s', e, s)$ has a density component that is strictly bounded away from zero on an interval $[0, \varepsilon]$ with $\varepsilon > 0$.

We are now ready to state our main result for GSMPs.

Theorem 4. *Suppose that*:

(i) *The GSMP corresponding to X is irreducible and satisfies the positive density assumption*;
(ii) $|\mathcal{P}| < \infty$;
(iii) $\int_{[0,\infty)} tF(\mathrm{d}t; e', s', e, s) < \infty$ *for all* (e', s', e, s).

Then, the steady-state simulation problem for the Markov process X is well-posed.

For a proof, see Glynn and Haas (2006). The above conditions are necessary, in the sense that if any of the three conditions above is violated, then there exist counterexamples.

8 Algorithmic identification of regenerative structure

Our discussion of Sections 6 and 7 makes clear that regenerative structure exists within the typical discrete-event steady-sate simulation. On the other hand, the TAVC estimator of Section 3, as well as the low bias estimators of Section 5, all depend upon the ability of the simulationist to identify the associated regeneration times. Of course, this identification is trivial in the setting of discrete state space Markov chains, where the regeneration times can be chosen to be those times at which the chain enters a fixed state. Unfortunately, identification of the regenerative structure guaranteed by Theorem 3 is not algorithmically trivial in general.

The main difficulty is that the regenerative structure of Theorem 3 involves the use of randomized regeneration. This means that the regeneration times cannot be identified purely on the basis of simulating X alone. Some additional random variables (i.e., the "randomization") must also be generated. This means, for example, that a "randomization post-processor" must be added to the typical discrete-event simulation in order to be able to exploit the regenerative structure that is theoretically guaranteed to exist.

We now proceed to describe the randomized regeneration structure that arises in the setting of discrete time Markov chains $X = (X_n\colon n \geqslant 0)$. The regenerative structure that can be required in continuous time is generally more complicated to describe; see Sigman (1990) for details. Fortunately, the discrete time theory covers discrete-event simulations. In that setting, the key discrete-time process is the sequence $(X(\Gamma_n)\colon\ n \geqslant 0)$, where $0 = \Gamma_0 < \Gamma_1 < \cdots$ are the epochs at which physical state transitions occur and $(X(t)\colon\ t \geqslant 0)$ is the continuous-time Markov process (associated with GSMPs) described earlier.

Let $(X_n\colon\ n \geqslant 0)$ be an S-valued discrete-time Markov chain satisfying the conditions of Theorem 3. Then, there exists a function $\lambda : S \to [0, 1]$, a subset $A \subseteq S$, an integer $m \geqslant 1$ and a probability distribution φ such that:

(i) $\mathrm{P}_x(X_m \in \cdot) \geqslant \lambda(x)\varphi(\cdot)$, $x \in S$;
(ii) $\mathrm{P}_x(X_n \in A$ infinitely often$) = 1$, $x \in S$;
(iii) $\inf\{\lambda(x)\colon\ x \in A\} > 0$.

To see how this gives rise to randomized regenerative structure, note that condition (i) guarantees that we can write

$$\mathrm{P}_x(X_m \in \cdot) = \lambda(x)\varphi(\cdot) + \bigl(1 - \lambda(x)\bigr)Q(x, \cdot), \tag{18}$$

where $Q(x, \cdot)$ is a probability distribution on S for each x. Hence, conditional on $X_n = x$, we can generate X_{n+m} by generating a Bernoulli r.v. having parameter $\lambda(X_n)$. If the Bernoulli r.v. takes on value 1, then we distribute X_{n+m} according to φ; otherwise, we distribute X_{n+m} according to $Q(X_n, \cdot)$. The segment $(X_{n+1}, \ldots, X_{n+m-1})$ is then generated from the conditional distribution, given the starting state X_n and ending state X_{n+m} for the full segment $(X_n, \ldots, X_{n+m})$. Whenever we distribute X_{n+m} according to φ, X_{n+m} is independent of the history of the chain up to and including step n. Conditions (ii) and (iii) guarantee that there exist infinitely many times $T(0) < T(0)+m \leqslant T(1) < T(1)+m \leqslant T(2) < \cdots$ (separated by gaps of at least m steps) at which the chain is distributed according to φ.

The random times $(T(n)\colon\ n \geqslant 0)$ form cycle boundaries that correspond to a regenerative process with one-dependent identically distributed cycles. In the special case that $m = 1$, the cycles are i.i.d. and the process X is regenerative in the classical sense. One difficulty with this means of identifying regenerative times is that the algorithm is invasive. By invasive, we mean that the algorithm impacts the way we generate sample replications of the process X. In particular, were we to straightforwardly adapt the above mechanism for constructing regeneration times, the basic algorithms used to simulate (for example) discrete-event systems would need to be modified.

In view of this, it is clearly desirable to develop an alternative implementation of the algorithm. Under the conditions above, it can be shown that there exists a function $w : S \times S \to [0, 1]$ for which

$$\lambda(x)\varphi(\mathrm{d}y) = w(x, y)\,\mathrm{P}_x(X_m \in \mathrm{d}y),$$
$$\bigl(1 - \lambda(x)\bigr)Q(x, \mathrm{d}y) = \bigl(1 - w(x, y)\bigr)\,\mathrm{P}_x(X_m \in \mathrm{d}y).$$

Suppose that one simulates a realization of the process X (using one's algorithm of choice). To identify the regeneration time, we apply the following "post-processor" to the sampled realization. If $X_n = x$ and $X_{n+m} = y$, generate a Bernoulli r.v. having probability $w(X_n, X_{n+m})$. If the Bernoulli r.v. is 1, then the process has distribution φ at time $n + m$; otherwise, it does not. This algorithm is probabilistically equivalent to the method described earlier, but it is noninvasive and requires only the post-processing step.

This idea can be applied even to discrete state space Markov chains in which consecutive entry times to a fixed state z constitute easily identified regenerations. To illustrate, suppose that $(P(x, y)\colon x, y \in S)$ is the one-step transition matrix of a discrete-time Markov chain to be simulated. Put $\varphi(y) = P(z, y)$ and

$$\lambda(x) = \inf_{y \in S} \frac{P(x, y)}{P(z, y)}$$

so that $P(x, y) \geqslant \lambda(x)P(z, y)$ for $x, y \in S$. Clearly, $\lambda(z) = 1$. However, $\lambda(x)$ typically will be positive for some states $x \neq z$. Hence, our randomized (classical) regenerations occur every time the chain leaves z, but generally occur more frequently. For this special case in which one sequence of regenerations is a superset of another sequence of regenerations, one can prove that the superset provides a statistically more efficient estimator of the TAVC (i.e., the D_{22} term in Theorem 2 is reduced); see Andradóttir et al. (1995) for details.

Returning to the discrete-event context, the key to practical implementation is computing the function w. This, in turn, involves computing $\mathrm{P}_x(X_m \in \cdot)$. This can easily be done for $m = 1$, but is nontrivial for values of $m \geqslant 2$. Henderson and Glynn (2001) argue that for a discrete-event simulation in which no "event cancellation" is possible, the minimal value of m compatible with conditions (i)–(iii) is m^*, where m^* is the minimal number of events that are ever simultaneously scheduled by the simulation. Hence, any discrete-event simulation possessing a so-called "single state" (i.e., a state in which only one event is scheduled) is easily amenable to regenerative methodology.

When $m^* > 1$, algorithmic identification of regeneration times is substantially more complicated. Suppose, however, that there is no event cancellation in the GSMP and that each "speed" $r(s, e)$ equals one. Assume the GSMP is initially in a state $x = (s, c)$ in which m clocks are active and in which event $\tilde{e} = \tilde{e}(x)$ is the one that has the most remaining time until its clock runs down to zero. Then, the m-step transition probability for $(X(\Gamma_i)\colon i \geqslant 0)$ can be easily computed on that part of the sample space on which the (random) trigger event e^*_m corresponding to transition m equals $\tilde{e}$. In other words, $\mathrm{P}_x(X(\Gamma_m) \in \cdot, e^*_m = \tilde{e}(x))$ can be written down explicitly, because the event $\{e^*_m = \tilde{e}(X_0)\}$ is precisely the event that all of the clocks scheduled prior to time m were set at values greater than that associated with $\tilde{e}(X_0)$. Thus, if we find a function λ and a probability φ for which

$$\mathrm{P}_x\bigl(X_m \in \cdot, e^*_m = \tilde{e}(X_0)\bigr) \geqslant \lambda(x)\varphi(\cdot),$$

we can implement the same approach as desired above, by letting w be chosen so that

$$w(x,y)\,\mathrm{P}_x\big(X_m\in \mathrm{d}y, e_m^*=\tilde{e}(X_0)\big)=\lambda(x)\varphi(\mathrm{d}y).$$

The only significant change is that we generate the Bernoulli r.v. having parameter $w(X_n, X_{n+m})$ only when $e_{n+m}^*=\tilde{e}(X_n)$; see Henderson and Glynn (1999) for a related idea.

This method (and analogous ideas for dealing with nonunit speeds and event cancellation) can be used to construct regenerations for a large class of GSMPs. Practically speaking, however, the method is difficult to apply for GSMPs for which m^* is large. The difficulty is that condition (i) becomes more demanding when m^* is large, because one must lower bound a joint density on a "clock space" having dimension at least m^*. Such lower bounds typically degenerate badly in high-dimensional contexts, leading to implementations in which regeneration occurs (very) infrequently.

Thus, the question of developing easily implemented and practically useful methods for algorithmic identification of regeneration times remains largely open at this point.

9 A martingale perspective on regeneration

To illustrate the connection between martingales and regeneration, we focus here on the case in which $V(t)=g(X(t))$, where $X=(X(t)\colon t\geqslant 0)$ is an irreducible finite state continuous time Markov chain with rate matrix $Q=(Q(x,y)\colon x,y\in S)$. Given the performance measure $g\colon S\to\mathbb{R}$ (where we choose to encode $g=(g(x)\colon x\in S)$ as a column vector), the linear system

$$Qh=-(g-\alpha e) \tag{19}$$

has a solution h. Here, e is the column vector in which all entries equal 1, and (19) is called Poisson's equation.

It is a standard fact in the theory of Markov processes that

$$h\big(X(t)\big)-\int_0^t (Qh)\big(X(s)\big)\,\mathrm{d}s$$

is then a martingale; see, for example, Karlin and Taylor (1981). In other words,

$$h\big(X(t)\big)+\int_0^t g\big(X(s)\big)\,\mathrm{d}s-\alpha t$$

enjoys the martingale property. Suppose that T is a stopping time adapted to X (so that $I(T\leqslant t)$ is a deterministic function of the path $(X(s)\colon 0\leqslant s\leqslant t)$) having finite expectation. Then, the optional sampling theorem for martingales (Breiman, 1968) can be applied, yielding

$$\mathrm{E}h\big(X(T)\big)+\mathrm{E}\int_0^T g\big(X(s)\big)\,\mathrm{d}s-\alpha\,\mathrm{E}T=\mathrm{E}h\big(X(0)\big). \tag{20}$$

The identity (20) holds in great generality for Markov processes in discrete and continuous time, provided that one suitably generalizes the rate matrix Q in an appropriate way; see Ethier and Kurtz (1986).

Note that if the stopping time T can be chosen so that $\mathrm{E}h(X(T)) = \mathrm{E}h(X(0))$, then α can be viewed as the ratio of expectations

$$\alpha = \frac{\mathrm{E}\int_0^T g(X(s))\,\mathrm{d}s}{\mathrm{E}T} \triangleq \frac{\mathrm{E}Y}{\mathrm{E}\tau}. \tag{21}$$

Hence, by simulating independent copies of the process X over the time interval $[0, T]$, α can be computed via a ratio estimator that enjoys precisely the same central limit and bias properties as the conventional regenerative estimator described earlier in this chapter. In particular, if $(Y_1, \tau_1), \ldots, (Y_n, \tau_n)$ are n i.i.d. copies of (Y, τ), then $\alpha_n = (Y_1 + \cdots + Y_n)/(\tau_1 + \cdots + \tau_n)$ satisfies the CLT

$$n^{1/2}(\alpha_n - \alpha) \Rightarrow \sigma \mathrm{N}(0, 1)$$

as $n \to \infty$, where $\sigma^2 = \mathrm{var}(Y - \alpha\tau)/(\mathrm{E}\tau)^2$, and the bias expansion

$$\mathrm{E}\alpha_n = \alpha - \frac{1}{n}\frac{\mathrm{E}(Y - \alpha\tau)\tau}{(\mathrm{E}\tau)^2} + \mathrm{o}\left(\frac{1}{n}\right)$$

holds (thereby providing a straightforward construction of a “low bias” estimator having bias $\mathrm{o}(1/n)$).

Of course, the key is to find a distribution for $X(0)$ and a random time T so that $\mathrm{E}h(X(T)) = \mathrm{E}h(X(0))$.

Since the simulationist does not know the solution to Poisson’s equation, one simply chooses $X(0)$ and T so that $X(0)$ and $X(T)$ have the same distribution. Of course, the easiest way to guarantee this is to let T be the first time that X returns to the state occupied at time $t = 0$. In this case, the above estimation procedure just reduces to the conventional regenerative method based on successive returns to a single state. Thus, the martingale perspective offers a complementary viewpoint regarding the regenerative method for computing steady-state expectations.

However, the optional sampling formula (20) offers the potential for developing new steady-state estimation algorithms. As an example, consider the successive times $\beta_1, \beta_2, \ldots$ at which X enters some fixed nonempty subset $A \subset S$. To be precise, let $\beta_0 = 0$ and put

$$\beta_i = \inf\bigl\{t > \beta_{i-1}\colon\ X(t) \in A, X(t-) \notin A\bigr\}.$$

Then, $(X(\beta_i)\colon\ i \geqslant 1)$ is an irreducible A-valued discrete time Markov chain with stationary distribution ν_A. If $\mathrm{E}_{\nu_A}(\cdot)$ is the expectation operator under which $X(0)$ has distribution ν_A, then (20) yields the equality

$$\mathrm{E}_{\nu_A} h\bigl(X(T)\bigr) + \mathrm{E}_{\nu_A} \int_0^{\beta_1} g\bigl(X(s)\bigr)\,\mathrm{d}s - \alpha\,\mathrm{E}_{\nu_A}\beta_1 = \mathrm{E}_{\nu_A} h\bigl(X(0)\bigr).$$

Because $\mathrm{E}_{\nu_A} h(X(T)) = \mathrm{E}_{\nu_A} h(X(0))$, we obtain the identity

$$\alpha = \frac{\mathrm{E}_{\nu_A} \int_0^{\beta_1} g(X(s))\,\mathrm{d}s}{\mathrm{E}_{\nu_A} \beta_1}. \tag{22}$$

Hence, the regenerative ratio formula (7) generalizes beyond the conventional setting in which cycles are defined in terms of i.i.d. cycles. (Note that the β_i's split the sample path for X into identically distributed cycles having a complex dependency structure.)

One might hope to algorithmically exploit (22) in the same way as for (21). If we could generate variates from ν_A, the algorithm would be clear. Just generate $X(0)$ from ν_A, simulate to time β_1, and compute (Y, τ), where

$$Y = \int_0^{\beta_1} g\bigl(X(s)\bigr)\,\mathrm{d}s,$$
$$\tau = \beta_1.$$

By simulating i.i.d. copies of (Y, τ), we can estimate α via $\alpha_n = (Y_1 + \cdots + Y_n)/(\tau_1 + \cdots + \tau_n)$. Precisely the same CLT and low bias estimation procedure as before can thus be utilized. Because a "multi-state" A is hit more frequently than any single state, we can expect this estimation procedure to be more efficient than the conventional regenerative method based on returns to a single fixed state.

The difficulty, of course, is that we typically are unable to generate variates from ν_A. However, by conditioning on $X(0)$ and $X(\beta_1)$ in (22), we can rewrite (22) as

$$\alpha = \frac{\sum_{x,y} \mathrm{E}_x Y(y) \nu_A(x)}{\sum_{x,y} \mathrm{E}_x \tau(y) \nu_A(x)} \triangleq \frac{\sum_{x,y} \mathrm{E}_x u(x,y)}{\sum_{x,y} \ell(x,y)}, \tag{23}$$

where

$$Y(y) = \int_0^{\beta_1} g\bigl(X(s)\bigr)\,\mathrm{d}s\, I\bigl(X(\beta_1) = y\bigr),$$
$$\tau(y) = \beta_1 I\bigl(X(\beta_1) = y\bigr).$$

Each term in the numerator and denominator can be estimated via simulation of X over $[0, t]$, namely

$$\hat{u}(t, x, y) = \frac{1}{J(t)} \sum_{i=1}^{J(t)} \int_{\beta_{i-1}}^{\beta_i} g\bigl(X(s)\bigr)\,\mathrm{d}s\, I\bigl(X(\beta_{i-1}) = x, X(\beta_i) = y\bigr),$$

$$\hat{\ell}(t, x, y) = \frac{1}{J(t)} \sum_{i=1}^{J(t)} (\beta_i - \beta_{i-1}) I\bigl(X(\beta_{i-1}) = x, X(\beta_i) = y\bigr),$$

where $J(t)$ is the number of times the process X enters A over $[0, t]$. The representation (23) leads to a point estimator

$$\frac{\sum_{x,y} \hat{u}(t, x, y)}{\sum_{x,y} \hat{\ell}(t, x, y)}$$

for α. A corresponding "plug-in" TAVC estimator can be implemented in a straightforward fashion when $|A| < \infty$. The plug-in estimator takes advantage of the fact that the TAVC estimator for the A-valued discrete-time Markov chain $(X(\beta_i)\colon i \geqslant 1)$ can be computed easily (by solving a linear system of $|A|$ equations in $|A|$ unknowns). This new estimator for the TAVC associated with simulation of X over $[0, t]$ is essentially the semi-regenerative TAVC estimator proposed by Calvin et al. (2006).

10 Efficiency improvement via regeneration: Computing steady state gradients

In many applications settings, it is of interest to compute the sensitivity of the system's performance to perturbations in an underlying parameter. For example, it may be that the arrival rate to a queue is only approximately known, so that computing the change in performance that corresponds to changing the arrival rate is relevant. In particular, computing the derivative (or, more generally, the gradient) of a steady-state performance measure with respect to the arrival rate is a computational problem of significant importance. Of course, such derivatives also play a key role in both simulation-based stochastic optimization and statistical analysis of complex stochastic systems; see Glynn (1990).

To be more precise, suppose that the probability distribution underlying the simulation of V depends on a parameter $\theta \in \mathbb{R}^d$. Let P_θ be the distribution corresponding to θ. Then, the steady-state mean of V depends on θ, so that $\alpha = \alpha(\theta)$. As noted earlier, the time average $\overline{V}(t)$ satisfies

$$\overline{V}(t) \Rightarrow \alpha(\theta)$$

under P_θ, suggesting that

$$\mathrm{E}_\theta \overline{V}(t) \to \alpha(\theta). \tag{24}$$

In significant generality, there exists a random process $L(\theta, t)$ (typically, a martingale), known as the "likelihood ratio process", such that

$$\mathrm{E}_\theta \overline{V}(t) = \mathrm{E}_{\theta_0} \overline{V}(t) L(\theta, t) \tag{25}$$

for $t \geqslant 0$, where $\mathrm{E}_{\theta_0}(\cdot)$ is the expectation operator corresponding to P_{θ_0}. Assuming that the gradient can be interchanged with the expectation operator $\mathrm{E}_{\theta_0}(\cdot)$, we find that

$$\nabla \mathrm{E}_\theta \overline{V}(t)|_{\theta=\theta_0} = \mathrm{E}_{\theta_0} \overline{V}(t) \nabla L(\theta, t)|_{\theta=\theta_0}.$$

In view of (24), it seems reasonable to expect that

$$\nabla \mathrm{E}_\theta \overline{V}(t)|_{\theta=\theta_0} \to \nabla\alpha(\theta_0)$$

as $t \to \infty$, so that

$$\mathrm{E}_{\theta_0} \overline{V}(t) \nabla L(\theta_0, t) \to \nabla\alpha(\theta_0)$$

as $t \to \infty$. In particular, assume that a bias expansion similar to that derived in Section 5 holds, so that

$$\mathrm{E}_{\theta_0} \overline{V}(t) \nabla L(\theta_0, t) = \nabla\alpha(\theta_0) + \frac{1}{t}\nabla b(\theta_0) + \mathrm{o}\left(\frac{1}{t}\right) \tag{26}$$

as $t \to \infty$. On the other hand, $(L(\theta, t)\colon\ t \geqslant 0)$ is typically a martingale process for each θ. Given that $(h^{-1}(L(\theta_0 + he_i, t) - L(\theta_0, t))\colon\ t \geqslant 0)$ is then a martingale for each unit vector e_i, one expects $(\nabla L(\theta_0, t)e_i\colon\ t \geqslant 0)$ to be a martingale as well. The martingale CLT (see, for example, p. 476 of Billingsley, 1995) then suggests that there exists a (deterministic) constant $\gamma \in (0, \infty)$ such that

$$t^{-1/2}\nabla L(\theta_0, t)e_i \Rightarrow \gamma\,\mathrm{N}(0, 1)$$

as $t \to \infty$. Slutsky's lemma then implies the weak convergence statement

$$t^{-1/2}\overline{V}(t)\nabla L(\theta_0, t)e_i \Rightarrow \alpha\gamma\,\mathrm{N}(0, 1)$$

as $t \to \infty$, so that we expect

$$t^{-1}\operatorname{var}\overline{V}(t)\nabla L(\theta_0, t)e_i \to \alpha^2\gamma^2\operatorname{var}\mathrm{N}(0, 1) \tag{27}$$

as $t \to \infty$.

For a given computer budget c, how many independent replications m of length c/m should one simulate to minimize the mean square error of the resulting estimator? Note that the bias of each replication is then of order m/c (in view of (26)). The variance of a replication of length c/m is of order c/m (see (27)). So, the sample mean over m such replications has a variance of order c/m^2. The value of m that minimizes the mean square error of the corresponding gradient estimator is then of order $c^{3/4}$, yielding an estimator with a root mean square error of order $c^{-1/4}$.

However, if regenerative structure can be identified algorithmically, then a different means of estimating $\nabla\alpha(\theta_0)$ is available to simulationist. If the process V is regenerative under P_θ, the ratio formula

$$\alpha(\theta) = \frac{\mathrm{E}_\theta \int_0^{\tau_1} V(s)\,\mathrm{d}s}{\mathrm{E}_\theta \tau_1} \tag{28}$$

holds (provided that the process is initiated with a regeneration at $t = 0$, so that $T(0) = 0$). Assuming the existence of a likelihood ratio process $(L(\theta, t)\colon\ t \geqslant 0)$, we expect to be able to extend the identity (25) separately to the numerator and denominator of (28) to

$$\alpha(\theta) = \frac{\mathrm{E}_{\theta_0} \int_0^{\tau_1} V(s)\,\mathrm{d}s\, L(\theta, \tau_1)}{\mathrm{E}_{\theta_0} \tau_1 L(\theta, \tau_1)}. \tag{29}$$

Assuming that the gradient operator can be interchanged with the expectation operator $\mathrm{E}_{\theta_0}(\cdot)$ (as above), we find that

$$\nabla\alpha(\theta_0) = \frac{\mathrm{E}_{\theta_0}\big[\int_0^{\tau_1} (V(s) - \alpha(\theta_0))\,\mathrm{d}s\, \nabla L(\theta, \tau_1)\big]}{\mathrm{E}_{\theta_0} \tau_1}. \tag{30}$$

Consequently, $\nabla\alpha(\theta_0)$ can be expressed as $\nabla\alpha(\theta_0) = k(\mathrm{E}\xi)$, where

$$\xi = \left(\int_0^{\tau_1} V(s)\,\mathrm{d}s, \tau_1, \int_0^{\tau_1} V(s)\,\mathrm{d}s\, \nabla L(\theta_0, \tau_1), \tau_1 \nabla L(\theta_0, \tau_1)\right)$$

and

$$k(x_1, x_2, x_3, x_4) = \frac{x_3 - (x_1/x_2)x_4}{x_2}.$$

Suppose that one simulates n i.i.d. cycles of v under the probability P_{θ_0}, thereby obtaining n independent copies $\xi_1, \xi_2, \ldots, \xi_n$ of the random vector ξ. The estimator $\nabla\alpha_n(\theta_0) \triangleq k(n^{-1} \sum_{i=1}^n \xi_i)$ can then be analyzed via "delta-method" type arguments (see, for example, Serfling, 1980) to establish that the estimator converges at rate $n^{-1/2}$ (and hence, in units of computer time, at rate $c^{-1/2}$) to the gradient $\nabla\alpha(\theta_0)$. This $c^{-1/2}$ convergence rate is to be contrasted with the $c^{-1/4}$ rate observed earlier, and shows clearly that regenerative structure can be usefully exploited in obtaining substantial efficiency improvements.

11 Efficiency improvement via regeneration: Computing infinite horizon discounted reward

We now offer a second illustration of the principle that the presence of regenerative structure can be usefully exploited to obtain efficiency improvements. Consider the infinite horizon expected discounted reward $\alpha = \mathrm{E}D$, where

$$D = \int_0^\infty \mathrm{e}^{-rt} V(t)\,\mathrm{d}t$$

for some $r > 0$. From a computational standpoint, an algorithm based on simulating i.i.d. copies of the r.v. D cannot be operationalized, because it takes infinite time to generate the above r.v. As a consequence, one needs to consider computationally feasible alternatives.

One such approach is to exploit regeneration. In particular, suppose that V is regenerative with regeneration times $0 = T(0) < T(1) < \cdots$ (so that

V regenerates at $T = 0$) that split the sample path of V into i.i.d. cycles. Then,

$$\begin{aligned}\alpha &= \mathrm{E}\int_0^\infty \mathrm{e}^{-rt}V(t)\,\mathrm{d}t \\ &= \mathrm{E}\int_0^{\tau_1} \mathrm{e}^{-rt}V(t)\,\mathrm{d}t + \mathrm{E}\mathrm{e}^{-r\tau_1}\int_0^\infty \mathrm{e}^{-rt}V(\tau_1 + t)\,\mathrm{d}t \\ &= \mathrm{E}\int_0^{\tau_1} \mathrm{e}^{-rt}V(t)\,\mathrm{d}t + \mathrm{E}\mathrm{e}^{-r\tau_1}\alpha\end{aligned}$$

so that $\alpha = k(\mathrm{E}\xi)$, where

$$\xi = \left(\int_0^{\tau_1} \mathrm{e}^{-rt}V(t)\,\mathrm{d}t, \mathrm{e}^{-r\tau_1}\right)$$

and $k(x_1, x_2) = x_1/(1 - x_2)$.

As in Section 10, the corresponding regenerative estimator for α is $\alpha_n = k(n^{-1}\sum_{i=1}^n \xi_i)$, where $\xi_1, \xi_2, \ldots, \xi_n$ are i.i.d. copies of ξ obtained by simulating n independent cycles of V. Note that the estimator for α can be computed in finite time and is computationally feasible. Furthermore, the delta method again applies, yielding the conclusion that α_n typically converges to α at rate $c^{-1/2}$ in the computer budget c. Thus, use of regeneration in this setting makes feasible and practical a computation that is problematic in its original form.

References

Anderson, T.W. (1971). *The Statistical Analysis of Time Series*. Wiley, New York.

Andradóttir, S., Calvin, J.M., Glynn, P.W. (1995). Accelerated regeneration for Markov chain simulations. *Probability in the Engineering and Informational Sciences* 9, 497–523.

Billingsley, P. (1995). *Probability and Measure*, 3rd edition. Wiley, New York.

Breiman, L. (1968). *Probability*. *Addison–Wesley Series in Statistics*. Addison–Wesley, Reading, MA.

Calvin, J. (1994). Return state independent quantities in regenerative simulation. *Operations Research* 42, 531–542.

Calvin, J.M., Glynn, P.W., Nakayama, M.K. (2006). The semi-regenerative method of simulation output analysis. *ACM Transactions on Modeling and Computer Simulation*, in press.

Cox, D., Smith, W. (1961). *Queues*. Methuen, London.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton, NJ.

Crane, M., Iglehart, D. (1974a). Simulating stable stochastic systems, I: General multi-server queues. *Journal of the Association for Computing Machinery* 21, 103–113.

Crane, M., Iglehart, D. (1974b). Simulating stable stochastic systems, II: Markov chains. *Journal of the Association for Computing Machinery* 21, 114–123.

Crane, M., Iglehart, D. (1975). Simulating stable stochastic systems, III: Regenerative processes and discrete-event simulations. *Operations Research* 23, 33–45.

Efron, B., Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, New York.

Ethier, S.N., Kurtz, T.G. (1986). *Markov Processes: Characterization and Convergence*. *Wiley Series in Probability and Statistics*. Wiley, New York.

Fishman, G. (1973). Statistical analysis for queueing simulations. *Management Science* 20, 363–369.

Fishman, G. (1974). Estimation in multiserver queueing simulations. *Operations Research* 22, 72–78.

Glynn, P.W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM* 33, 75–84.

Glynn, P.W. (1994). Some topics in regenerative steady-state simulation. *Acta Applicandae Mathematicae* 34 (1/2), 225–236.

Glynn, P.W., Haas, P.J. (2006). A law of large numbers and functional central limit theorem for generalized semi-Markov processes. *Stochastic Models*, in press.

Glynn, P.W., Heidelberger, P. (1990). Bias properties of budget constrained simulation. *Operations Research* 38 (5), 801–814.

Glynn, P.W., Heidelberger, P. (1992). Jackknifing under a budget constraint. *ORSA Journal on Computing* 4 (3), 226–234.

Glynn, P.W., Iglehart, D.L. (1986). Recursive moment formulas for the regenerative method of simulation. In: Janssen, J. (Ed.), *Semi-Markov Models*. Plenum Press, New York.

Glynn, P.W., Iglehart, D.L. (1987). A joint central limit theorem for the sample mean and regenerative variance estimator. *Annals of Operations Research* 8, 41–55.

Glynn, P.W., Iglehart, D.L. (1993). Conditions for the applicability of the regenerative method. *Management Science* 39, 1108–1111.

Glynn, P.W., Whitt, W. (1993). Limit theorems for cumulative processes. *Stochastic Processes and Their Applications* 47, 299–314.

Glynn, P.W., Whitt, W. (2002). Necessary conditions in limit theorems for cumulative processes. *Stochastic Processes and Their Applications* 98 (2), 199–209.

Henderson, S.G., Glynn, P.W. (1999). Can the regenerative method be applied to discrete-event simulation? In: Farrington, P.A., Nembhard, H.B., Evans, J., Sturrock, D. (Eds.), *Proceedings of the 1999 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 367–373.

Henderson, S.G., Glynn, P.W. (2001). Regenerative steady-state simulation of discrete-event systems. *ACM Transactions on Modeling and Computer Simulation* 11 (4), 313–345.

Ibragimov, I.A., Has'minskii, R.Z. (1981). *Statistical Estimation – Asymptotic Theory*. *Applications of Mathematics*. Springer-Verlag, New York.

Kabak, I. (1968). Stopping rules for queueing simulations. *Operations Research* 16, 431–437.

Karlin, S., Taylor, H.M. (1981). *A Second Course in Stochastic Processes*. Academic Press, New York.

Lewis, P.A.W., Orav, E.J. (1989). *Simulation Methodology for Statisticians, Operations Analysts, and Engineers*. *Wadsworth & Brooks/Cole Statistics/Probability Series*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Meketon, M.S., Heidelberger, P. (1982). A renewal theoretic approach to bias reduction in regenerative simulations. *Management Science* 28 (2), 173–181.

Meyn, S., Tweedie, R. (1993). Criteria for stability of Markovian processes III: Foster–Lyapunov criteria for continuous time processes, with examples. *Advances in Applied Probability* 25, 518–548.

Miller, R.G. (1974). The jackknife – A review. *Biometrika* 61 (1), 1–15.

Serfling, R.J. (1980). *Approximation Theorems of Mathematical Statistics*. *Wiley Series in Probability and Statistics*. Wiley, New York.

Sigman, K. (1990). One-dependent regenerative processes and queues in continuous time. *Mathematics of Operations Research* 15, 175–189.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13017-0

Chapter 17

Selecting the Best System

Seong-Hee Kim

School of Industrial & Systems Engineering, Georgia Institute of Technology, USA
E-mail: skim@isye.gatech.edu

Barry L. Nelson

Department of Industrial Engineering & Management Sciences, Northwestern University, USA
E-mail: nelsonb@northwestern.edu

Abstract

We describe the basic principles of ranking and selection, a collection of experiment-design techniques for comparing "populations" with the goal of finding the best among them. We then describe the challenges and opportunities encountered in adapting ranking-and-selection techniques to stochastic simulation problems, along with key theorems, results and analysis tools that have proven useful in extending them to this setting. Some specific procedures are presented along with a numerical illustration.

1 Introduction

Over the last twenty years there has been considerable effort expended to develop statistically valid ranking-and-selection (R&S) procedures to compare a finite number of simulated alternatives. There exist at least four classes of comparison problems that arise in simulation studies: selecting the system with the largest or smallest expected performance measure (selection of the best), comparing all alternatives against a standard (comparison with a standard), selecting the system with the largest probability of actually being the best performer (multinomial selection), and selecting the system with the largest probability of success (Bernoulli selection). For all of these problems, a constraint is imposed either on the probability of correct selection (PCS) or on the simulation budget. Some procedures find a desirable system with a guarantee on the PCS, while other procedures maximize the PCS under the budget constraint. Our focus in this chapter is on selection-of-the-best problems with a PCS constraint. A good procedure is one that delivers the desired PCS efficiently (with minimal simulated data) and is robust to modest violations of its underlying assumptions. Other types of comparison problems and procedures will be discussed briefly in Section 7. In this chapter "best" means maximum expected value of performance, such as expected throughput or profit.

Traditional roles for R&S are selecting the best system from among a (typically small) number of simulated alternatives and screening a relatively large number of simulated alternatives to quickly discard those whose performance is clearly inferior to the best. More recently, R&S procedures are playing an important role in optimization via simulation. Many algorithms for optimization via simulation search the feasible solution space by some combination of randomly sampling solutions and exploring the neighborhood of good solutions (see Chapters 18–21). R&S procedures can be embedded within these algorithms to help them make improving moves correctly and efficiently. In addition, at the end of an optimization-via-simulation search, R&S procedures can be applied to those solutions that were visited by the search to provide a statistical guarantee that the solution returned as best is at least the best of all the solutions actually simulated. See, for instance, Boesel et al. (2003) and Pichitlamken and Nelson (2001) for more on the application of R&S in this context.

Rather than present a comprehensive survey of R&S procedures, or provide a guide for applying them, our goal is to explain how such procedures are constructed, emphasizing issues that are central to designing procedures for computer simulation, and reviewing some key theorems that have proven useful in deriving procedures. We do, however, present three specific R&S procedures as illustrations. See Goldsman and Nelson (1998) and Law and Kelton (2000) for detailed "how to" guides, Bechhofer et al. (1995) for a comprehensive survey of R&S procedures and Hochberg and Tamhane (1987) or Hsu (1996) for closely related multiple comparison procedures (MCPs).

The chapter is organized as follows: In Section 2 we show how R&S procedures are derived in an easy, but unrealistic, setting. Section 3 lists the challenges and opportunities encountered in simulation problems, along with key theorems and results that have proven useful in extending R&S procedures to this setting. Three specific procedures are presented in Section 4, followed by a numerical illustration in Section 5. Section 6 reviews asymptotic analysis regimes for R&S. Section 7 describes other formulations of the R&S problem and gives appropriate references. Section 8 closes the chapter by speculating on future research directions in this area.

2 Basics of ranking and selection

In this section we employ the simplest possible setting to illustrate how R&S procedures address comparison problems. This setting (i.i.d. normal data with known, common variance) allows us to focus on key techniques before moving on to the technical difficulties that arise in designing procedures for realistic simulation problems.

R&S traces its origins to two papers: Bechhofer (1954) established the *indifference-zone formulation*, while Gupta (1956, 1965) is credited with the *subset selection formulation* of the problem. Both approaches are reviewed in this section, and both were developed to compensate for the limited inference provided by hypothesis tests for the homogeneity of k population parameters (usually means). In many experiments, rejecting the hypothesis H_0: $\mu_1 = \mu_2 = \cdots = \mu_k$, where μ_i is the parameter associated with the ith population, leads naturally to questions about which one has the largest or smallest parameter. R&S tries to answer such questions. MCPs also provide inference beyond rejection of homogeneity; there is a close connection between R&S and MCPs, as we demonstrate later.

Suppose that there are k systems. Let X_{ij} represent the jth output from system i and let $\mathbf{X}_i = \{X_{ij}; j = 1, 2, \ldots\}$ denote the output sequence from system i. In this section, we assume that the X_{ij} are i.i.d. normal with means $\mu_i = \mathrm{E}[X_{ij}]$ and variances $\sigma_i^2 = \mathrm{Var}[X_{ij}]$. Further, we assume that the processes $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ are mutually independent, and the variances are known and equal; that is, $\sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2 = \sigma^2$. These are unrealistic assumptions that will be relaxed later, but we adopt them here because we can derive R&S procedures in a way that illustrates the key issues. Throughout the chapter we assume that a larger mean is better, and we let $\mu_k \geqslant \mu_{k-1} \geqslant \cdots \geqslant \mu_1$, so that (unknown to us) system k is the best system.

2.1 Subset-selection formulation

Suppose that we have n outputs from each of the systems. Our goal is to use these data to obtain a subset $I \subseteq \{1, 2, \ldots, k\}$ such that

$$\Pr\{k \in I\} \geqslant 1 - \alpha, \tag{1}$$

where $1/k < 1-\alpha < 1$. Ideally $|I|$ is small, the best case being $|I| = 1$. Gupta's solution was to include in the set I all systems ℓ such that

$$\overline{X}_\ell(n) \geqslant \max_{i \neq \ell} \overline{X}_i(n) - h\sigma\sqrt{\frac{2}{n}}, \tag{2}$$

where $\overline{X}_i(n)$ is the sample mean of the (first) n outputs from system i, and h is a constant whose value will depend on k and $1 - \alpha$. The proof that rule (2) provides guarantee (1) is instructive and shows what the value of h should be

$$\begin{aligned}
&\Pr\{k \in I\} \\
&\quad = \Pr\left\{\overline{X}_k(n) \geqslant \max_{i \neq k} \overline{X}_i(n) - h\sigma\sqrt{\frac{2}{n}}\right\} \\
&\quad = \Pr\left\{\overline{X}_k(n) \geqslant \overline{X}_i(n) - h\sigma\sqrt{\frac{2}{n}}, \forall i \neq k\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \Pr\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} \leqslant h - \frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}, \forall i \neq k\right\} \\
&\geqslant \Pr\{Z_i \leqslant h, i = 1, 2, \ldots, k-1\} = 1 - \alpha,
\end{aligned}
$$

where $(Z_1, Z_2, \ldots, Z_{k-1})$ have a multivariate normal distribution with means 0, variances 1 and common pairwise correlations 1/2. Therefore, to provide the guarantee (1), h needs to be the $1-\alpha$ quantile of the maximum of such a multivariate normal random vector, a quantile that turns out to be relatively easy to evaluate numerically. Notice the inequality in the final step where we make use of the fact that $\mu_k \geqslant \mu_i$.

A theme that runs throughout much of R&S is first using appropriate standardization of estimators and then bounding the resulting probability statements in such a way that a difficult multivariate probability statement becomes one that is readily solvable.

2.2 Indifference-zone formulation

A disadvantage of the subset-selection procedure in Section 2.1 is that the retained set I may, and likely will, contain more than one system. However, there is no procedure that can guarantee a subset of size 1 and satisfy (1) for arbitrary n. Even when n is under our control, as it is in computer simulation, the appropriate value will depend on the true differences $\mu_k - \mu_i$, $\forall i \neq k$. To address this problem, Bechhofer (1954) suggested the following compromise: guarantee to select the single best system, k, whenever $\mu_k - \mu_{k-1} \geqslant \delta$, where $\delta > 0$ is the smallest difference the experimenter feels is worth detecting. Specifically, the procedure should guarantee

$$
\Pr\{\text{select } k | \mu_k - \mu_{k-1} \geqslant \delta\} \geqslant 1 - \alpha, \tag{3}
$$

where $1/k < 1 - \alpha < 1$. If there are systems whose means are within δ of the best, then the experimenter is “indifferent” to which of these is selected, leading to the term indifference-zone (IZ) formulation.

The procedure is as follows: From each system, take

$$
n = \left\lceil \frac{2h^2\sigma^2}{\delta^2} \right\rceil \tag{4}
$$

outputs, where h is an appropriate constant (determined below) and $\lceil x \rceil$ means to round x up; then select the system with the largest sample mean as the best. Assuming $\mu_k - \mu_{k-1} \geqslant \delta$,

$$
\begin{aligned}
&\Pr\{\text{select } k\} \\
&\quad = \Pr\{\overline{X}_k(n) > \overline{X}_i(n), \forall i \neq k\} \\
&\quad = \Pr\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} < -\frac{(\mu_i - \mu_k)}{\sigma\sqrt{2/n}}, \forall i \neq k\right\}
\end{aligned}
$$

$$\begin{aligned}
&\geqslant \Pr\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} < \frac{\delta}{\sigma\sqrt{2/n}}, \forall i \neq k\right\} \\
&\geqslant \Pr\left\{\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{\sigma\sqrt{2/n}} < h, \forall i \neq k\right\} \\
&= \Pr\{Z_i < h, i = 1, 2, \ldots, k-1\} = 1 - \alpha,
\end{aligned}$$

where again $(Z_1, Z_2, \ldots, Z_{k-1})$ has a multivariate normal distribution with means 0, variances 1, and common pairwise correlations $1/2$, implying h needs to be the $1-\alpha$ quantile of the maximum of such a multivariate normal random vector.

Notice that the first inequality results from the assumption that $\mu_k - \mu_{k-1} \geqslant \delta$, while the second occurs because $\sqrt{n} \geqslant \sqrt{2}h\sigma/\delta$. Both of these tricks are standard: the first frees the probability statement of dependence on the true means, while the second frees it of dependence on the value of the variance.

It is worth noting that, over all configurations of the true means such that $\mu_k - \mu_{k-1} \geqslant \delta$, the configuration $\mu_i = \mu_k - \delta, \forall i \neq k$ minimizes the PCS; it is therefore known as the *least-favorable configuration* (LFC). In this chapter we break from the statistics literature in that we will not be concerned with identifying the LFC; our only interest is insuring that (3) is met.

Bechhofer's procedure is essentially a power calculation: how large a sample is required to detect differences of at least δ? When true differences are greater than δ, Bechhofer's n may be much larger than needed. By taking observations and making decisions sequentially, it is often possible to reach an earlier decision. Sequential selection procedures can be traced back at least to Wald (1947), but here we present a procedure due to Paulson (1964) that better illustrates the approach that has had the most impact in computer simulation. Paulson's procedure takes observations *fully sequentially* – meaning one at a time – and *eliminates* systems from continued sampling when it is statistically clear that they are inferior. Thus, simulation for a problem with a single dominant alternative may terminate very quickly.

Using the same notation as above, let $\overline{X}_i(r)$ be the sample mean of the first r outputs of system i. At each stage $r = 1, 2, \ldots, n$, one output is taken from each system whose index is in I, where initially $I = \{1, 2, \ldots, k\}$. At stage r system ℓ is retained in I only if

$$\overline{X}_\ell(r) \geqslant \max_{i \in I} \overline{X}_i(r) - \max\left\{0, \frac{a}{r} - \lambda\right\}, \tag{5}$$

where $a > 0$ and $0 < \lambda < \delta$ are constants to be determined, and $n = \lfloor a/\lambda \rfloor$, with $\lfloor\cdot\rfloor$ meaning round down. The procedure ends when $|I| = 1$, which requires no more than $n + 1$ stages. Parallels with Gupta's subset selection and Bechhofer's IZ ranking are obvious: At each stage a subset selection is performed, with the hedging factor $(a/r - \lambda)$ decreasing as more data are obtained. In the end, if the procedure makes it that far, the system with the largest sample mean is selected.

The following result is used to establish the PCS: Suppose $Z_1, Z_2, \ldots$ are i.i.d. $\mathrm{N}(\mu, \sigma^2)$ with $\mu < 0$. Then it can be shown that

$$\Pr\left\{\overline{Z}(r) > \frac{a}{r},\ \text{for some } r < \infty\right\} \leqslant \exp\left(\frac{2\mu}{\sigma^2}a\right). \tag{6}$$

This result is a consequence of Wald's lemma (Wald, 1947, p. 146). *Large deviation results, frequently based on the analysis of approximating Brownian motion processes, are central to the design of fully sequential procedures that involve frequent looks at the data.*

The approach in this case is to bound the probability of an *incorrect selection* (ICS). An ICS event occurs if system k is eliminated at some point during the procedure. Let $\Pr\{\mathrm{ICS}_i\}$ be the probability of an incorrect selection if only systems i and k are included in the competition.

The first key inequality is

$$\Pr\{\mathrm{ICS}\} \leqslant \sum_{i=1}^{k-1} \Pr\{\mathrm{ICS}_i\}. \tag{7}$$

Decomposition into some form of paired comparisons is a key step in many sequential procedures.

This decomposition allows us to focus only on $\Pr\{\mathrm{ICS}_i\}$. Notice that

$$\begin{aligned}
\Pr\{\mathrm{ICS}_i\} &\leqslant \Pr\left\{\overline{X}_k(r) < \overline{X}_i(r) + \lambda - \frac{a}{r},\ \text{for some } r \leqslant n+1\right\} \\
&= \Pr\left\{\overline{X}_i(r) - \overline{X}_k(r) + \lambda > \frac{a}{r},\ \text{for some } r \leqslant n+1\right\} \\
&\leqslant \Pr\left\{\overline{X}_i(r) - \overline{X}_k(r) + \lambda > \frac{a}{r},\ \text{for some } r < \infty\right\} \\
&\leqslant \exp\left(\frac{(\mu_i - \mu_k + \lambda)}{\sigma^2}a\right) \\
&\leqslant \exp\left(\frac{(\lambda - \delta)}{\sigma^2}a\right).
\end{aligned}$$

The third inequality comes from the large deviation result (6), while the fourth inequality exploits the indifference-zone assumption. If we set

$$a = \ln\left(\frac{k-1}{\alpha}\right)\frac{\sigma^2}{\delta - \lambda} \tag{8}$$

then $\Pr\{\mathrm{ICS}_i\} \leqslant \alpha/(k-1)$ and

$$\Pr\{\mathrm{ICS}\} \leqslant (k-1)\frac{\alpha}{(k-1)} = \alpha.$$

2.3 Connection to multiple comparisons

MCPs approach the comparison problem by providing simultaneous confidence intervals on selected differences among the systems' parameters. Hochberg and Tamhane (1987) and Hsu (1996) are good comprehensive references. As noted by Hsu (1996, pp. 100–102), the connection between R&S and MCPs comes through multiple comparisons with the best (MCB). MCB forms simultaneous confidence intervals for $\mu_i - \max_{\ell \neq i} \mu_\ell$, $i = 1, 2, \ldots, k$, the difference between each system and the best of the rest. Specialized to the known-variance case, the intervals take the form

$$\begin{aligned}\mu_i - \max_{\ell \neq i} \mu_\ell \in \Bigg[&-\left(\overline{X}_i(n) - \max_{\ell \neq i} \overline{X}_\ell(n) - h\sigma\sqrt{\frac{2}{n}} \right)^-, \\ &\left(\overline{X}_i(n) - \max_{\ell \neq i} \overline{X}_\ell(n) + h\sigma\sqrt{\frac{2}{n}} \right)^+ \Bigg],\end{aligned} \tag{9}$$

where h is the same critical value used in Bechhofer's and Gupta's procedures, $-x^- = \min\{0, x\}$ and $x^+ = \max\{0, x\}$. Under our assumptions these k confidence intervals are simultaneously correct with probability greater than or equal to $1 - \alpha$.

Consider the set I containing the indices of all systems whose MCB upper confidence bound is greater than 0. Thus, for $i \in I$,

$$\overline{X}_i(n) > \max_{\ell \neq i} \overline{X}_\ell(n) - h\sigma\sqrt{\frac{2}{n}},$$

meaning these are the same systems that would be retained by Gupta's subset-selection procedure. Since $\mu_k - \max_{\ell \neq k} \mu_\ell > 0$ and these intervals are simultaneously correct with probability greater than or equal to $1 - \alpha$, system k will be in the subset identified by the MCB upper bounds with the required probability.

Now suppose that n has been selected such that $n \geqslant 2h^2\sigma^2/\delta^2$, implying that

$$h\sigma\sqrt{\frac{2}{n}} \leqslant \delta$$

as in Bechhofer's procedure. Let B be the index of the system with the largest sample mean. Then the MCB lower bounds guarantee with probability greater than or equal to $1 - \alpha$ that

$$\begin{aligned}\mu_B - \max_{\ell \neq B} \mu_\ell &\geqslant -\left(\overline{X}_B(n) - \max_{\ell \neq B} \overline{X}_\ell(n) - h\sigma\sqrt{\frac{2}{n}} \right)^- \\ &\geqslant -\delta.\end{aligned}$$

The final inequality follows because $\overline{X}_B(n) - \max_{\ell \neq B} \overline{X}_\ell(n) \geqslant 0$ by the definition of B, and $h\sigma\sqrt{2/n} \leqslant \delta$ because of our choice of n. This establishes that the system selected by Bechhofer's procedure is guaranteed to be within δ of the true best, with probability greater than or equal to $1-\alpha$, *under any configuration of the means*. Further, if $\mu_k - \mu_{k-1} > \delta$, then $\Pr\{B = k\} \geqslant 1-\alpha$ as required.

As a consequence of this analysis both Bechhofer's and Gupta's procedures can be augmented with MCB confidence intervals "for free", and Bechhofer's procedure is guaranteed to select a system within δ of the best. Nelson and Matejcik (1995) establish very mild conditions under which these results hold for far more general R&S procedures.

3 Simulation issues and key results

In the previous section we illustrated different approaches to the R&S problem under assumptions such as independence, normality, and known and equal variances. Unfortunately, such assumptions rarely hold in simulation experiments. There are also opportunities available in simulation experiments that are not present in physical experiments. In the following subsections we describe these issues and opportunities, and present key theorems and results that have been useful in deriving R&S procedures that overcome or exploit them.

3.1 Unknown and unequal variances

Unknown and unequal variances across alternatives is a fact of life in system simulation problems, and the variances can differ dramatically. In the simple inventory model presented in Section 5 the ratio of the largest to smallest variance is almost 4.

There are many subset-selection procedures that permit an unknown, common variance (see Goldsman and Nelson, 1998, for one). When variances are unknown and unequal, however, the subset-selection problem is essentially equivalent to the famous Behrens–Fisher problem. One approach is to work with the standardized random variables

$$\frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{(S_i^2/n + S_k^2/n)^{1/2}}, \quad i = 1, 2, \ldots, k-1, \tag{10}$$

where S_i^2 is the sample variance of the outputs from system i. Neither the joint nor marginal distributions of these quantities are conveniently characterized. If we break the required joint probability statement up into statements about the individual terms, using techniques described below, then there are at least two solutions available. Welch (1938) suggested approximating each term in (10) as having a $t_{\hat{\nu}}$ distribution, where the degrees of freedom $\hat{\nu}$ is an approximation based on the values of S_i^2 and S_k^2. Banerjee (1961) proposed a probability bound that we specialize to our case:

Theorem 1 (Banerjee, 1961). *Suppose* Z *is* $\mathrm{N}(0,1)$ *and independent of* Y_i *and* Y_k, *which are themselves independent* χ^2_ν *random variables. Then for arbitrary but fixed* $0 \leqslant \gamma \leqslant 1$,

$$\Pr\left\{\frac{Z^2}{\gamma\frac{Y_i}{\nu} + (1-\gamma)\frac{Y_k}{\nu}} \leqslant t^2_{1-\alpha/2,\nu}\right\} \geqslant 1-\alpha, \tag{11}$$

where $t_{1-\alpha/2,\nu}$ *is the* $1-\alpha/2$ *quantile of the t distribution with* ν *degrees of freedom.*

To employ Banerjee's inequality in our context, identify

$$Z = \frac{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k)}{(\sigma_i^2/n + \sigma_k^2/n)^{1/2}}$$

and

$$\begin{aligned}\gamma\frac{Y_i}{\nu} + (1-\gamma)\frac{Y_k}{\nu} &= \frac{S_i^2/n + S_k^2/n}{\sigma_i^2/n + \sigma_k^2/n}\\ &= \left(\frac{\sigma_i^2}{\sigma_i^2+\sigma_k^2}\right)\frac{S_i^2}{\sigma_i^2} + \left(\frac{\sigma_k^2}{\sigma_i^2+\sigma_k^2}\right)\frac{S_k^2}{\sigma_k^2}.\end{aligned}$$

This inequality is used in Procedure NSGS presented in Section 4.

For some time it has been known that it is not possible to provide a guaranteed PCS, in the IZ sense, with a single stage of sampling when variances are unknown (see Dudewicz, 1995, for a comprehensive discussion of this result). Thus, practically useful IZ procedures work sequentially – meaning two or more stages of sampling – with the first stage providing variance estimates that help determine how much, if any, additional sampling is needed in the succeeding stages. However, one cannot simply substitute variance estimators into Bechhofer's or Paulson's procedures and hope to achieve a guaranteed PCS. Instead, the uncertainty in the variance estimators enters into the determination of the sample sizes, invariably leading to more sampling than would take place if the variances were known.

A fundamental result in parametric statistics is the following: If $X_1, X_2, \ldots, X_n$ are i.i.d. $\mathrm{N}(\mu, \sigma^2)$, then $\overline{X}$ and S^2 are independent random variables. The result extends in the natural way to random vectors $\mathbf{X}_j$ that are multivariate normal. An extension of a different sort, due to Stein (1945), is fundamental to R&S procedures with unknown variances:

Theorem 2 (Stein, 1945). *Suppose* $X_1, X_2, \ldots, X_n$ *are i.i.d.* $\mathrm{N}(\mu, \sigma^2)$, *and* S^2 *is* $\sigma^2\chi^2_\nu/\nu$ *and independent of* $\sum_{i=1}^n X_j$ *and of* $X_{n+1}, X_{n+2}, \ldots$.

(1) *If* $N \geqslant n$ *is a function only of* S^2 *then*

$$\frac{\overline{X}(N) - \mu}{S/\sqrt{N}} \sim t_\nu. \tag{12}$$

(2) *If* $\xi > 0$ *and*

$$N = \max\left\{\left\lceil \frac{S^2}{\xi^2} \right\rceil, n+1\right\}$$

then, for any weights $w_1, w_2, \ldots, w_N$ *satisfying* $\sum_{j=1}^{N} w_j = 1, w_1 = w_2 = \cdots = w_n$, *and* $S^2 \sum_{j=1}^{N} w_j^2 = \xi^2$, *we have*

$$\frac{\sum_{j=1}^{N} w_j X_j - \mu}{\xi} \sim t_\nu. \tag{13}$$

In the usual case where S^2 is the sample variance of the first n observations, $\nu = n - 1$. The importance of this result in R&S is that it allows determination of a sample size large enough to attain the desired power against differences of at least δ without requiring knowledge of the process variance.

If comparisons of only $k = 2$ systems were necessary, then Stein's result would be enough (at least in the i.i.d. normal case). But our problem is multivariate and requires joint probability statements about

$$\frac{\overline{X}_i(N_i) - \overline{X}_k(N_k) - (\mu_i - \mu_k)}{\mathcal{S}_{ik}}, \quad i = 1, 2, \ldots, k-1, \tag{14}$$

where $\mathcal{S}_{ik}^2$ is a variance estimate of the difference between systems i and k based on an initial sample of size (say) n, and N_i and N_k are the final sample sizes from systems i and k. The joint distribution of these random variables is quite messy in general, even if all systems are simulated independently (as we assume in this section). One approach is to condition on $\mathcal{S}_{ik}$ and $\overline{X}_k(N_k)$ and apply inequalities such as the following to bound the joint probability:

Theorem 3 (Kimbal, 1951). *Let* $V_1, V_2, \ldots, V_k$ *be independent random variables, and let* $g_j(v_1, v_2, \ldots, v_k)$, $j = 1, 2, \ldots, p$, *be nonnegative, real-valued functions, each one nondecreasing in each of its arguments. Then*

$$\mathrm{E}\left[\prod_{j=1}^{p} g_j(V_1, V_2, \ldots, V_k)\right] \geqslant \prod_{j=1}^{p} \mathrm{E}\big[g_j(V_1, V_2, \ldots, V_k)\big].$$

Kimball's theorem is actually only for the case $k = 1$; see Hochberg and Tamhane (1987) for the extension.

Theorem 4 (Slepian, 1962). *Let* $(Z_1, Z_2, \ldots, Z_k)$ *have a* k*-variate normal distribution with zero mean vector, unit variances and correlation matrix* $\mathbf{R} = \{\rho_{ij}\}$. *Let* $\xi_1, \xi_2, \ldots, \xi_k$ *be some constants. If all the* $\rho_{ij} \geqslant 0$, *then*

$$\Pr\left\{\bigcap_{i=1}^{k} (Z_i \leqslant \xi_i)\right\} \geqslant \prod_{i=1}^{k} \Pr\{Z_i \leqslant \xi_i\}.$$

Notice that, conditional on the $\mathcal{S}_{ik}^2$, the terms in (14) are positively correlated (due to the common $\overline{X}_k(N_k)$ term), providing the opening to apply Slepian's inequality. Kimball's inequality then can be applied to simplify the unconditioning on $\mathcal{S}_{ik}^2$. Both of these ideas are employed in the design of Procedure NSGS in Section 4.

3.2 Initial sample size problem

When variances are unknown, then at least two stages of sampling are required to deliver a guaranteed PCS. In a typical two-stage R&S procedure, such as Rinott's (1978) procedure, the total sample size required of, say, system i is

$$N_i = \max\left\{n_0, \left\lceil \left(\frac{hS_i}{\delta}\right)^2 \right\rceil\right\}, \tag{15}$$

where $h = h(k, 1-\alpha, n_0)$ is a constant determined by k, the number of systems being compared; $1-\alpha$, the desired confidence level; and n_0, the number of first-stage observations used to produce the variance estimator, S_i^2. The constant h increases in k, and decreases in α and n_0. The experiment design factor that is under our control is n_0.

Figure 1 presents the typical form of $\mathrm{E}[N_i]$ as a function of n_0. The figure shows that increasing n_0, up to a point, decreases $\mathrm{E}[N_i]$, but if n_0 is too large then more data are obtained in the first stage than required to deliver the PCS guarantee. Unfortunately, the location of the minimizing value of n_0 depends on the unknown variance. Nevertheless, it is clear that there is a huge penalty for selecting n_0 too small, which forces an excessive second-stage sample to

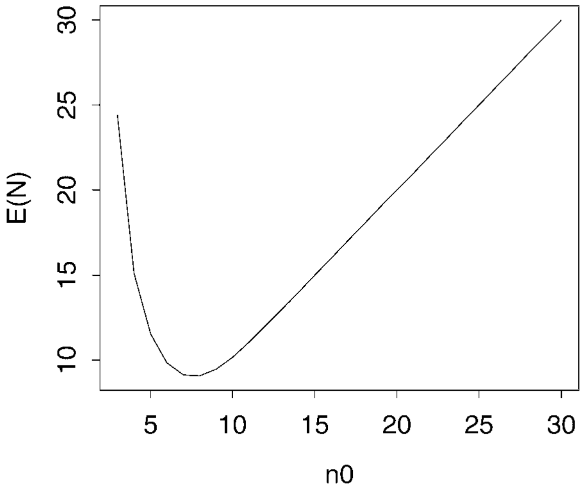


Fig. 1. Illustration of the impact of n_0 on $\mathrm{E}[N]$.

compensate for the highly unstable variance estimator. Taking $n_0 \geqslant 10$ is a common recommendation.

3.3 Nonnormality of output data

Raw output data from industrial and service simulations are rarely normally distributed. Surprisingly, nonnormality is usually not a concern in simulation experiments that (a) are designed to make multiple independent replications, and (b) use a within-replication average of a large numbers of raw simulation outputs as the basic summary measure. This is frequently the situation for so-called "terminating simulations" in which the initial conditions and stopping time for each replication are an inherent part of the definition of the system. A standard example is a store that opens empty at 6 a.m., then closes when the last customer to arrive before 9 p.m. leaves the store. If the output of interest is the average customer delay in the checkout line over the course of the day, and comparisons will be based on the expected value of this average, and the average is over many individual customer delays, then the Central Limit Theorem suggests that the replication averages will be approximately normally distributed.

Difficulties arise in so-called "steady-state simulations" where the parameter of interest is defined by a limit as the time index of a stochastic process approaches infinity (and therefore forgets its initial conditions). Some steady-state simulations are amenable to multiple replications of each alternative and within-replication averages as summary statistics, in which case the preceding discussion applies. Unfortunately, severe estimator bias due to residual effects of the initial conditions sometimes force an experiment design consisting of a single, long replication from each alternative. The raw outputs within each replication are typically neither normally distributed nor independent. For example, waiting times of individual customers in a queueing system are usually dependent because a long delay for one customer tends to increase the delays of the customers who follow. The best we can hope for is an approximately stationary output process from each system, but neither normality nor independence.

The most common approach for dealing with this problem is to transform the raw data into *batch means*, which are averages of large numbers of raw outputs. The batch means are often far less dependent and nonnormal than the raw output data. There are problems with the batching approach for R&S, however. If a "stage" is defined by batch means rather than raw output, then the simulation effort consumed by a stage is a multiple of the batch size. When a large batch size is required to achieve approximate independence – and batch sizes of several thousand are common – then the selection procedure is forced to make decisions at long intervals, wasting outputs and time. This inefficiency becomes serious when fully sequential procedures are employed because the elimination decisions for clearly inferior systems must wait for an entire batch to be formed. Therefore, for steady-state simulations, selection procedures that use individual raw outputs as basic observations are desirable.

Although no known procedures provide a guaranteed PCS for single-replication designs, some procedures have shown good empirical performance (e.g., Sullivan and Wilson, 1989), while others have been shown to be asymptotically valid (e.g., Procedure $\mathcal{KN}$++ in Section 4). See Law and Kelton (2000) or Chapter 15 for a general discussion of replications versus batching, Glynn and Iglehart (1990) for conditions under which the batch means method is asymptotically valid for confidence intervals and Section 6 for a review of asymptotic analysis of R&S procedures.

3.4 Common random numbers

The procedures described in Section 2 assume that data across the k alternative systems are independent. In simulation experiments this assumption can be made valid by using different sequences of random numbers to drive the simulation of each system (see Chapter 3). However, since we are making comparisons, there is a potential advantage of using common random numbers (CRN) to drive the simulation of each system because

$$\mathrm{Var}[X_{ij} - X_{\ell j}] = \mathrm{Var}[X_{ij}] + \mathrm{Var}[X_{\ell j}] - 2\,\mathrm{Cov}[X_{ij}, X_{\ell j}].$$

If implemented correctly (see Banks et al., 2005), CRN tends to make $\mathrm{Cov}[X_{ij}, X_{\ell j}] > 0$ thereby reducing the variance of the difference.

R&S procedures often need to make probability statements about the collection of random variables

$$\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k), \quad i = 1, 2, \ldots, k-1. \tag{16}$$

The appearance of the common term $\overline{X}_k(n)$ causes dependence among these random variables, but it is often easy to model or tightly bound. The introduction of CRN induces dependence between $\overline{X}_i(n)$ and $\overline{X}_k(n)$ as well. Even though the sign of the induced covariance is believed known, its value is not, making it difficult to say anything about the dependence among the differences (16).

Two approaches are frequently used. The first is to replace the basic data $\{X_{ij};\, i = 1, 2, \ldots, k;\, j = 1, 2, \ldots, n\}$ with pairwise differences $\{X_{ij} - X_{\ell j};\, i \neq \ell;\, j = 1, 2, \ldots, n\}$ because the variance of the sample mean of the difference includes the effect of the CRN-induced covariance. The second is to apply the Bonferroni inequality to break up joint statements about (16) into statements about the individual terms. Recall that for events $\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_{k-1}$, the Bonferroni inequality states that

$$\Pr\left\{\bigcap_{i=1}^{k-1} \mathcal{E}_i\right\} \geqslant 1 - \sum_{i=1}^{k-1} \Pr\{\mathcal{E}_i^{\mathrm{c}}\}. \tag{17}$$

In the R&S context $\mathcal{E}_i$ corresponds to an event like $\{\overline{X}_i(n) - \overline{X}_k(n) - (\mu_i - \mu_k) \leqslant h\}$.

Approaches based on the Bonferroni inequality make no assumption about the induced dependence, and therefore are very conservative. A more aggressive approach is to assume some structure for the dependence induced by CRN. One standard assumption is that all pairwise correlations $\rho = \mathrm{Corr}[X_{ij}, X_{\ell j}]$ are positive, and identical, and all variances are equal; this is known as *compound symmetry*. Nelson and Matejcik (1995) extended Rinott's (1978) procedure – one of the simplest and most popular IZ procedures – in conjunction with CRN under a more general structure called *sphericity*. The specific assumption is

$$\mathrm{Cov}[X_{ij}, X_{\ell j}] = \begin{cases} 2\beta_i + \tau^2, & i = \ell, \\ \beta_i + \beta_\ell, & i \neq \ell, \end{cases} \tag{18}$$

with $\tau^2 > 0$, which is equivalent to assuming that $\mathrm{Var}[X_{ij} - X_{\ell j}] = 2\tau^2$ for all $i \neq \ell$, a type of variance balance. This particular structure is useful because there exists an estimator $\hat{\tau}^2$ of τ^2 that is independent of the sample means and has a χ^2 distribution (allowing a pivotal quantity to be formed and Stein's theorem to be applied). Nelson and Matejcik (1995) showed that procedures based on this assumption are robust to departures from sphericity, at least in part because assuming sphericity is like assuming that all pairwise correlations equal the average pairwise correlation.

3.5 The sequential nature of simulation

Suppose an IZ ranking procedure is applied in the study of k new blood pressure medications. Then "replications" correspond to patients, and the idea of using a fully sequential procedure (assign one patient at a time to each drug, then wait for the results before recruiting the next patient) seems absurd. In simulation experiments, however, data are naturally generated sequentially, at least within each simulated alternative, making multi-stage procedures much more attractive. However, there are some issues:

- In multiple-replication designs, sequential sampling is particularly attractive. All that needs to be retained to start the next stage of sampling is the ending random number seeds from the previous stage. In single-replication designs it can be more difficult to resume sampling from a previous stage, since the entire state of the system must be retained and restored.
- A hidden cost of using multi-stage procedures is the computational overhead in switching among the simulations of the k alternatives. On a single-processor computer, switching can involve saving output, state and seed information from the current system; swapping the program for the current system out of, and for the next system into, active memory; and restoring previous state and seed information for the next

system. Thus, the overall computation effort includes both the cost of generating simulated data and the cost of switching. Hong and Nelson (2005) look at sequential IZ procedures that attempt to minimize the total computational cost.
- If k processors are available, then an attractive option is to assign each system to a processor and simulate in parallel. This is highly effective in conjunction with R&S procedures that require little or no coordination between the simulations of each system, such as subset-selection procedures or IZ-ranking procedures that use only variance information (and not differences among the sample means). Unfortunately, a fully sequential procedure with elimination would defeat much of the benefit of parallel processing because communication among the processors is required after generating each output.

Many sequential procedures are based on results for Brownian motion processes. Let $\mathcal{B}(t; \Delta)$ be a standard Brownian motion process with drift Δ. Consider the partial sum of the pairwise difference $D_i(r) = \sum_{j=1}^{r}(X_{kj} - X_{ij})$, $r = 1, 2, \ldots$. If the X_{ij} are i.i.d. normal, and $\mu_k - \mu_i = \delta$, then $\{D_i(r), r = 1, 2, \ldots\} \overset{\mathcal{D}}{=} \{\sigma\mathcal{B}(t; \delta/\sigma), t = 1, 2, \ldots\}$, where $\sigma^2 = \mathrm{Var}[X_{kj} - X_{ij}]$ (with or without CRN). In other words, $D_i(r)$ is a Brownian motion process with drift observed only at discrete (integer) points in time. A great deal is known about the probability of Brownian motion processes crossing boundaries in various ways (see, for instance, Siegmund (1985) or Jennison and Turnbull (2000)); we display one specific result below. Thus, it seems natural to design R&S procedures for $\sigma\mathcal{B}(t; \delta/\sigma)$ and apply them to $D_i(r)$.

Let $c(t)$ be a symmetric (about 0) continuation region for $\sigma\mathcal{B}(t; \delta/\sigma)$, and let an incorrect selection correspond to the process exiting the region in the wrong direction (down, when the drift is positive). If $T = \inf\{t \geqslant 0: |\sigma\mathcal{B}(t; \delta/\sigma)| > c(t)\}$, then

$$\Pr\{\mathrm{ICS}_i\} = \Pr\left\{\sigma\mathcal{B}\left(T; \frac{\delta}{\sigma}\right) < 0\right\}.$$

Of course $\sigma\mathcal{B}(t; \delta/\sigma)$ is only an approximation for $D_i(r)$. However, Jennison et al. (1980) show that under very general conditions, $\Pr\{\mathrm{ICS}_i\}$ is no greater if the Brownian motion process is observed at discrete times; thus, procedures designed for $\sigma\mathcal{B}(t; \delta/\sigma)$ provide an upper bound on the probability of incorrect selection for $D_i(r)$. In conjunction with a decomposition into pairwise comparisons, as in (7), this result can be used to derive R&S procedures for $k \geqslant 2$.

Fabian (1974) tightened the triangular continuation region used by Paulson, and this was exploited by Hartmann (1988, 1991), Kim and Nelson (2001, 2006) and Hong and Nelson (2005).

Theorem 5 (Fabian, 1974). *Let $\{\mathcal{B}(t, \Delta), t \geqslant 0\}$ be a standard Brownian motion with drift $\Delta > 0$. Let*

$$l(t) = -a + \lambda t,$$
$$u(t) = a - \lambda t,$$

for some $a > 0$ and $\lambda = \Delta/(2b)$ for some positive integer b. Let $c(t)$ denote the continuation region $(l(t), u(t))$ and let T be the first time that $\mathcal{B}(t, \Delta) \notin c(t)$. Then

$$\Pr\bigl\{\mathcal{B}(T, \Delta) < 0\bigr\} \leqslant \sum_{j=1}^{b} (-1)^{j+1}\biggl(1 - \frac{1}{2}\mathcal{I}(j = b)\biggr) \exp\bigl\{-2a\lambda(2b - j)j\bigr\}.$$

Fabian's bound on Pr{ICS} is particularly useful because a is the term that depends on the sample variance (see Paulson's a in Equation (8) for intuition). Thus, appropriately standardized, $\exp(-a)$ is related to the moment generating function of a chi-squared random variable, which simplifies unconditioning on the sample variance.

3.6 Large number of alternatives

The number of alternatives of interest in simulation problems can be quite large, with 100 or more being relatively common. However, Bechhofer-like IZ procedures were developed for relatively small numbers of alternatives, say no more than 20. They can be inefficient when the number of alternatives is large because they were developed to protect against the LFC – the configuration of system means under which it is most difficult to correctly select the best – to free the procedure from dependence on the true differences among the means. The Slippage Configuration (SC), $\mu_i = \mu_k - \delta$ for $i = 1, 2, \ldots, k-1$, is known to be the LFC for many procedures.

When the number of systems is large we rarely encounter anything remotely like the SC configuration, because large numbers of alternatives typically result from taking all feasible combinations of some controllable decision variables. Thus, the performance measures of the systems are likely to be spread out, rather than all clustered near the best. Paulson-like procedures with elimination might seem to be a cure for this ill, but the inequalities used to decompose the problem of k systems into paired comparisons with system k are typically quite conservative and become much more so with increasing k (although Kim and Nelson's, 2001, fully sequential procedure $\mathcal{KN}$, described in the next section, has been shown to work well for up to $k = 500$ systems).

To overcome the inefficiency of IZ approaches for large numbers of alternatives, one idea is to try to gain the benefits of screening, as in Paulson-like procedures, but avoid the conservatism required to compensate for so many looks at the data. Nelson et al. (2001) proposed spending some of the α for incorrect selection on an initial screening stage (using a Gupta-like subset-selection procedure), and spending the remainder on a second ranking stage (using a Bechhofer-like IZ procedure). Additive and multiplicative α spending is possible, depending on the situation (see Nelson et al., 2001; Wilson, 2001). The resulting procedure, named NSGS, is presented in the next section.

This so-called "α-spending" approach – spreading the probability of incorrect selection across multiple stages – is a general-purpose tool, and there is no inherent reason to use only a single split. See Jennison and Turnbull (2000) for a thorough discussion.

4 Example procedures

In this section we present three specific procedures to illustrate the concepts described in earlier sections. The NSGS procedure, due to Nelson et al. (2001), and the $\mathcal{KN}$ procedure, due to Kim and Nelson (2001), are appropriate for terminating simulations or for steady-state simulations when multiple replications are employed. Procedure $\mathcal{KN}$++, due to Kim and Nelson (2006), is specifically designed for steady-state simulations employing a single replication from each alternative. All of the procedures employ the IZ approach and utilize elimination to gain efficiency in the case of many systems. In all three procedures variances are considered unknown and unequal.

The NSGS procedure requires that the output data from each system be i.i.d. normal, and that outputs across systems be independent, which leaves out CRN. NSGS is the combination of a Gupta-like subset-selection procedure, to reduce the number of alternatives still in play after the first stage of sampling, and a Bechhofer-like ranking procedure applied to the systems in the subset. The procedure uses α-spending between the subset selection and ranking to control the overall PCS. Banerjee's inequality allows the subset-selection procedure to handle unequal variances.

Procedure NSGS.

(1) *Setup*. Select the overall desired PCS $1-\alpha$, IZ parameter δ, and common first-stage size $n_0 \geqslant 2$. Set

$$t = t_{n_0-1,1-(1-\alpha/2)^{1/(k-1)}}$$

and obtain Rinott's constant $h = h(n_0, k, 1-\alpha/2)$ from the tables in Wilcox (1984) or Bechhofer et al. (1995). See also Table 8.3 in Goldsman and Nelson (1998).

(2) *Initialization*. Obtain n_0 outputs X_{ij}, $j = 1, 2, \ldots, n_0$, from each system i, $i = 1, 2, \ldots, k$, and let $\overline{X}_i(n_0) = n_0^{-1}\sum_{j=1}^{n_0} X_{ij}$ denote the sample

mean of the first n_0 outputs from system i. Calculate the marginal sample variances

$$S_i^2 = \frac{1}{n_0 - 1}\sum_{j=1}^{n_0}\bigl(X_{ij} - \overline{X}_i(n_0)\bigr)^2$$

for $i = 1, 2, \ldots, k$.

(3) *Subset selection*. Calculate the quantity

$$W_{i\ell} = t\left(\frac{S_i^2 + S_\ell^2}{n_0}\right)^{1/2}$$

for all $i \neq \ell$. Form the screening subset I, containing every alternative i such that $1 \leqslant i \leqslant k$ and

$$\overline{X}_i(n_0) \geqslant \overline{X}_\ell(n_0) - (W_{i\ell} - \delta)^+ \quad \text{for all } \ell \neq i.$$

(4) *Ranking*. If $|I| = 1$, then stop and return the system in I as the best. Otherwise, for all $i \in I$, calculate the second-stage sample sizes

$$N_i = \max\left\{n_0, \left\lceil \left(\frac{hS_i}{\delta}\right)^2 \right\rceil\right\},$$

where $\lceil\cdot\rceil$ is the ceiling function.

(5) Take $N_i - n_0$ additional outputs from all systems $i \in I$.

(6) Calculate the overall sample means $\overline{X}_i(N_i)$ for all $i \in I$. Select the system with the largest $\overline{X}_i(N_i)$ as best.

Nelson et al. (2001) showed that any subset-selection procedure and any two-stage IZ ranking procedure that satisfy certain mild conditions can be combined in this way while guaranteeing the overall probability of correct selection. The NGSG procedure can handle a relatively large number of systems because the first-stage screening is pretty tight. Nelson et al. (2001) provide a revised version of the NGSG procedure, the Group-Screening procedure, in which one can avoid simulating all the systems simultaneously. Boesel et al. (2003) extended the Group-Screening procedure for "clean up" after optimization via simulation.

The $\mathcal{KN}$ procedure is *fully sequential* because it takes only a single basic output from each alternative still in contention at each stage. Also, if there exists clear evidence that a system is inferior, then it will be eliminated from consideration immediately – unlike the NSGS procedure, where elimination occurs only after the first stage. $\mathcal{KN}$ also requires i.i.d. normal data, but does allow CRN. $\mathcal{KN}$ exploits the ideas of using paired differences, and controlling the Pr{ICS} on pairs to control it overall. Fabian's result is used to bound the error of a Brownian motion process that approximates each pair.

Procedure $\mathcal{KN}$.

(1) *Setup.* Select the overall desired PCS $1-\alpha$, IZ parameter δ and common first-stage sample size $n_0 \geqslant 2$. Set

$$\eta = \frac{1}{2}\left[\left(\frac{2\alpha}{k-1}\right)^{-2/(n_0-1)} - 1\right].$$

(2) *Initialization.* Let $I = \{1, 2, \ldots, k\}$ be the set of systems still in contention, and let $h^2 = 2\eta(n_0 - 1)$.

Obtain n_0 outputs X_{ij}, $j = 1, 2, \ldots, n_0$, from each system i, $i = 1, 2, \ldots, k$, and let $\overline{X}_i(n_0) = n_0^{-1}\sum_{j=1}^{n_0} X_{ij}$ denote the sample mean of the first n_0 outputs from system i.

For all $i \neq \ell$ calculate

$$S_{i\ell}^2 = \frac{1}{n_0 - 1}\sum_{j=1}^{n_0}\left(X_{ij} - X_{\ell j} - \left[\overline{X}_i(n_0) - \overline{X}_\ell(n_0)\right]\right)^2,$$

the sample variance of the difference between systems i and ℓ. Set $r = n_0$.

(3) *Screening.* Set $I^{old} = I$. Let

$$I = \Big\{i\colon i \in I^{old} \text{ and } \\ \overline{X}_i(r) \geqslant \overline{X}_\ell(r) - W_{i\ell}(r), \forall \ell \in I^{old}, \ell \neq i\Big\},$$

where

$$W_{i\ell}(r) = \max\left\{0, \frac{\delta}{2r}\left(\frac{h^2 S_{i\ell}^2}{\delta^2} - r\right)\right\}.$$

(4) *Stopping rule.* If $|I| = 1$, then stop and select the system whose index is in I as the best.

Otherwise, take one additional output $X_{i,r+1}$ from each system $i \in I$, set $r = r + 1$ and go to *Screening*.

The $\mathcal{KN}$ procedure requires simulation of all systems simultaneously and a lot of switching among them. As discussed in Section 3, the switching cost can overwhelm the sampling cost, but this has become less of an issue in modern computing environments.

Both NSGS and $\mathcal{KN}$ can be applied to steady-state simulations if one is willing to use within-replication averages or batch means as the basic observations. However, as discussed in Section 3, employing within-replication averages or batch means as basic observations may be inefficient, so it is desirable to use individual outputs from within a single replication of each system if possible. Damerdji and Nakayama (1999) developed two-stage multiple-comparison procedures to select the best system for steady-state simulation that use batch means in the first stage of sampling, but can use individual outputs thereafter.

Similarly, Goldsman et al. (2001) and Kim and Nelson (2006) proposed three R&S procedures that make a single replication from each system and use individual output as basic observations. One is a two-stage procedure based on Rinott's procedure, and the others are extensions of $\mathcal{KN}$ to steady-state simulation. One extension of $\mathcal{KN}$, called $\mathcal{KN}$++, updates the variance estimators as more outputs are available and has been shown to be highly efficient. We present the procedure below.

In $\mathcal{KN}$++, we assume that the output from each system i, X_{ij}, $j = 1, 2, \ldots,$ is a stationary stochastic process that satisfies a Functional Central Limit Theorem condition (see Kim and Nelson, 2006, for detailed conditions), and further that the systems are simulated independently. Variance estimation centers on the *asymptotic variance constant* $v_i^2 = \lim_{r\to\infty} r \operatorname{Var}[\overline{X}_i(r)]$. See Goldsman et al. (2001) and Chapter 15 for reviews of different methods for the estimation of v_i^2. $\mathcal{KN}$++ extends $\mathcal{KN}$ to steady-state simulation by replacing its first-stage variance estimator with an estimator of the appropriate asymptotic variance constant. Moreover, $\mathcal{KN}$++ updates the variance estimator as more data are obtained based on a batching sequence m_r which is an integer-valued and nondecreasing function of r. The batching sequence needs to be carefully chosen to guarantee the strong consistency of the variance estimator in use; Goldsman et al. give three examples of such batching sequences. In general, m_r satisfies the property that $m_r \to \infty$ as $r \to \infty$.

Procedure $\mathcal{KN}$++.

(1) *Setup.* Select the overall desired PCS $1 - \alpha$, indifference-zone parameter δ, common first-stage sample size $n_0 \geqslant 2$ and initial batch size $m_{n_0} < n_0$. Set $r = n_0$. Calculate

$$\eta = \frac{1}{2}\left\{\left[2\left(1 - (1-\alpha)^{1/k-1}\right)\right]^{-2/f} - 1\right\}.$$

(2) *Initialization.* Let $I = \{1, 2, \ldots, k\}$ be the set of systems still in contention, and let $h^2 = 2\eta f$, where f is a function of the number of batches, b_r that depends on the variance estimator in use.

Obtain n_0 outputs X_{ij}, $j = 1, 2, \ldots, n_0$, from each system $i = 1, 2, \ldots, k$.

(3) *Update.* If m_r has changed since the last update, then for all $i \neq \ell$, calculate $m_r V_{i\ell}^2(r)$, the sample asymptotic variance of the difference between systems i and ℓ based on b_r batches of size m_r. Update f, η and h^2.

(4) *Screening.* Set $I^{old} = I$. Let

$$I = \left\{i: i \in I^{old} \text{ and } \overline{X}_i(r) \geqslant \overline{X}_\ell(r) - W_{i\ell}(r), \forall \ell \in I^{old}, \ell \neq i\right\},$$

where

$$W_{i\ell}(r) = \max\left\{0, \frac{\delta}{2r}\left(\frac{h^2 m_r V_{i\ell}^2(r)}{\delta^2} - r\right)\right\}.$$

(5) *Stopping rule.* If $|I| = 1$, then stop and select the system whose index is in I as the best.

Otherwise, take one additional output $X_{i,r+1}$ from each system $i \in I$ and set $r = r + 1$ and go to *Update*.

Even if the output data fed to $\mathcal{KN}$++ were i.i.d. normal, the procedure does not provide a guaranteed PCS in finite samples. However, using techniques described in Section 6.1, $\mathcal{KN}$++ can be shown to guarantee PCS $\geqslant 1 - \alpha$ asymptotically.

5 Application

This section illustrates NSGS and $\mathcal{KN}$ using an (s, S) inventory system with the five inventory policies as described in Koenig and Law (1985). The goal of this study is to compare the five policies given in Table 1 and find the one with the smallest expected average cost per month for the first 30 months of operation. Table 1 also contains the expected cost (in thousands of dollars) of each policy, which can be analytically computed in this case. We set $\delta = \$1000$, $n_0 = 10$ initial replications, and $1 - \alpha = 0.95$.

Table 2 shows the results of the simulation study for each procedure, including the total number of outputs taken and the sample average cost per month for each policy. In NSGS, policies 3, 4 and 5 were eliminated after the first stage of sampling, so only policies 1 and 2 received second-stage samples. In $\mathcal{KN}$, only policies 4 and 5 were eliminated after the first stage, but the elimination of policies 3 and 1 occurred after they received 16 and 98 observations, respectively. This illustrates the value of the tighter initial screen in NSGS, which takes only one look at the data, and the potential savings from taking many looks, as $\mathcal{KN}$ does. Both procedures chose policy 2 as the best (which is in fact correct). Since the true difference is larger than δ, NSGS and $\mathcal{KN}$ will choose the true best with 95% confidence. However, in general we do not have any

Table 1.
The five alternative inventory policies

Policy i	s	S	Expected cost
1	20	40	114.176
2	20	80	112.742
3	40	60	130.550
4	40	100	130.699
5	60	100	147.382

Table 2.
Simulation results of the (s, S) inventory example

Policy i	NSGS		$\mathcal{KN}$	
	# Obs.	Average	# Obs.	Average
1	209	114.243	98	114.274
2	349	112.761	98	113.612
3	10	130.257	16	130.331
4	10	128.990	10	128.990
5	10	147.133	10	147.133
Total	588		232	

information about the true differences; therefore, the most we can conclude without prior knowledge is that policy 2 is either the true best, or has expected cost per month within \$1000 of the true best policy, with 95% confidence.

6 Asymptotic analysis

In order of importance, the key performance measures for R&S procedures are the ability to deliver the nominal PCS and the ability to deliver it efficiently. Although many procedures provide a guaranteed PCS under ideal conditions (e.g., i.i.d. normal outputs), and the expected sample size of simple procedures can be explicitly calculated, when conditions are not ideal, or when the procedure is more complex (e.g., it includes early elimination), small-sample performance may be difficult to derive. Fortunately, asymptotic analysis – driving the sample sizes to infinity – can sometimes provide meaningful insights. The power of asymptotic analysis is that many of the problem-specific details that thwart small-sample analysis wash out in the limit. Appropriate asymptotic analysis can establish conditions under which procedures work (at least approximately), and the superiority of one procedure over another. In the R&S literature there are at least three asymptotic regimes:

PCS as $\delta \to 0$. To evaluate the ability of a procedure to provide a PCS guarantee under a range of conditions, the indifference-zone parameter δ may be driven to zero. Done naively, this drives the sample sizes from all systems to infinity and the PCS to 1. Therefore, to make the analysis useful, the selection problem must become more difficult as $\delta \to 0$. We describe this approach in Section 6.1.

Efficiency as $\delta \to 0$. The indifference-zone parameter δ may also be driven to zero to evaluate the efficiency of a procedure that estimates unknown variances, relative to a corresponding known-variance procedure. To date this type of analysis has only been applied to procedures whose sample sizes are independent of the true means (that is, the procedure does not take advantage of a favorable configuration of the means, e.g., Bechhofer, 1954), so there is no need to change the selection problem as $\delta \to 0$. We briefly describe this approach in Section 6.2.

Efficiency as $(1-\alpha) \to 1$. To compare the efficiency of competing procedures, the nominal PCS may be driven to 1. This, too, will drive the sample sizes to infinity, but if the rate at which they grow can be determined then that rate can be compared to the rate achieved by other procedures. We describe this approach in Section 6.3.

See also Damerdji and Nakayama (1999) for a related asymptotic analysis of multiple-comparison procedures.

6.1 Asymptotic probability of correct selection

There is a close connection between the PCS in R&S and the power in statistical hypothesis testing. Consider a hypothesis testing problem of the form

$$\begin{aligned} &H_0\colon\ \theta = \theta_0,\\ &H_1\colon\ \theta > \theta_0. \end{aligned}$$

Suppose that the power of the test cannot be calculated explicitly. As the sample size n goes to infinity, any reasonable test has asymptotic power 1 against any fixed alternative (say, $\theta = \theta_0 + \delta$). As noted by Lehmann (1999, Section 3.3), the trick is to embed the actual situation into a suitable sequence (n, θ_n) that makes the discrimination problem more difficult as the sample size increases in such a way that a meaningful limit less than 1 is reached. A sequence that frequently works is

$$\theta_n = \theta_0 + \frac{\delta}{\sqrt{n}} + \mathrm{o}\left(\frac{1}{\sqrt{n}}\right)$$

with $\delta > 0$.

IZ R&S procedures are essentially power calculations, since their goal is to detect the best with given probability (power) when the best is at least a significant amount $\delta > 0$ better than the rest (think H_1: $\mu_k > \mu_{k-1} + \delta$). However, instead of n driving the parameter θ_n, as in the hypothesis test, it makes more sense to have $\delta \to 0$ drive N_i, the number of observations to be taken from system i; frequently $\sqrt{N_i} \propto 1/\delta$.

Mukhopadhyay and Solanky (1994) say that an IZ procedure is *asymptotically consistent* if

$$\liminf_{\delta \to 0} \mathrm{PCS} \geqslant 1 - \alpha$$

for all $\mu_1, \mu_2, \ldots, \mu_k$ such that $\mu_k - \mu_{k-1} \geqslant \delta$. For a procedure that assumes normally distributed output data, Dalal and Hall (1977) declare an IZ procedure to be *asymptotically robust* if

$$\liminf_{\delta \to 0} \inf_{F \in \mathcal{F}} \mathrm{PCS} \geqslant 1 - \alpha,$$

where $\mathcal{F}$ is a suitable family of location parameter distributions $F(x-\mu_i)$ (containing the normal distribution) with $\mu_1, \mu_2, \ldots, \mu_k$ such that $\mu_k - \mu_{k-1} \geqslant \delta$.

An example of a procedure that does not provide a guaranteed PCS for finite samples, but can be shown to be asymptotically consistent, is due to Robbins et al. (1968). Their procedure generalizes Bechhofer's (1954) known, common-variance procedure in Section 2.2 to the unknown common variance case (still assuming normally distributed data). They suggest taking N observations from each system, where

$$N = N(\delta) = \inf\left\{n \geqslant n_0\colon n \geqslant \frac{h^2 S^2(n)}{\delta^2}\right\}$$

with $n_0 \geqslant 2$ and $S^2(n)$ the usual pooled estimator of σ^2 based on n observations. Notice that the variance estimator is updated as more data are collected, which makes it impossible to establish the finite-sample PCS.

The proof of asymptotic consistency illustrates a key idea in this approach. After some manipulation one can show that

$$\text{PCS} \geqslant \mathrm{E}\left[\int_{-\infty}^{\infty} \left\{\Phi\left(y + \frac{\sqrt{N}\delta}{\sigma}\right)\right\}^{k-1} \phi(y)\,\mathrm{d}y\right], \tag{19}$$

where Φ and ϕ are the c.d.f. and p.d.f., respectively, of the standard normal distribution. Now since $\sqrt{N}\delta/\sigma \to h$ with probability 1 as $\delta \to 0$, the right-hand side of (19) converges to

$$\int_{-\infty}^{\infty} \left\{\Phi(y+h)\right\}^{k-1} \phi(y)\,\mathrm{d}y = 1 - \alpha.$$

Notice that in the limit the unknown-variance procedure behaves like Bechhofer's known-variance procedure. The asymptotic validity of $\mathcal{KN}$++ (see Section 4) is based on an analogous argument showing that as $\delta \to 0$ the (appropriately standardized) output processes behave like (known variance and drift) Brownian motion processes (Kim and Nelson, 2006). See also Damerdji et al. (1996).

6.2 Asymptotic efficiency

Let n be the sample size (per system) of a Bechhofer-like, known-variance R&S procedure, and let N be the sample size of a corresponding unknown-variance procedure where an initial n_0 observations from each system are used to estimate the unknown variance. Typically N takes the form

$$N = \max\left\{n_0, \left\lceil \left(\frac{hS}{\delta}\right)^2 \right\rceil\right\},$$

where S^2 is a pooled estimator of the unknown variance and h is an appropriately adjusted constant.

Mukhopadhyay and Solanky (1994) say that a procedure is *asymptotically first-order efficient* if

$$\lim_{\delta\to 0} \mathrm{E}\left(\frac{N}{n}\right) = 1$$

and *asymptotically second-order efficient* if $\lim_{\delta\to 0} \mathrm{E}(N - n)$ is bounded. They show that the typical procedure for which n_0 is a fixed value, the variance is estimated only once, and N grows as $1/\delta^2$ is neither asymptotically first- nor second-order efficient. However, if N grows somewhat more slowly than $1/\delta^2$ then an asymptotically first-order efficient procedure can be obtained, while asymptotic second-order efficiency typically requires that the variance estimator be updated as more data are obtained.

6.3 Asymptotic sample size

Suppose that we want to know the expected sample size of Paulson's procedure in Section 2.2. The fact that systems can be eliminated before the terminal stage implies that the expected sample size depends on the differences between the true means, and that we must account for the complication that any system has a chance to eliminate any other. However, consider what happens as we drive $(1 - \alpha) \to 1$ (the following heuristic argument is made precise by Perng, 1969):

- As $(1 - \alpha) \to 1$, the procedure stops making mistakes; the best system survives and all of the inferior systems are eliminated by the best one.
- As the sample sizes are driven to infinity, $\overline{X}_i(r)$ behaves more and more like μ_i. Thus, the stage at which system $i \neq k$ is eliminated is the first r for which

  $$\mu_i \leqslant \mu_k - \left(\frac{a}{r} - \lambda\right).$$

  This occurs (approximately) when $r_i = a/(\mu_k - \mu_i + \lambda)$.
- Recall that

  $$a = \ln\left(\frac{k-1}{\alpha}\right)\frac{\sigma^2}{\delta - \lambda}.$$

  Therefore, as $(1 - \alpha) \to 1$, the expected sample size from system $i \neq k$ is approximately

  $$r_i \approx \ln\left(\frac{k-1}{\alpha}\right)\frac{\sigma^2}{(\delta - \lambda)(\mu_k - \mu_i + \lambda)}$$

  while for $i = k$ it is

  $$r_k \approx \ln\left(\frac{k-1}{\alpha}\right)\frac{\sigma^2}{(\delta - \lambda)(\mu_k - \mu_{k-1} + \lambda)}.$$

Thus, the expected total sample size as $(1-\alpha) \to 1$, is approximately equal to $\sum_{i=1}^{k} r_i$.

Notice that the impact of the true differences $\mu_k - \mu_i$ and the choice of λ become apparent from this analysis. The growth rate of $\ln((k-1)/\alpha)$ is common to many procedures (see Dudewicz, 1969), so the differences in their asymptotic efficiency is the term that multiplies $\ln((k-1)/\alpha)$. For an example of this type of analysis for a more complex procedure see Jennison et al. (1982).

7 Other formulations

Throughout this chapter we have focused on the problem of finding the best when the best is defined as the system with the largest or smallest mean performance measure. As discussed in Section 1, there exist other types of comparison problems. Here we briefly visit each type of comparison problem and provide useful references.

7.1 Comparisons with a standard

The goal of comparison with a standard is to find systems whose expected performance measures are larger (or smaller) than a standard and, if there are any, to find the one with the largest (or smallest) expected performance. For this type of problem, each alternative needs to be compared to the standard as well as the other alternative systems, and the standard may be a known value or the expected value of a designated system (simulated or real). Such procedures first appeared in Paulson (1952) and Bechhofer and Turnbull (1978).

Clearly, the standard could be treated as just another system and the problem formulated as selection of the best. Specially tailored procedures are required when the standard is to be given special status, specifically a guarantee that no alternative will be selected unless it beats the standard substantially.

Let μ_0 denote the expected performance of the standard (which may be known or unknown), and let $\mu_1, \mu_2, \ldots, \mu_k$ be the unknown means of the alternatives, as in selection of the best. In comparisons with a standard we require

$$\Pr\{\text{select } 0 \mid \mu_0 \geqslant \mu_k\} \geqslant 1-\alpha, \tag{20}$$

$$\Pr\{\text{select } k \mid \mu_k - \mu_0 \geqslant \delta, \mu_k - \mu_{k-1} \geqslant \delta\} \geqslant 1-\alpha. \tag{21}$$

Thus, we try to protect the standard, but if the best system is substantially better then we want to select it.

Nelson and Goldsman (2001) proposed two-stage procedures for this problem that are specifically designed for computer simulation. Similar to Paulson (1952) and Bechhofer and Turnbull (1978), at the end of their procedures the standard is retained if $\overline{X}_0 + c > \overline{X}_i$ for $i = 1, 2, \ldots, k$, otherwise the system with the largest sample mean is selected. The following result provides guidance for designing the algorithm and specifying the value of $c > 0$:

Theorem 6 (Nelson and Goldsman, 2001). *If the distribution of $\overline{X}_{ij} - \mu_i$ is independent of μ_i for $i = 0, 1, 2, \ldots, k$, and if*

$$\begin{aligned}
&\Pr\bigl\{(\overline{X}_i - \overline{X}_0) - (\mu_i - \mu_0) \leqslant c, i = 1, 2, \ldots, k\bigr\} \geqslant 1 - \alpha,\\
&\Pr\bigl\{(\overline{X}_k - \overline{X}_0) - (\mu_k - \mu_0) > c - \delta,\\
&\qquad (\overline{X}_k - \overline{X}_i) - (\mu_k - \mu_i) > -\delta, i = 1, 2, \ldots, k - 1\bigr\} \geqslant 1 - \alpha,
\end{aligned}$$

then (20) *and* (21) *hold.*

The two conditions are intuitive: The first insures that, when the standard is best, no inferior system's sample mean beats it by too much. The second condition guarantees that when system k is best by δ or more, then its sample mean is enough bigger than the standard's sample mean, and is bigger than the sample mean of every other system, so that it is selected.

Kim (2005) proposed fully sequential procedures for comparison with a standard. A procedure such as Paulson (1964) or $\mathcal{KN}$ is not directly applicable because it would require $\mu_0 \geqslant \mu_k + \delta$, not just $\mu_0 \geqslant \mu_k$, for the standard to be retained with the desired probability. But since $\mathcal{KN}$ and other procedures that are similar to Paulson (1964) focus on all pairwise comparisons, and the identity of the standard is known, the following reformulation in Kim (2005) works: In any comparison with the standard, revise (5) from

$$\overline{X}_0(r) \geqslant \max_{i\in I} \overline{X}_i(r) - \max\left\{0, \frac{a}{r} - \lambda\right\}$$

to

$$\overline{X}_0(r) + \frac{\delta}{2} \geqslant \max_{i\in I} \overline{X}_i(r) - \max\left\{0, \frac{a}{r} - \lambda\right\} \tag{22}$$

and further, select a and λ to detect differences of size $\delta/2$ instead of δ.

Why does this work? Suppose that $\mu_0 = \mu_k$ so that the standard should be retained. Then $\overline{X}_0(r) + \delta/2$ has expected value at least $\delta/2$ better than all the other systems and will be retained with the desired probability. On the other hand, if $\mu_k = \mu_0 + \delta$, so that system k should be selected, then $\overline{X}_0(r) + \delta/2$ has expected value that is $\delta/2$ inferior to the best and will be eliminated with the appropriate probability. The procedure is set up for, and detects, differences of size δ for comparisons among the alternatives, but whenever the standard is involved in a comparison, the procedure is adjusted to detect $\delta/2$.

7.2 Selecting the system most likely to be the best

In multinomial selection problems, the definition of "best" is the system that is *mostly likely* to be the best in a single trial. Historically, these procedures were designed for experiments that have a categorical response (e.g., which among 5 soft drinks a subject will say that they prefer). If there are k categories, p_i is the probability that the ith category is selected in a single trial, and the trials are independent, then the number of times each category is selected has a multinomial distribution. More precisely, let N_i be the number of times that category i is chosen in n independent trials. Then

$$\Pr\{N_1 = n_1, N_2 = n_2, \ldots, N_k = n_k\} = \frac{n!}{\prod_{i=1}^{k} n_i!} \prod_{i=1}^{k} p_i^{n_i},$$

where $\sum_{i=1}^{k} n_i = n$.

For convenience of notation (but unknown to us), assume that $p_k \geqslant p_{k-1} \geqslant \cdots \geqslant p_1$. Therefore, a correct selection in this context is selecting category k. Multinomial selection procedures seek to provide a guaranteed PCS. The compromise that makes this possible is to guarantee the PCS whenever $p_k/p_{k-1} \geqslant \theta$, where $\theta > 1$ is interpreted as the smallest p_k/p_{k-1} ratio worth detecting (and therefore defines another form of indifference zone). Bechhofer et al. (1959) proposed a single-stage procedure that satisfies this requirement. Other work on this problem includes Bechhofer and Goldsman (1986), who proposed a procedure that uses closed, sequential sampling. See Bechhofer et al. (1995) for a review.

Goldsman (1984a, 1984b) first suggested the more general use of this type of procedure to find the simulated system most likely to produce the "most desirable" observation on a given trial, where "most desirable" can be almost any criterion of goodness. This often means identifying the system i with the largest value of p_i, where $p_i = \Pr\{X_{ij} > X_{\ell j}, \forall \ell \neq i\}$ for a problem in which a larger simulation output response is better. For instance, in a reliability setting the simulation output X might be the time to system failure and the goal is to select the system that is most likely to survive the longest. The key difference from the categorical data context is that *each trial involves obtaining a response value from each simulated system and the winner is determined by comparing these values*. Stated differently, a trial or replication produces a vector response $(X_{1j}, X_{2j}, \ldots, X_{kj})$ that is transformed into a categorical response $(0, 0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 indicating the system with the largest output on the jth replication.

With this in mind, Miller et al. (1998) devised a single-stage procedure that achieves a higher probability of correct selection than do Bechhofer et al. (1959) in the case where both the replications (vector observations) and the systems themselves are simulated independently (no common random numbers). The key insight is that the formation of vector observations by replication number – $(X_{1j}, X_{2j}, \ldots, X_{kj})$ – is arbitrary; any vector formed with one output from each system has the same distribution. Thus, n replications from each system can form n^k vector observations. Of course, these vectors are no longer independent, since they share observations, so this is not the same as having

n^k independent replications. However, Miller et al. (1998) showed that forming all vector comparisons increases the effective sample size by at least one third, and their procedure exploits this additional information to achieve the desired PCS with fewer total replications.

The role of CRN in multinomial selection is interesting and worthy of discussion. In the means-based procedures that are the focus of this chapter, CRN was introduced as an experiment design technique to increase efficiency but it has no effect on the problem parameters, specifically $\mu_1, \mu_2, \ldots, \mu_k$. However, in multinomial selection the value of $p_i = \Pr\{X_{ij} > X_{\ell j}, \forall \ell \neq i\}$ will, in general, be different if the systems are simulated with CRN as opposed to independently, as noted by Mata (1993). Miller and Bauer (1997) observed that the identity of the best is typically the same with or without CRN, although this is not guaranteed, but the relative dominance of the best can increase or decrease even if the identity is unchanged. Thus, in multinomial selection the decision as to whether or not to use CRN should be based on whether the actual performance of the real systems would be affected by like or common factors, or whether their actual performance would be independent.

7.3 Selecting the largest probability of success

In Bernoulli selection problems, the basic output from each system on each independent replication, denoted X_{ij}, takes either the value 1 (“success”) or 0 (“failure”), and the best system is the one with the largest probability of success, $p_i = \Pr\{X_{ij} = 1\}$. Simulation applications include comparing systems in terms of their ability to survive a mission or to meet a goal such as on-time performance. To our knowledge there has been little research on, or application of, Bernoulli selection in simulation despite the obvious relevance.

Assume that (unknown to us) $p_k \geqslant p_{k-1} \geqslant \cdots \geqslant p_1$ so that a correct selection is choosing system k. At least three types of indifference-zone parameters have been considered in Bernoulli selection.

Difference: $p_k - p_{k-1} \geqslant \delta$.
Odds ratio: $(p_k/(1 - p_k))/(p_{k-1}/(1 - p_{k-1})) \geqslant \theta$.
Relative risk: $p_k/p_{k-1} \geqslant \theta$.

Here $\delta > 0$ and $\theta > 1$ are user-specified parameters. A PCS $\geqslant 1 - \alpha$ is desired in any case. Clearly the Difference formulation is analogous to the IZ formulation for normal-theory procedures described throughout this chapter. A concern about the Difference formulation is that it seems unnatural for a significant difference not to be tied to the sizes of the success probabilities themselves. The other two formulations attempt to incorporate this feature. See Chapter 7 of Bechhofer et al. (1995) for a discussion of this issue and a list of procedures.

To obtain a sense of the analysis involved in developing Bernoulli selection procedures, suppose that the IZ is of the odds-ratio form, there are only $k = 2$ systems and the two systems are simulated independently (no CRN). We want to develop a procedure that terminates when $\sum_j (X_{2j} - X_{1j}) = \pm a$, where a is a nonnegative integer. Thus, the procedure terminates whenever system 2 has a more successes than system 1, or vice versa. In this case a correct selection will occur if $\sum_j (X_{2j} - X_{1j}) = a$.

Let $S_n = \sum_{j=1}^{n} (X_{2j} - X_{1j})$, $n = 0, 1, \ldots$, a random walk on $\{-a, -a+1, \ldots, a-1, a\}$ with initial state $S_0 = 0$. Although we could work with this process, it will be more useful to consider a related process,

$$Y_m = \text{value of } S_n \text{ after its } m\text{th change in state.}$$

Stated differently, $\{Y_m; m = 0, 1, 2, \ldots\}$ is the process that results from ignoring the transitions of $\{S_n\}$ that do not change its state. Assume that

$$\frac{p_k/(1-p_k)}{p_{k-1}/(1-p_{k-1})} = \theta$$

so that the IZ condition is an equality. It is easy to show that $\{Y_m; m = 0, 1, 2, \ldots\}$ is a time-homogeneous discrete-time Markov chain with one-step transition probabilities

$$q_{ij} = \Pr\{Y_{m+1} = j | Y_m = i\} = \begin{cases} 1, & i = j = a, -a, \\ \frac{\theta}{1+\theta}, & j = i+1, i < a, \\ \frac{1}{1+\theta}, & j = i-1, i > -a, \\ 0, & \text{otherwise.} \end{cases}$$

Notice that the IZ assumption leads to transition probabilities that are independent of the actual values of p_1 and p_2.

Using standard Gambler's ruin results (e.g., Ross, 2000), the probability that the process is absorbed in state a – the state that would cause us to declare system 2 as best – is $\theta^a/(1+\theta^a)$. Therefore, to obtain PCS $\geqslant 1-\alpha$ we set

$$a = \left\lfloor \frac{\ln((1-\alpha)/\alpha)}{\ln(\theta)} \right\rfloor.$$

Random-walk analysis is at the heart of many sequential procedures for Bernoulli selection, and Smith (1995) shows that it is often useful for evaluating the efficiency of such procedures.

The role of CRN in Bernoulli selection is largely unexplored. Continuing the previous example, suppose now that the data are generated as follows:

$$X_{ij} = \begin{cases} 0, & U_j \leqslant 1 - p_i, \\ 1, & \text{otherwise,} \end{cases}$$

for $i = 1, 2$, where $U_1, U_2, \ldots$ are i.i.d. U(0, 1) random variables. This set up induces the largest possible correlation between two Bernoulli random variables and has a profound effect on our procedure because now the outcome

$(X_{2j} = 0, X_{1j} = 1)$ *cannot occur*. Therefore, the one-step transition probabilities of $\{Y_m\}$ become

$$q_{ij} = \Pr\{Y_{m+1} = j | Y_m = i\} = \begin{cases} 1, & i = j = a, -a, \\ 1, & j = i + 1, i < a, \\ 0, & \text{otherwise}, \end{cases}$$

and the PCS of the selection procedure is 1. This might seem like a desirable outcome until you consider the efficiency of the procedure. Remember that the cost of running the procedure is not the number of Y_m's that are required, but rather the number of X_{ij}'s. In the independent case, the expected number of X_{ij}'s required for each transition of Y_m is

$$\frac{2}{p_2 - p_1 + 2p_1(1 - p_2)}$$

but under CRN it is greater, specifically,

$$\frac{2}{p_2 - p_1}.$$

In many cases this is enough to make the procedure *less efficient* when CRN is employed (assuming the value of a is not altered). For instance, if $p_2 = 4/5$, $p_1 = 3/4$, $1 - \alpha = 0.95$ and $\theta = 4/3$, then we can show that the expected number of outputs that must be generated under independent sampling is about 357, while under CRN it is 400. Obviously a should be altered when CRN is employed – in fact $a = 1$ is adequate in this illustration – but to date no procedure has been developed. Tamhane (1980, 1985) does provide a procedure for $k = 2$ systems, but it requires being able to provide an upper bound on $\Pr\{X_{1j} \neq X_{2j}\}$.

7.4 Bayesian procedures

Instead of providing a PCS guarantee, Bayesian procedures attempt to allocate a finite computation budget to maximize the posterior PCS of the selected system. Chen et al. (2000) and Chick and Inoue (2001) are two recent references; see Chapter 9 for a thorough review of this approach.

8 Future directions

The following are some directions in which future breakthroughs are most needed:

- Procedures specifically designed for very large numbers of alternatives, particularly when the alternatives are not all available at the same time (such as occurs during the search phase of an optimization-via-simulation algorithm).

- Procedures that exploit common random numbers for very large numbers of alternatives without employing such conservative inequalities that the impact of CRN is overwhelmed.
- Development of constrained selection-of-the-best procedures; for instance, procedures that select the best based on one performance measure, subject to a constraint or condition on a different measure.

Acknowledgements

Portions of this chapter were published previously in the *Proceedings of the 2003 Winter Simulation Conference*. This work was partially supported by NSF Grant Nos. DMI-0217690 and DMI-0400260. The authors gratefully acknowledge the many helpful comments of Ajit Tamhane.

References

Banerjee, S. (1961). On confidence interval for two-means problem based on separate estimates of variances and tabulated values of t-table. *Sankhyā A* 23, 359–378.

Banks, J., Carson, J.S., Nelson, B.L., Nicol, D. (2005). *Discrete-Event System Simulation*, 4th edition. Prentice Hall, Upper Saddle River, NJ.

Bechhofer, R.E. (1954). A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Annals of Mathematical Statistics* 25, 16–39.

Bechhofer, R.E., Goldsman, D. (1986). Truncation of the Bechhofer–Kiefer–Sobel sequential procedure for selecting the multinomial event which has the largest probability, II: Extended tables and an improved procedure. *Communications in Statistics – Simulation and Computation B* 15, 829–851.

Bechhofer, R.E., Turnbull, B.W. (1978). Two $(k + 1)$-decision selection procedures for comparing k normal means with a specified standard. *Journal of the American Statistical Association* 73, 385–392.

Bechhofer, R.E., Elmaghraby, S., Morse, N. (1959). A single-sample multiple-decision procedure for selecting the multinomial event which has the highest probability. *Annals of Mathematical Statistics* 30, 102–119.

Bechhofer, R.E., Santner, T.J., Goldsman, D. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. Wiley, New York.

Boesel, J., Nelson, B.L., Kim, S.-H. (2003). Using ranking and selection to "clean up" after simulation optimization. *Operations Research* 51, 814–825.

Chen, H.C., Chen, C.H., Yücesan, E. (2000). Computing efforts allocation for ordinal optimization and discrete event simulation. *IEEE Transactions on Automatic Control* 45, 960–964.

Chick, S., Inoue, K. (2001). New two-stage and sequential procedures for selecting the best simulated system. *Operations Research* 49, 1609–1624.

Dalal, S.R., Hall, W.J. (1977). Most economical robust selection procedures for location parameters. *The Annals of Statistics* 7, 1321–1328.

Damerdji, H., Nakayama, M.K. (1999). Two-stage multiple-comparison procedures for steady-state simulation. *ACM TOMACS* 9, 1–30.

Damerdji, H., Glynn, P.W., Nakayama, M.K., Wilson, J.R. (1996). Selecting the best system in transient simulations with variances known. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 281–286.

Dudewicz, E.J. (1969). An approximation to the sample size in selection problems. *Annals of Mathematical Statistics* 40, 492–497.

Dudewicz, E.J. (1995). The heteroscedastic method: Fifty+ years of progress 1945–2000, and professor Minoru Siotani's award-winning contributions. *American Journal of Mathematical and Management Sciences* 15, 179–197.

Fabian, V. (1974). Note on Anderson's sequential procedures with triangular boundary. *The Annals of Statistics* 2, 170–176.

Glynn, P.W., Iglehart, D.L. (1990). Simulation output analysis using standardized time series. *Mathematics of Operations Research* 15, 1–16.

Goldsman, D. (1984a). A multinomial ranking and selection procedure: Simulation and applications. In: Shepard, S., Pooch, U.W., Pegden, C.D. (Eds.), *Proceedings of the 1984 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 259–264.

Goldsman, D. (1984b). On selecting the best of K systems: An expository survey of indifference-zone multinomial procedures. In: Shepard, S., Pooch, U.W., Pegden, C.D. (Eds.), *Proceedings of the 1984 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 107–112.

Goldsman, D., Nelson, B.L. (1998). Comparing systems via simulation. In: Banks, J. (Ed.), *Handbook of Simulation*. Wiley, New York, pp. 273–306.

Goldsman, D., Kim, S.-H., Marshall, W., Nelson, B.L. (2001). Ranking and selection procedures for steady-state simulation: Perspectives and procedures. *INFORMS Journal on Computing* 14, 2–19.

Gupta, S.S. (1956). On a decision rule for a problem in ranking means. Doctoral dissertation, Institute of Statistics, University of North Carolina, Chapel Hill, NC.

Gupta, S.S. (1965). On some multiple decision (ranking and selection) rules. *Technometrics* 7, 225–245.

Hartmann, M. (1988). An improvement on Paulson's sequential ranking procedure. *Sequential Analysis* 7, 363–372.

Hartmann, M. (1991). An improvement on Paulson's procedure for selecting the population with the largest mean from k normal populations with a common unknown variance. *Sequential Analysis* 10, 1–16.

Hochberg, Y., Tamhane, A.C. (1987). *Multiple Comparison Procedures*. Wiley, New York.

Hong, L.J., Nelson, B.L. (2005). The tradeoff between sampling and switching: New sequential procedures for indifference-zone selection. *IIE Transactions* 37, 623–634.

Hsu, J.C. (1996). *Multiple Comparisons: Theory and Methods*. Chapman and Hall, New York.

Jennison, C., Turnbull, B.W. (2000). *Group Sequential Methods with Applications to Clinical Trials*. Chapman and Hall, New York.

Jennison, C., Johnstone, I.M., Turnbull, B.W. (1980). Asymptotically optimal procedures for sequential adaptive selection of the best of several normal means. Technical report, Department of ORIE, Cornell University, Ithaca, NY.

Jennison, C., Johnstone, I.M., Turnbull, B.W. (1982). Asymptotically optimal procedures for sequential adaptive selection of the best of several normal means. In: Gupta, S.S., Berger, J.O. (Eds.), *Statistical Decision Theory and Related Topics III*, vol. 2. Academic Press, New York, pp. 55–86.

Koenig, L.W., Law, A.M. (1985). A procedure for selecting a subset of size m containing the ℓ best of k independent normal populations, with applications to simulation. *Communications in Statistics – Simulation and Computation B* 14, 719–734.

Kim, S.-H. (2005). Comparison with a standard via fully sequential procedures. *ACM TOMACS* 15, 155–174.

Kim, S.-H., Nelson, B.L. (2001). A fully sequential procedure for indifference-zone selection in simulation. *ACM TOMACS* 11, 251–273.

Kim, S.-H., Nelson, B.L. (2006). On the asymptotic validity of fully sequential selection procedures for steady-state simulation, *Operations Research*, in press.

Kimball, A.W. (1951). On dependent tests of significance in the analysis of variance. *Annals of Mathematical Statistics* 22, 600–602.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Lehmann, E.L. (1999). *Elements of Large-Sample Theory*. Springer-Verlag, New York.

Mata, F. (1993). Common random numbers and multinomial selection. *Computers and Industrial Engineering* 25, 33–36.

Miller, J.O., Bauer, K.W. (1997). How common random numbers affect multinomial selection. In: Andradóttir, S., Healy, K.J., Withers, D.H., Nelson, B.L. (Eds.), *Proceedings of the 1997 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 342–347.

Miller, J.O., Nelson, B.L., Reilly, C.H. (1998). Efficient multinomial selection in simulation. *Naval Research Logistics* 45, 459–482.

Mukhopadhyay, N., Solanky, T.K.S. (1994). *Multistage Selection and Ranking Procedures: Second-Order Asymptotics*. Dekker, New York.

Nelson, B.L., Goldsman, D. (2001). Comparisons with a standard in simulation experiments. *Management Science* 47, 449–463.

Nelson, B.L., Matejcik, F.J. (1995). Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Science* 41, 1935–1945.

Nelson, B.L., Swann, J., Goldsman, D., Song, W.-M.T. (2001). Simple procedures for selecting the best system when the number of alternatives is large. *Operations Research* 49, 950–963.

Paulson, E. (1952). On the comparison of several experimental categories with a control. *Annals of Mathematical Statistics* 23, 239–246.

Paulson, E. (1964). A sequential procedure for selecting the population with the largest mean from k normal populations. *Annals of Mathematical Statistics* 35, 174–180.

Perng, S.K. (1969). A comparison of the asymptotic expected sample sizes of two sequential procedures for ranking problem. *Annals of Mathematical Statistics* 40, 2198–2202.

Pichitlamken, J., Nelson, B.L. (2001). Selection-of-the-best procedures for optimization via simulation. In: *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 401–407.

Rinott, Y. (1978). On two-stage selection procedures and related probability-inequalities. *Communications in Statistics – Theory and Methods A* 7, 799–811.

Robbins, H., Sobel, M., Starr, N. (1968). A sequential procedure for selecting the largest of k means. *Annals of Mathematical Statistics* 39, 88–92.

Ross, S.M. (2000). *Introduction to Probability Models*, 7th edition. Academic Press, New York.

Siegmund, D. (1985). *Sequential Analysis: Tests and Confidence Intervals*. Springer-Verlag, New York.

Slepian, D. (1962). The one-sided barrier problem for Gaussian noise. *Bell Systems Technical Journal* 41, 463–501.

Smith, M.J. (1995). Ranking and selection: Open sequential procedures for Bernoulli populations. Unpublished MS thesis, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Stein, C. (1945). A two-sample test for a linear hypothesis whose power is independent of the variance. *Annals of Mathematical Statistics* 16, 243–258.

Sullivan, D.W., Wilson, J.R. (1989). Restricted subset selection for simulation. *Operations Research* 37, 52–71.

Tamhane, A.C. (1980). Selecting the better Bernoulli treatment using a matched samples design. *Journal of the Royal Statistical Society B* 42, 26–30.

Tamhane, A.C. (1985). Some sequential procedures for selecting the better Bernoulli treatment using a matched samples design. *Journal of the American Statistical Association* 80, 455–460.

Wald, A. (1947). *Sequential Analysis*. Wiley, New York.

Welch, B.L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika* 25, 350–362.

Wilcox, R.R. (1984). A table for Rinott's selection procedure. *Journal of Quality Technology* 16, 97–100.

Wilson, J.R. (2001). A multiplicative decomposition property of the screening-and-selection procedures of Nelson et al. *Operations Research* 49, 964–966.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13018-2

Chapter 18

Metamodel-Based Simulation Optimization

Russell R. Barton

The Pennsylvania State University, USA
E-mail: rbarton@psu.edu

Martin Meckesheimer

Boeing Company, USA
E-mail: martin.meckesheimer@boeing.com

Abstract

Simulation models allow the user to understand system performance and assist in behavior prediction, to support system diagnostics and design. Iterative optimization methods are often used in conjunction with engineering simulation models to search for designs with desired properties. These optimization methods can be difficult to employ with a discrete-event simulation, due to the stochastic nature of the response(s) and the potentially extensive run times. A metamodel, or model of the simulation model, simplifies the simulation optimization in two ways: the metamodel response is deterministic rather than stochastic, and the run times are generally much shorter than the original simulation. Metamodels based on first- or second-order polynomials generally provide good fit only locally, and so a series of metamodels are fit as the optimization progresses. Other classes of metamodels can provide good global fit; in these cases one can fit a (global) metamodel once, at the start of the optimization, and use it to find a design that will meet the optimality criteria. Both approaches are discussed in this chapter and illustrated with an example.

1 Introduction

Simulation models provide insight on the behavior of real systems. This insight can be used to improve system performance by ad hoc changes to the system design parameter values, or the simulation model may be analyzed repeatedly to find a set of design parameters that provide the best simulated performance. We define simulation optimization as the latter case: repeated analysis of the simulation model with different values of design parameters, in an attempt to identify best simulated system performance. The design parameters of the real system are set to the 'optimal' parameter values determined by the simulation optimization exercise, rather than in an ad hoc manner based on qualitative insights gained from exercising the simulation model. We will use the following notation to represent the general simulation optimization problem, following Andradóttir (1998):

$$\min_{\theta \in \Theta} f(\theta), \tag{1}$$

where θ is the (possibly vector-valued) design parameter of the system being simulated, and the feasible region $\Theta \in \mathbb{R}^d$ is the set of possible values of θ. The optimization model response function is represented by $f(\theta)$ which is usually the expected value (long-term average) of some simulated system performance measure Y as a function of the design parameter vector θ. That is,

$$f(\theta) = \mathrm{E}\big(Y(\theta)\big).$$

The form of f is not known. Its value is estimated using n runs of the simulation model under the design scenario specified by θ,

$$\widehat{f(\theta)} = \sum_{i=1}^{n} Y_i \Big/ n, \tag{2}$$

where the dependence of Y on the value of θ has been suppressed. While $f(\theta)$ is deterministic, its estimate is stochastic, since the simulation run time must be finite (so $n < \infty$).

Simulation optimization strategies depend on the nature of f and Θ. When the feasible set of design parameter vector values Θ is a discrete set, appropriate optimization methods include ranking and selection (Chapter 17), random search (Chapter 20) and metaheuristics (Chapter 21). If Θ is continuous and f is differentiable, then gradient-based methods (Chapter 19) or metamodel-based optimization (this chapter) can be used. This structure is shown in Figure 1.

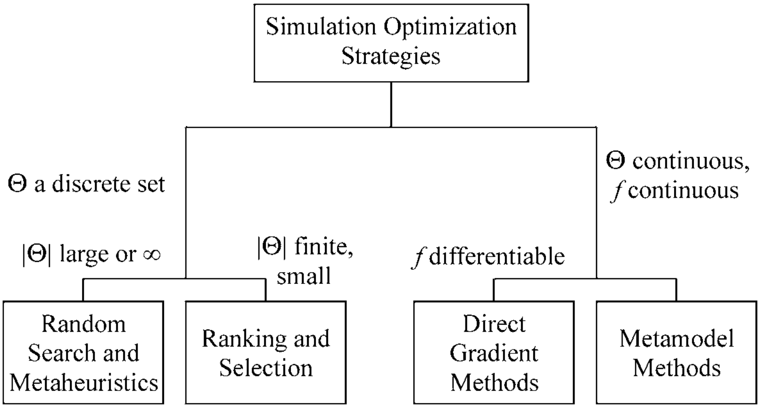


Fig. 1. Simulation optimization strategy depends on the nature of Θ and f.

Table 1.
Simulation optimization strategies (+, advantage; −, disadvantage)

Strategy	Coding modifications to simulation	External 'custom' (non-RSM) metamodel code	External standard statistical (RSM) code	External optimizer code	Efficiency	Provides insight on local response behavior	Provides insight on global response behavior
Random search and metaheuristics				−			
Ranking and selection				−			
Direct gradient methods	−			−	+	+	
Response surface methodology			−			+	
Global metamodel optimization		−		−		+	+

The properties of these general classes of optimization strategies are shown in Table 1. Random search and metaheuristics attempt to select θ values from Θ efficiently. This generally requires a specialized code external to the simulation package. Ranking and selection methods assume that a set of θ alternatives is given, and determine the number of simulation replications and run lengths required for each alternative to give a pre-specified probability of selecting the best or near-best θ. The effectiveness of ranking and selection in an optimization setting depends on the method for choosing candidates. Again, external calculations are necessary to compute run lengths and replications.

If Θ is continuous, other search methods can be employed. Stochastic gradient-based optimization methods such as stochastic approximation can use efficient methods for estimating the gradient of f such as likelihood ratios (these require modification of the simulation code – see Table 1), or less efficient finite-difference approximations. These methods search Θ to find optimum system performance without attempting to provide a global approximation to f, an efficiency advantage (see Section 5.4 in Fu, 1994). The stochastic optimization code is usually external to the simulation code, and can be complex to implement. Simplex search combines features of ranking and selection and gradient optimization, but can fail on stochastic responses with large variation (Tomick et al., 1995). Metamodel-based optimization methods fall in two categories, both of which use an indirect-gradient optimization strategy. It is indirect because the gradient is computed for the metamodel (a deterministic function) rather than for the simulation response. Response surface methodology (RSM) is a metamodel-based optimization method that builds linear or quadratic local approximations to f to be used by a deterministic gradient-based optimization strategy. Old local models are discarded and new ones are fitted at the end of each line search cycle. The global metamodel-based optimization methods build a single global metamodel (usually requiring a much larger set of simulation runs) which is then optimized using a gradient-based strategy.

In some cases metamodel-based optimization can be used with discrete-valued design parameters. If the parameter has a natural integer order, say the number of machines in a work cell, then the continuous approximation can be solved using a metamodel strategy, and the solution parameter value rounded to the nearest (feasible) integer value.

This brief description of simulation optimization serves to place metamodel-based optimization in the overall context of simulation optimization. Fu (2001) provides an overview of simulation optimization methods and the implementations that exist in commercial software.

The remainder of this chapter focuses on metamodel-based optimization strategies. The next section provides an overview of metamodel types and their appropriateness for discrete-event simulation response functions. The following section describes the overall strategy of metamodel-based optimization, and highlights the differences between local and global metamodel approaches. Section 4 describes the RSM approach in more detail and highlights important issues in its use. RSM is illustrated using a network routing design example. Section 5 examines one global metamodel-based approach and applies it to the same optimization case. Section 6 provides a summary and describes how metamodels can be used for robust design, whose objective is to simultaneously optimize the expected value and the standard deviation of the response. Any discussion of metamodel-based optimization necessitates reference to concepts from a variety of fields, including simulation, statistics, response surfaces, and nonlinear optimization. For an introduction to concepts and terminology in these areas, see Banks et al. (2005) and Law and Kelton (2000) for simulation, Box and Draper (1987), Khuri and Cornell (1987), Montgomery (2001), Myers and Montgomery (2002) and Santner et al. (2003) for statistics and response surfaces, and Bertsekas (1999) for nonlinear optimization.

2 Metamodels and simulation

Experimentation with computer simulation models of proposed or existing real systems is often used to make decisions on changes to the system design. Analysts exercise the simulation model because cost, time or other constraints prohibit experimentation with the real system. For the extensive experimentation required for optimization, the simulation models themselves may require excessive computation, and so simpler approximations are often constructed; models of the model, called metamodels (Kleijnen, 1975a, 1975b, 1987) or surrogate models (Yesilyurt and Patera, 1995). These metamodels are usually deterministic approximating functions for f that are inexpensive to compute. Running multiple replications of the simulation to produce $\widehat{f(\theta)}$ is expensive;

running the metamodel once produces the deterministic value $g(\theta)$ which approximates $f(\theta)$ with low computational expense.

The major issues in metamodeling include: (i) the choice of a functional form for g, (ii) the design of experiments, that is, the selection of a set of θ values at which to observe $Y(\theta)$ by running the simulation model, the assignment of random number streams, the length of runs, etc., (iii) fitting g to the simulation response using the experimental data, and (iv) the assessment of the adequacy of the fitted metamodel (confidence intervals, hypothesis tests, lack of fit and other model diagnostics). We will restrict this discussion to the case of a single output performance measure, say total cost. Multiple output measures would each require a separate metamodel.

The functional form for g is typically a linear combination of basis functions from a parametric family. There are choices for parametric families (polynomials, sine functions, piecewise polynomials, etc.) and choices for fitting; that is, choosing the 'best' representation from within a family (via least squares, maximum likelihood, cross-validation, etc.). This section draws from earlier metamodel review papers (Barton, 1992, 1998), with a focus on the most promising metamodel and experiment design strategies for simulation optimization.

2.1 Response surface metamodels

Response surface models were developed over fifty years ago for "the exploration and exploitation" of stochastic response functions (Box and Wilson, 1951; Box, 1954). They are used in conjunction with response surface methodology, the most commonly used approach to metamodel-based simulation optimization. This metamodel family consists of first or second-order polynomial probability models fitted to observed values of Y, the system response. A full second-order response surface model would be

$$Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k + \varepsilon, \quad \varepsilon \sim \mathrm{NID}\bigl(0, \sigma^2\bigr), \tag{3}$$

where NID indicates that the deviations have independent (and identical) normal distributions.

Suppose that an experiment has been conducted, with simulation runs at parameter settings $\theta^1, \theta^2, \ldots, \theta^n$ and corresponding observed responses (perhaps averages of replications) of $y^1, \ldots, y^n$. Let y represent the vector of responses. For metamodel prediction, maximum likelihood (equivalently, least squares) estimators for the β_0, β_i, β_{ik} and σ^2 are computed, and used in the prediction equation

$$g(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k. \tag{4}$$

Response surface metamodels can be fit using standard statistical packages. Experiment designs for RSM models and other details of this method are discussed in Section 4.

2.2 Regression spline metamodels

If a linear or quadratic polynomial regression does not provide a good fit, it is natural to think of higher-order polynomial approximations. Any polynomial regression model can be constructed from linear combinations of the functions $\prod_k \theta_{j_k}$, where the index j_k may take the same value more than once. This choice for a basis has drawbacks. The high-order polynomial achieves a good fit by adjusting coefficients to achieve cancellation of large oscillations over most of the range. This reliance on cancellation makes high-order polynomial fits nonrobust. Figure 2 shows a 14th degree polynomial fitted to a sample of deterministic responses at 15 points evenly spaced in the interval $[-5, 5]$ for the response function $f(\theta) = 1/(1 + \theta^2)$. The magnitude of the overshoot at the extremes in the figure would increase if the number of design points and the degree of the polynomial were further increased.

If a linear, quadratic, or cubic approximation to the function is adequate, then polynomial basis functions can be used to construct an effective metamodel. If this is not adequate, the simulation modeler should consider other basis functions from which to build the metamodel. The difficulties with polynomial basis functions are avoided if: (i) they are applied to a small region, and (ii) only low order polynomials are used. This is the motivation for metamodels based on piecewise polynomial basis functions. When continuity restrictions are applied to adjacent pieces, the piecewise polynomials are called splines. The (univariate) metamodel can be written as

$$g(\theta) = \sum_k \beta_k B_k, \tag{5}$$

where the B_k are quadratic or cubic piecewise polynomial basis functions. The basis functions can be described most simply for the univariate case. The domain is divided into intervals $[t_1, t_2), [t_2, t_3), \ldots, [t_{m-1}, t_m)$ whose endpoints

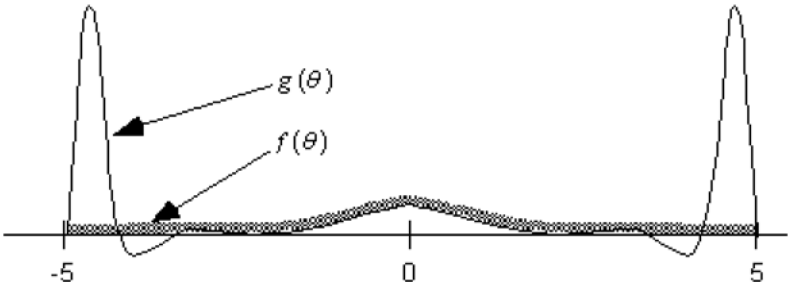


Fig. 2. 14th degree regression polynomial $g(\theta)$ fitted to data from $f(\theta) = 1/(1 + \theta^2)$.

are called knots. Two sets of spline basis functions are commonly used, the truncated power function basis and the B-spline basis (de Boor, 1978). Since most simulation model output functions will not be deterministic, interpolating splines will not be satisfactory. The motivation for smoothing splines is based on an explicit tradeoff between the fit/accuracy of the approximation at known points and smoothness of the resulting metamodel. The fit term is represented as a sum of squared differences of the metamodel and simulation model responses at each of the experimental runs. The smoothness is represented by an integral of the square of some derivative over the region of validity of the metamodel. The relative weight of these objectives is captured by the smoothing parameter, λ: $\lambda = 0$ provides interpolation with no constraint on smoothness. The function that minimizes this quantity will be a spline of order q, which is in C^{q-2} (continuous derivatives up to the $(q-2)$nd derivative) and is a piecewise polynomial with terms up to θ^{q-1}. The knots will occur at points in Θ corresponding to the observed data, θ^i.

An important issue is the selection of the value for the smoothing parameter λ. The value may be chosen by visual examination of the fit, or by minimizing cross-validation (like residual sum of squares), or generalized cross-validation (GCV) (an adjusted residual sum of squares). Eubank (1988) and Craven and Wahba (1979) discuss these approaches.

Smoothing splines are appropriate for simulation metamodels, but the developments have focused on univariate and bivariate functions. The global metamodel example in Section 5 uses the bivariate smoothing spline code of Dierckx (1981, 1993). Unfortunately, the most popular and effective multivariate spline methods are based on interpolating splines, which have little applicability for simulation optimization (Breiman, 1991; Friedman, 1991).

2.3 Spatial correlation (kriging) metamodels

Sacks et al. (1989) and numerous references therein develop a spatial correlation parametric regression modeling approach. The expected smoothness of the function is captured by a spatial correlation function. Spatial correlation models, also called kriging models, have recently become popular for deterministic simulation metamodels (Simpson et al., 1998; Booker et al., 1999). They are more flexible than polynomial models in fitting arbitrary smooth response functions, and seem to be less sensitive than radial basis functions to small changes in the experiment design (Meckesheimer et al., 2002).

Mitchell and Morris (1992) describe the spatial correlation model that is appropriate for (stochastic) simulation responses. The probability model represents the simulation response, Y as

$$Y(\theta) = Z(\theta) + \varepsilon, \tag{6}$$

where ε are independent Gaussian random quantities with mean zero and variance α^2 and Z represents a Gaussian process with mean $\mu(\theta) = \mathrm{E}(Y(\theta))$ and variance σ^2 that exhibits *spatial correlation*,

$$\mathrm{Cov}\big(Z(u), Z(v)\big) = \sigma^2 \mathrm{R}(u, v),$$

where R describes the spatial correlation function. Mitchell and Morris (1992) list four spatial correlation functions, the most commonly used being

$$\mathrm{R}(u, v) = \prod_{j=1}^{p} \exp\big(-\omega_j |u_j - v_j|^2\big), \tag{7}$$

where the index j runs over the dimension of Θ. This Gaussian correlation structure gives an infinitely differentiable metamodel.

Suppose that an experiment has been conducted, with simulation runs at parameter settings $\theta^1, \theta^2, \ldots, \theta^n$ and corresponding observed responses (perhaps averages of replications) of $y^1, \ldots, y^n$. Let y represent the vector of responses. For metamodel prediction, maximum likelihood estimators for the ω_j, μ, σ^2 and α^2 are computed, and used in the prediction equation for $\mathrm{E}(Y)$

$$g(\theta) = \mu + r'(\theta) C^{-1} (y - \mu \mathbf{1}), \tag{8}$$

where $r'(\theta)$ has components $\sigma^2 \mathrm{R}(\theta, \theta^i)$, $C_{jk} = \sigma^2 \mathrm{R}(\theta^i, \theta^k) + \alpha^2 I(i = k)$, and I is the indicator function. The matrix C depends on θ^i but not on θ. Using γ_i to represent the elements of the matrix–vector product $C^{-1}(y - \mu\mathbf{1})$ makes the form of the basis functions of θ for the spatial correlation metamodel clearer (with μ, γ_i and ω_j as the fitted coefficients).

$$g(\theta) = \mu + \sum_{i=1}^{n} \gamma_i \prod_{j=1}^{p} \exp\big(-\omega_j |\theta_j - \theta_j^i|^2\big).$$

For fitting deterministic simulation models, the spatial correlation model excludes the ε term, and the resulting approximation provides an interpolating fit to the experimental data.

Although discussed in Barton (1992), Mitchell and Morris (1992) and Barton (1998), spatial correlation models have not been applied in the discrete-event simulation context until recently. See Kleijnen (2005) for an overview and Kleijnen and van Beers (2005) for an assessment of robustness of this approach in the presence of heterogeneous variance that often characterizes simulation response functions. The book by Santner et al. (2003) gives a good review of spatial correlation models, and provides the PeRK code for fitting and prediction.

Factorial experiment designs can cause ill-conditioned likelihood functions for spatial correlation metamodels. Orthogonal array, Latin hypercube and orthogonal array-based Latin hypercubes have been shown to be effective (Jin et al., 2000; Meckesheimer et al., 2002). A set of C routines written by Art Owen at Stanford University generate orthogonal arrays, Latin hypercube designs, and orthogonal array based Latin hypercube designs. The routines are available from Statlib at *http://lib.stat.cmu.edu*.

2.4 Radial basis function metamodels

Radial basis functions (RBF) provide an alternative approach to multivariate metamodeling. In an empirical comparison, Franke (1982) found radial basis functions to be superior to thin plate splines, cubic splines, B-splines, and several others for fitting deterministic response functions. Tu and Barton (1997) found them to provide effective metamodels for electronic circuit simulation models, and Shin et al. (2002) applied a radial basis function metamodel to a queueing simulation and cited its potential.

The radial basis function approximation consists of a sum of radially symmetric functions centered at different points in the domain Θ. The original development by Hardy (1971) introduced simple "multiquadric" basis functions $\|\theta - c^k\|$ (where $\|\cdot\|$ denotes Euclidean distance) to give the metamodel form

$$g(\theta) = \sum_{k=1}^{r} \gamma_k \|\theta - c^k\|. \tag{9}$$

The parameters to be chosen are the basis function centers c^k, and the coefficients (positive or negative) γ_k, $k = 1, \ldots, r$. If the basis function centers are chosen to be the experiment design points ($r = n$ and $c^k = \theta^k$), then the approximation provides an interpolating fit. Shin et al. (2002) used a Gaussian basis function, $\exp(-\|\theta - c^k\|^2/2\sigma^2)$. Fitting in the noninterpolating case is by least squares.

Radial basis functions can be used with many kinds of experiment designs. Because of the radial symmetry of the functions, the responses are sensitive to scaling of the design points and the axes. This problem is avoided by scaling variables to $+/-1$ and using the same number of levels for each design variable. In a computational study on deterministic response functions, factorial designs generally provided better fit compared with Latin hypercube designs, except, in some instances, near the center of the design space (Hussain et al., 2002). Radial basis function metamodels are easy to code due to the simple form of (9). Example code is provided by (Watlington, 2005).

2.5 Neural network metamodels

Artificial neural networks (ANNs) can approximate arbitrary smooth functions and can be fitted using noisy response values. ANNs were developed to mimic neural processing, and can be implemented on a digital computer or in parallel using networks of numerical processors, whose inputs and outputs are linked according to specific topologies. For an introduction to neural networks, see Másson and Wang (1990).

ANNs used for function approximation are typically multi-layer feedforward networks. Feedforward layered networks have the flexibility to approximate smooth functions arbitrarily well, provided sufficient nodes and layers. This follows from the work of Kolmogorov (1961) whose results imply that any continuous function f can be reproduced over a compact subset by a three-layer feedforward neural network. While there are some approximation schemes using three layers, most approximations use a two layer network structure, with a single hidden layer and a single output node for models having a univariate dependent variable. The overall metamodel is then a linear combination of linear or nonlinear functions of the argument vector θ. Figure 3 shows the structure for a two-layer feedforward network. The function t is a monotone threshold function, $t(u) \to 0$, $u \to -\infty$ and $t(u) \to 1$, $u \to \infty$. The symbol denotes the element by element weighting of elements of the parameter vector θ by weight coefficients ω. This can be a simple dot product, e.g., $\omega_1 \otimes \theta = \sum_i \omega_{1,i}\theta_i$. The transition value for u is the threshold δ_k.

Commonly used threshold functions include the sigmoid functions: $t(u) = 1/2 + \arctan(u)/\pi$, $t(u) = 1/(1 + \exp(-u))$, $t(u) = 1/2 + \tanh(u)/2$.

While t functions are usually threshold functions, it is useful to imagine more general functions, and to think of neural networks as a technique for computing metamodel coefficients and predicted values for a broad class of metamodels, rather than as representing a particular class of modeling techniques. For example, if the t functions are products of power functions of the θ's, then the model will be a polynomial regression, with λ_k values corresponding to the usual β coefficients.

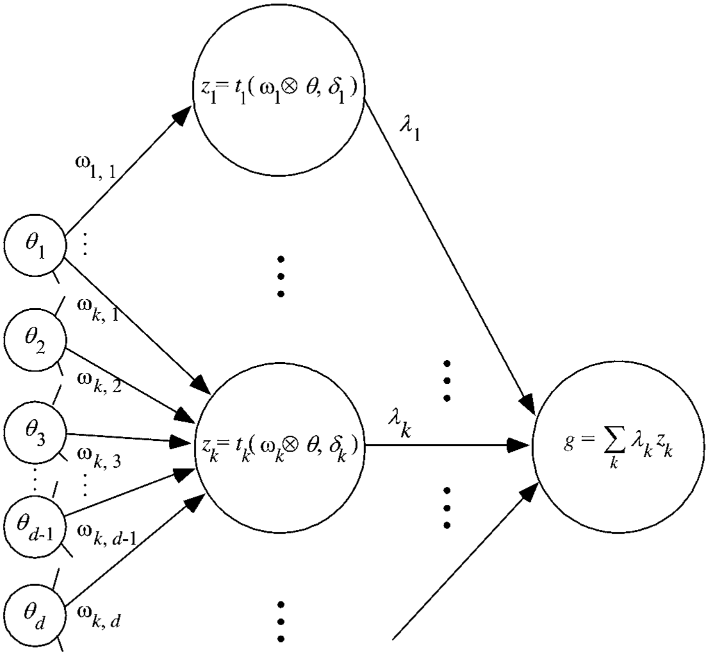


Fig. 3. General structure for two-layer feedforward neural network.

There are many sources for neural network code. See, for example, MATLAB (2005) and Netlab (2005).

2.6 Validating and assessing metamodel fits

Response surface models are validated by a statistical test for lack of fit. This test requires repeated simulation runs under the same parameter setting, e.g., $\theta^i = \theta^{i+1} = \cdots = \theta^{i+r-1}$, where r is the number of replications. This permits the construction of a pure error mean square term, which can be compared with the lack of fit mean square using an F test.

For other metamodel types, fit is usually determined by cross-validation (Stone, 1974). Let $\{y, \theta\}$ represent the experimental results used for fitting, with fitted metamodel function $g(\theta)$ and suppose that $\{y_{-k}, \theta_{-k}\}$ composes the set of experimental results less the kth design point (all replicates) and $g_{-k}(\theta^k)$ is the cross-validation predictive value from the model fitted to $\{y_{-k}, \theta_{-k}\}$. The difference $g_{-k}(\theta^k) - g(\theta)$ can be computed for each design point k, and the sum of squares used as an error measure. Meckesheimer et al. (2002) explored how well the cross-validation error measure approximates the mean squared error of the metamodel fit. The study was restricted to deterministic simulation output using designs with no replications. For radial basis functions the approximation was reasonably good, but for the spatial correlation model, a 'leave-r–out' cross-validation produced a better approximation, where $r = \sqrt{n}$. For additional details on metamodel fitting and validation issues, see Kleijnen and Sargent (2000).

3 Metamodel-based optimization

Law and Kelton (2000) provide a table showing simulation optimization technologies included with commercial simulation software. The list (repeated with web sites in Fu, 2001) shows heuristic search methods including genetic algorithms, tabu search and simulated annealing, but does not include any optimizers solely based on metamodels. Law and McComas (2002) show a similar list in their empirical study. Neural network global metamodels are used in several of these optimizers to screen unpromising new points generated by the heuristic search or to suggest new points to evaluate via simulation (April et al., 2003). For example, see the description of the OptQuest algorithm in Glover et al. (1996).

While metamodels play a role in some of these optimizers, none optimize the metamodel function directly, in a way that might be considered metamodel-based optimization. Further, neither the textbook of Law and Kelton (2000) nor that of Banks et al. (2005) discuss metamodel-based optimization. Fu (2003) states "it is a little baffling that sequential RSM using regression – very well established in the literature *and* quite general and easy to implement – has not been incorporated into any of the commercial packages".

Why then should one be interested in this approach? Metamodel-based optimization simplifies dealing with issues that complicate direct optimization of the simulation model, such as multiple local optima, multiple objectives, and constraints on design parameters and/or responses. This is because the implicitly represented stochastic response of the simulation is replaced by an explicit deterministic metamodel response function. Techniques developed for deterministic optimization can be applied to these metamodel objectives. For example, see Boender and Rinooy Kan (1987), Floudas and Pardalos (1996) and Grossman (1996) for multiple local optima, Charnes and Cooper (1977), Zionts (1992) and Thurston et al. (1994) for handling multiple objectives, and Bazaraa et al. (1993) for handling constraints. Response surface models have substantial statistical theory behind them that permit assessment of the uncertainty about the exact value of the optimal design parameter values and the optimal response (see Myers and Montgomery, 2002; del Castillo and Cahya, 2001; Peterson et al., 2002). Further, the metamodels used during the optimization phase have other usefulness: they can provide insight on the behavior of the simulated system, sensitivity analysis, and the ability to do repeated "what if" analyses quickly. Rapid reporting of the response impacts the efficiency, effectiveness and satisfaction of human interactive design using repeated what if analyses (Simpson et al., 2003).

A significant advantage of a metamodel-based optimization strategy is the incorporation of knowledge of the smoothly varying response function. The metamodel enables a reduction in prediction variance by extending the effect of the law of large numbers over all points in the fitting design. That is, the prediction variance at a design point is less than that which results from an estimate based solely on the replicated simulation runs made at that point. This advantage comes at a cost: bias that is introduced when the metamodel fails to capture the true nature of the response surface (see Figure 2).

A metamodel-based optimization strategy consists of choosing a metamodel form, designing an experiment to fit the metamodel, fitting the metamodel and validating the quality of its fit, optimizing the metamodel (or using it to provide a search direction), and checking the performance of the simulation at the metamodel-predicted optimum (or in the metamodel-determined search direction). In some cases this process is repeated, with the new experiment design focused on the neighborhood of the predicted optimum. Two general strategies have been used for metamodel-based simulation optimization: global metamodel fit, followed by optimization, and iterated local metamodels. These strategies are illustrated graphically in Figure 4. A third strategy has been used with deterministic simulations: global metamodel fit with local updates (Alexandrov et al., 1998).

The iterative local metamodeling strategy is commonly used to determine an optimization search direction. This is followed by a line search using the simulation response directly. Because the metamodels are local, Taylor's theo-

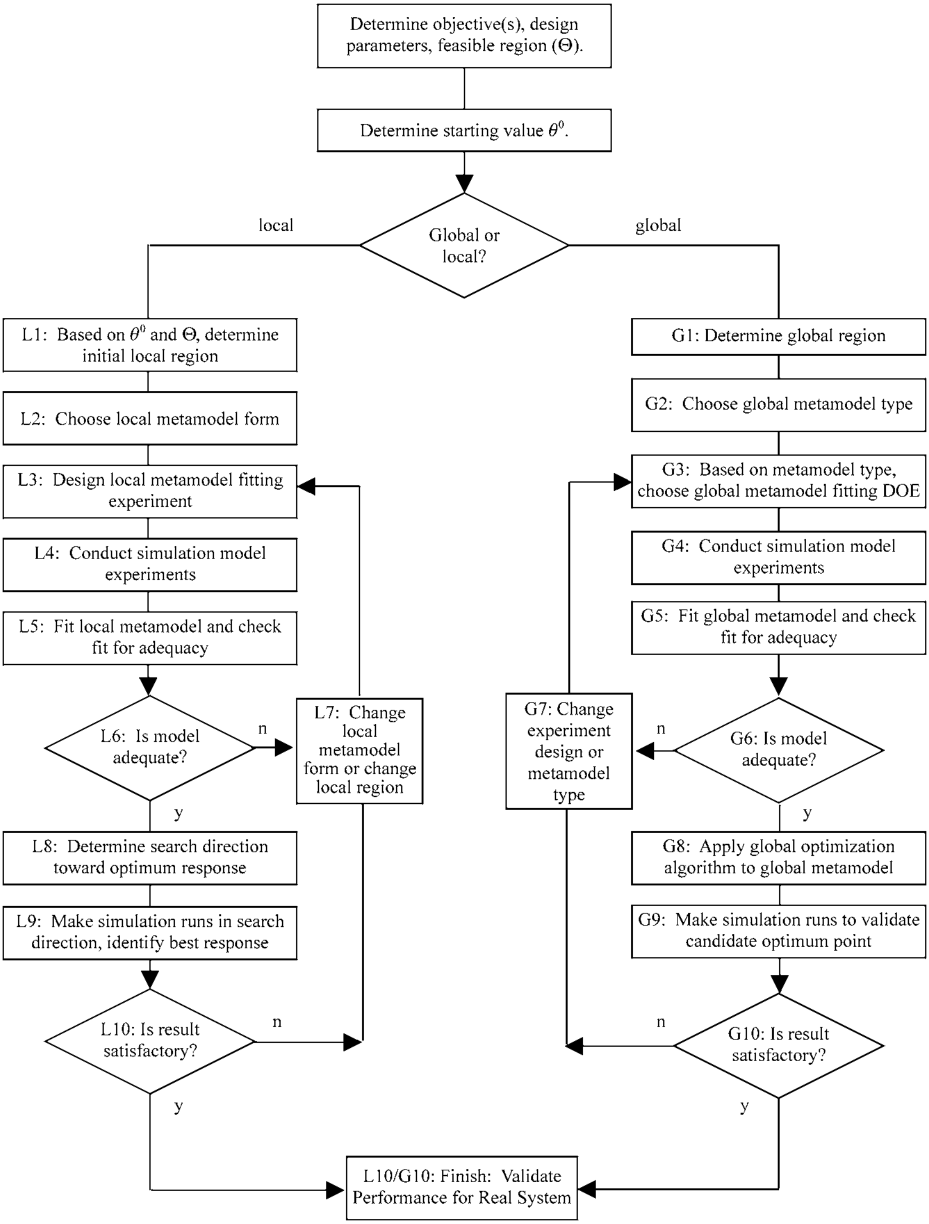

Determine objective(s), design parameters, feasible region (Θ).
Determine starting value θ^0.
Global or local?
local
global
L1: Based on θ^0 and Θ, determine initial local region
L2: Choose local metamodel form
L3: Design local metamodel fitting experiment
L4: Conduct simulation model experiments
L5: Fit local metamodel and check fit for adequacy
L6: Is model adequate?
n
y
L7: Change local metamodel form or change local region
L8: Determine search direction toward optimum response
L9: Make simulation runs in search direction, identify best response
L10: Is result satisfactory?
n
y
G1: Determine global region
G2: Choose global metamodel type
G3: Based on metamodel type, choose global metamodel fitting DOE
G4: Conduct simulation model experiments
G5: Fit global metamodel and check fit for adequacy
G6: Is model adequate?
n
y
G7: Change experiment design or metamodel type
G8: Apply global optimization algorithm to global metamodel
G9: Make simulation runs to validate candidate optimum point
G10: Is result satisfactory?
n
y
L10/G10: Finish: Validate Performance for Real System

Fig. 4. Global and local metamodel-based optimization strategies.

rem implies that linear and quadratic polynomial models can provide adequate fit. This is the scenario for response surface methodology. Of course, determining the meaning of 'local' is critical to the adequacy of these metamodels and to the success of the method. If the local region is too small, response surface characteristics will be swamped by variation in the simulation model output. If the local region is chosen too large, linear or quadratic approximations will be inadequate.

Global metamodel fits using polynomial response surface metamodels are rarely adequate. Instead, spline, neural network, spatial correlation or radial basis function approximation is recommended. The experiment designs for fitting global approximation metamodels differ from the central composite, factorial and fractional factorial designs of RSM optimization. Orthogonal array-based Latin hypercubes tend to perform well. Since global metamodels can have multiple local optima, a global optimization strategy is recommended.

4 Response surface methodology (RSM)

4.1 Origins and strategy of RSM

Response surface methodology has its origins in the work of Box and Wilson (1951). Their collaboration initiated at a chemical company when solving the problem of determining optimal operating conditions for chemical processes. Response surface methodology is used in many practical applications in which the goal is to identify the levels of p design factors or variables, $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$, that optimize a response, $f(\theta)$, over an experimental region. One of the earliest applications in simulation was by Biles (1974). Other early papers on RSM in a simulation context are referenced by Kleijnen (1975b). Since in simulation, RSM uses linear and quadratic model approximations to the simulation model, it is a metamodel-based optimization method. In simulation applications, the system response is obtained from simulation output data. We represent the simulation model outputs by the vector-valued function, as in (2).

In the previous section we introduced a general metamodel-based optimization strategy, which is similar to the formal RSM algorithms described in Neddermeijer et al. (2000) and Nicolai et al. (2004). Table 2 shows how the general strategy is applied specifically for RSM. The labels (L1–L10) map back to the general local metamodel based optimization strategy, illustrated graphically in Figure 4.

The strategy in RSM is to sequentially explore small (local) subregions of the experimental region and use line searches to find a new experimental subregion closer to the optimum. In this approach, first or second-order polynomial models are fit to observed values, y, of the system response. The observed system response values are obtained by means of an experiment, designed to provide a good model fit. The choice of models and designs is such that a series of first-order polynomial models are utilized initially, in order to approach a region in design space that is close to an optimum. This sequence of local approximations using first-order metamodels followed by line searches

Table 2.
RSM strategy for metamodel-based optimization

Phase I: First-order regression	Phase II: Second-order regression
L1: Determine initial local region	
Small enough so linear approximation adequate, large enough so expected effects will be significant	NA
L2: Choose a local metamodel form. See Section 4.2	
First-order polynomial	Second-order polynomial
L3: Design local metamodel fitting experiment. See Section 4.3	
Fractional factorial plus center point	Central composite, small composite or augmented fractional factorials
L5/L6/L7: Fit local metamodel and check fit for adequacy. Change model if necessary. Lack of fit test and tests for significance of regression coefficients. See Section 4.4	
See Figure 5(a)	See Figure 5(b)
L8/L9: Provide a search direction or optimize the metamodel. See Section 4.5	
Steepest ascent/descent	Direction based on canonical/ridge analysis
L10: Check the performance of the simulation at the metamodel-predicted optimum. Confirmation runs. See Section 4.6	

is often called Phase I (see left column in Table 2). When close to the optimum, one or more iterations using second-order models are used to optimize the response function (Phase II of the RSM approach). New experimental designs are constructed or augmented at the line search optimal point. In addition, the use of replications at a center point allows a 'pure error' calculation, which permits a check for lack of fit. The different actions to take based on the outcome of the regression analysis are summarized in Figure 5(a) for the first-order regression and in Figure 5(b) for the second-order regression. Each case is discussed in more detail in Section 4.4. The process comes to an end once the iteration-to-iteration improvement is not practically significant and a number of confirmation runs have been conducted to validate the results. More details on each of the steps are given in the sections indicated in Table 2.

4.2 Choosing a local metamodel form (L2)

The multiple regression model for Phase I is a first-order polynomial model,

$$Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \varepsilon, \qquad \varepsilon \sim \text{i.i.d.}\, \mathrm{N}(0, \sigma^2). \tag{10}$$

	$\beta = 0$	$\beta \neq 0$
LOF	AUGMENT DESIGN AND FIT QUADRATIC MODEL	AUGMENT DESIGN AND FIT QUADRATIC MODEL **GO TO PHASE II**
NO LOF	CHOOSE LARGER RANGE FOR A NEW FIRST-ORDER DESIGN ------OR------ INCREASE THE NUMBER OF REPLICATIONS	LINE SEARCH IN NEGATIVE GRADIENT DIRECTION **GO TO L8/L9**

(a)

	$\beta = 0$	$\beta \neq 0$
LOF	UNLIKELY - IF THIS OCCURS, CHOOSE SMALLER RANGE FOR A NEW SECOND-ORDER DESIGN	CHOOSE SMALLER RANGE FOR A NEW SECOND-ORDER DESIGN
NO LOF	UNLIKELY - IF THIS OCCURS, INCREASE THE NUMBER OF REPLICATIONS	LINE SEARCH IN DIRECTION BASED ON CANONICAL ANALYSIS **GO TO L8/L9**

(b)

Fig. 5. Actions based on model adequacy for RSM-based optimization. (a) Model adequacy for Phase I; (b) Model adequacy for Phase II.

Similarly, a full second-order model for Phase II is

$$
\begin{aligned}
&Y(\theta) = \beta_0 + \sum_{j=1}^{p} \beta_j \theta_j + \sum_{i=1}^{p} \sum_{k=i}^{p} \beta_{ik} \theta_i \theta_k + \varepsilon,\\
&\varepsilon \sim \text{i.i.d.}\, \mathrm{N}\big(0, \sigma^2\big).
\end{aligned}
\tag{11}
$$

The quantitative variables θ are often replaced by coded variables, which are typically scaled to $+/-1$,

$$
x_i = \frac{\theta_i - (\theta_{i_{\min}} + (\theta_{i_{\max}} - \theta_{i_{\min}})/2)}{\theta_{i,\max} - (\theta_{i_{\min}} + (\theta_{i_{\max}} - \theta_{i_{\min}})/2)} \quad \text{for } i = 1, \ldots, p.
$$

Let D denote the *design matrix*, which is different from the matrix of design parameter values used in the fitting runs $(\,\theta^1 \ldots \theta^i \ldots \theta^n\,)'$. Each column in the matrix corresponds to the function to be multiplied by the corresponding β coefficient in (10) or (11), say $\phi_i(\theta)$ for coefficient β_i. Even for a first-order polynomial regression shown in (10), D and $(\,\theta^1 \ldots \theta^i \ldots \theta^n\,)'$ are not the same; D is augmented with an initial column of ones for the intercept term (i.e., coefficient β_0), as shown in Equation (12). For the $p = 2$ case with n experiment runs,

$$
\begin{aligned}
&(\,\theta^1 \;\ldots\; \theta^i \;\ldots\; \theta^n\,)' = \begin{pmatrix} \theta_1^1 & \theta_2^1 \\ \theta_1^i & \theta_2^i \\ \theta_1^n & \theta_2^n \end{pmatrix},\\
&D = \begin{pmatrix} \phi_0(\theta^1) & \phi_1(\theta^1) & \phi_2(\theta^1) \\ \phi_0(\theta^i) & \phi_1(\theta^i) & \phi_2(\theta^i) \\ \phi_0(\theta^n) & \phi_1(\theta^n) & \phi_2(\theta^n) \end{pmatrix} = \begin{pmatrix} 1 & \theta_1^1 & \theta_2^1 \\ 1 & \theta_1^i & \theta_2^i \\ 1 & \theta_1^n & \theta_2^n \end{pmatrix},
\end{aligned}
\tag{12}
$$

where scaled x values could be substituted for θ values throughout.

The multiple regression metamodel that is constructed assuming a true response of the form (10) or (11) substitutes 0 for ε and estimates (denoted by b's) for the unknown β coefficients. The b vector is calculated using an existing set of (x, y) data, where x_j^i is the value of the jth design parameter $(j = 1, 2, \ldots, p)$ in the ith run of the system $(i = 1, 2, \ldots, n)$. Let x^i denote the vector of values for the ith run. Finally, y^i is the value of the response in the ith run of the system.

For the single response case, under the assumption of independent, identically distributed random perturbations from run to run, b is found by solving the least-squares equations,

$$b = \left(D'D\right)^{-1}D'y. \tag{13}$$

For many simulation situations, the usual assumption $\varepsilon \sim$ i.i.d. $\mathrm{N}(0, \sigma^2)$ does not hold. In many cases this is because the variance increases with the mean. In some cases it is by deliberate intent, through the use of common and antithetic random numbers, for example. In this more general setting normality is still assumed, but the perturbations can be dependent and have different variances. In this case one has $\varepsilon \sim \mathrm{N}(0, \Sigma_Y)$, where Σ_Y is the variance–covariance matrix for the ε values. The vector β can then be estimated using weighted least squares with weight matrix $W = \Sigma_Y^{-1}$

$$b = \left(D'WD\right)^{-1}D'Wy.$$

In most cases Σ_Y is unknown and W is an estimate of Σ_Y^{-1} based on sample data.

In some cases, a transformation of the response produces i.i.d. error. There are a number of transformations that can be used for variance stabilizing purposes and to improve the analysis. A family of power transformations has been proposed by Box and Cox (1964). These transformations are of the form

$$y^\lambda = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \lambda \neq 0, \\ \log(y), & \lambda = 0, \end{cases}$$

where λ typically takes on values of -1 (reciprocal), 0 (log), 1/2 (square root) and 2 (square). Statistical software can estimate the value of λ for the Box–Cox transformation by the method of maximum likelihood. Alternatively, a variance stabilizing transformation can be selected via a plot of log standard deviation of response vs. log mean over all design points. See Chapter 3 of Montgomery (2001) for a detailed discussion of variance-stabilizing transformations. In practice, the analysis is conducted with the transformed response. The results are then transformed back to their original scale for easier interpretation. Transformations of the response are also discussed in Kleijnen (1987).

4.3 Designing local metamodel fitting experiments (L3)

RSM estimates first and second derivatives by regression over a finite region. The concept is similar to the idea of local vs. infinitesimal sensitivity analysis. RSM uses a local measure by running the simulation at a number of points, often with repeated runs at each point (replications) because the response is stochastic. This set of points is the experiment design for that particular RSM step. There are a number of issues to be addressed in order to create such a fitting data set. For simulation experiments, these include (i) choosing the experimental conditions (the θ^i's), (ii) choosing the simulation run lengths, and (iii) choosing the pseudo-random number assignment strategy within and across runs.

There are many criteria for designing a fitting experiment. In the context of response surface methodology:

- the design should result in a good fit of the model to the data in a minimum number of experimental runs,
- it should control errors due to both variance and bias,
- it should allow models of increasing order to be constructed sequentially and allow experiments to be conducted in blocks to accommodate Phase I and Phase II models, and
- it should be robust to the presence of outliers in the data, should allow for lack of fit testing and should provide an estimate of experimental error.

Equations (10), (11) and (13) imply that b can be characterized as a random variable with $\mathrm{E}(b) = \beta$, with variance–covariance matrix

$$\Sigma_b = \sigma^2 \bigl(D'D\bigr)^{-1}$$

and variance of a predicted value at θ_0, say based on b,

$$\mathrm{Var}\bigl(g(\theta_0)\bigr) = \sigma^2 \phi(\theta_0)' \bigl(D'D\bigr)^{-1} \phi(\theta_0),$$

where the ϕ terms are the functions to be multiplied by the corresponding β coefficient as in (12).

Many of the measures of experiment design goodness, that is, the goodness of $(\ \theta^1 \dots \theta^i \dots \theta^n\ )'$, attempt to minimize some measure associated with Σ_b or $\mathrm{Var}(g(\theta_0))$. For example, a confidence ellipsoid for the true vector β has a form based on Σ_b

$$(\beta - b)' \bigl(D'D\bigr)^{-1} (\beta - b) = K_\alpha,$$

where the constant K_α depends on the confidence level desired, $100(1 - \alpha)\%$. Minimizing the volume of this ellipsoid corresponds to maximizing the determinant of $(D'D)$. This is one measure of design goodness. For the non-i.i.d. case one can substitute $(D'WD)$ for $(D'D)$ in the equations above.

Phase I Designs. Common first-order RSM designs include factorial designs, which comprise full and fractional factorial designs. In factorial designs, each dimension of the design space is covered by a series of (typically) uniformly spaced values; their Cartesian product provides a map of the entire response surface of the system. Usually, first-order designs have only two levels for each design parameter. The number of points required in full factorial designs becomes prohibitively large as the number of factors in the model increases, so fractional factorials are often used as an efficient alternative to full factorial designs when there are many design parameters.

Phase II Designs. Common second-order designs include the central composite (CCD, Box and Wilson, 1951), Box–Behnken (Box and Behnken, 1960), small composite (Draper and Guttman, 1986; Draper and Lin, 1990) and three-level experimental designs (Morris, 2000). A CCD is a two level 2^{p-k} or 2^p factorial design, augmented by n_0 center points and two axial points, so that the quadratic effects can be estimated. This design consists of $2^{p-k} + 2p + n_0$ total design points to estimate $2p + p(p - 1)/2 + 1$ model coefficients. Central composite designs have been used effectively for polynomial models with interaction terms; however, they become impractical for a large number of factors, since they are based on expensive factorial designs. Myers and Montgomery (2002) provide a thorough discussion on response surface modeling and application of factorial designs and other geometric design strategies. The Box–Behnken designs are combinations of incomplete block designs that require fewer levels than CCDs. Small composite designs combine axial runs with fractional-factorials. Morris designs are specially constructed fractional-factorial designs. The designs presented by Morris (2000) are easy to construct and have excellent properties.

Note that there is a significant difference in the number of runs required to fit a ‘first-order’ (linear approximation) vs. the number of runs to fit a quadratic approximation. In terms of the minimum number of runs required, each requires as many runs as there are terms in the model. For a first-order model, there are $p + 1$ terms (including the intercept). For a quadratic model, there are a total of $(p+1)(p+2)/2$ terms. For an optimization on seven factors, the linear approximation requires 8 runs and the quadratic model requires 36 runs, not counting replications. For this reason, an important part of the RSM strategy is to use a linear approximation whenever it is adequate. In particular, when the optimization begins, there is no reason to expect that the initial design parameters are near optimal. If the initial point is far from optimal, the gradient direction may provide an adequate search direction, and a linear approximation may provide an adequate fit.

Replicate runs and variance reduction methods

Another important aspect of experimental design is the use of replicates. One consideration when using replicates is determining at which points those replicates should be conducted. While replicates are often done at the center point of a factorial design to provide an estimate of error and detect curvature, there may be situations for which it may be more appropriate to add replicates at some of the factorial points. For example, in the presence of nonhomogeneous variance in the design space, Kleijnen (2005) suggests increasing the number of replicates to reduce the magnitudes of the variances to reduce the noise at single replicates. In earlier work, Kleijnen and van Groenendaal (1995) propose selecting the additional replicates such that the variances of average responses become a constant. Alternatively, one might consider adjusting the run length of the simulation and partition that run into subruns to obtain replicates (Law and Kelton, 2000). These strategies to reduce variance at high-variance design points are plagued by the $\sqrt{n}$ rule: standard deviations decrease only in the square root of the number of replications (or total simulation run length). If the simulation modeler can tolerate higher prediction error in regions with high response means, then variance stabilizing transformations provide a less costly way to achieve equal variance across the design region.

Variance reduction techniques for simulation models are discussed by Donohue (1995) and McAllister et al. (2001), for example. However, some of these strategies may affect the optimality properties of the experimental design, and the final choice of design strategy (including the number and location of replicates) will depend on the computational expense of the simulation model, the objective and the assumptions of the analysis. Myers and Montgomery (2002) provide a more extensive discussion on the effect of replicates and the design choice.

Control of the random numbers in a simulation permits additional manipulation of the variance/covariance structure of the responses. If RSM is being applied to a discrete-event stochastic simulation model, there are special strategies for selecting the random number streams to improve the precision and accuracy of the fitted model. The original paper in this area is Schruben and Margolin (1978). Donohue et al. (1993a, 1993b, 1995) also discuss designs for fitting quadratic models to discrete-event simulation data. The statistical aspect of analysis with induced correlation and/or control variates are described in Nozari et al. (1987) and Tew and Wilson (1992, 1994). See also Donohue (1995).

4.4 Assessing the adequacy of the metamodel fit (L5/L6/L7)

Testing the model adequacy for RSM is done using a lack-of-fit test, a statistical test that compares model fitting error with "pure" variation within replicated observations. This measures the adequacy of the response surface model. The validation strategy is illustrated in Table 2 and Figure 5, and depends on whether the RSM is in Phase I or Phase II.

For Phase I, if there is no lack of fit, and the regression coefficients are statistically significant ($\beta \neq 0$), a new line search begins in the direction suggested by the regression coefficients. If there is lack of fit, or lack of significance of the model, other steps must be taken.

If there is lack of fit, the design should be augmented with additional runs to fit a local quadratic approximation. Either a central composite design or a small composite design is usually used. An alternative is to use a Resolution V 2^{p-k} design combined with $2p$ axial points and a center point. Such designs are called small composite designs (Draper and Guttman, 1986). Optimization steps using quadratic metamodels are considered to be Phase II of RSM.

On the other hand, if there is no lack of fit, but there is no statistical significance for the first-order regression model, then additional replications should be taken at the fractional factorial points, or a new first-order design should be constructed with a larger range for the design variables. The possible actions following the analysis of the first-order regression model are summarized in Figure 5(a).

For Phase II, cases where there is lack of fit or lack of model significance require a modification of the design: either more replications, or a smaller range for the design.

If some or all of the quadratic model coefficients are statistically significant and there is no lack of fit, then the search direction is selected based on the nature of the quadratic model, as described in Section 4.5. The possible actions following the analysis of the second-order regression model are summarized in Figure 5(b).

4.5 Conducting simulation runs in the search direction (L8/L9)

For Phase I, when the analysis shows that the first-order model is adequate, the regression coefficients are used to identify the gradient. For a minimization problem, the line search proceeds in the negative gradient direction (path of steepest descent), typically beginning at the center of the fractional factorial design region. The path of steepest descent is the direction perpendicular to the contour lines of the response surface. The strategy is to conduct a series of experimental runs along the path of steepest descent indicated by the vector of estimated response surface coefficients b. The step size is usually chosen as the distance from the center of the design region to the edge of the design region. Additional steps are taken in this direction until no further reduction in the objective function occurs. Since the response is stochastic, an unusually good or unusually bad observed value might lead to a premature termination of the search. To reduce this chance, Myers and Khuri (1979) recommend a hypothesis test for significance of the change. del Castillo (1997) and Miró-Quesada and del Castillo (2004) fit a (univariate) polynomial to the responses along the search direction and use the predicted minimum as the end of the line search. They also show that the stopping rule based on three consecutive observations without a decrease is effective.

If multiple first-order fits and line searches are performed, conjugate-gradient or quasi-Newton directions might be chosen instead of the gradient direction (see Joshi et al., 1998).

Phase II generally follows a sequence of one or more Phase I iterations. Eventually the first-order model fails to fit the local response function adequately, and the design is augmented to fit a second-order model for the current design region. This initiates Phase II of the method.

During the analysis of the second-order response surface model, it is sometimes convenient to translate the variables to a new center and rotate the axes so that they correspond to the principal axes of the matrix of second derivatives of the quadratic approximation to the underlying response function, also called the Hessian. The new center is the stationary point of the quadratic model (where the gradient is zero), and the process of rotation and translating the axes to the stationary point is called canonical analysis.

A canonical analysis permits an easy characterization of the local shape of the response function and helps determine whether the estimated stationary point is a maximum, a minimum, or a saddle point. If the eigenvalues of the Hessian are all positive, then the quadratic approximation is bowl-shaped, and a predicted minimum exists, which can be found by setting the gradient of the quadratic approximation equal to zero and solving for x^*. The search direction in this case is toward x^*. If the eigenvalues of H are not all positive, then the search is in the direction of the negative gradient of the fitted quadratic, or may be a ridge direction: a direction that gives the best predicted value of the quadratic on a hypersphere of fixed radius (the radius is typically chosen to be 1 or $\sqrt{n}$ if the factorial values of the variables have been scaled to $+/-1$ over the fitting design).

After a line search is completed, RSM may terminate, either because the budget of runs has been exhausted, or because the optimal point of the most recent line search is very close to the optimal point of the previous cycle.

4.6 Validation of the optimum: checking performance (L9/L10)

Once a local optimum has been found, a number of confirmation runs should be conducted to validate the results. In addition, Peterson et al. (2002) and del Castillo and Cahya (2001) discuss the computation of confidence regions on the stationary point of a response surface. A confidence region provides a measure of quality for the point estimate and can be useful when analyzing problems with multiple responses. Furthermore, confidence regions indicate how robust the solution is. This may be of advantage in determining whether a new metamodel-fitting experiment design, centered about the optimum, should be conducted to refine knowledge of the response function near the optimum.

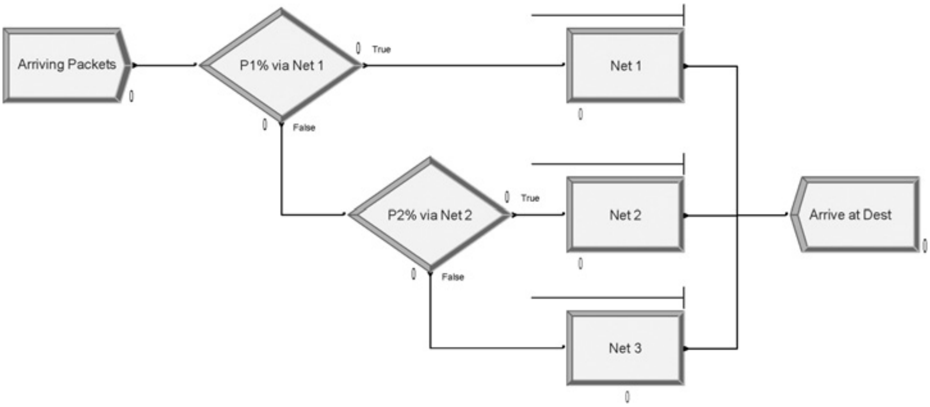


Fig. 6. Arena model for the queuing system design example.

4.7 *Illustration of response surface methodology: queuing example*

There are many operational and strategic issues associated with conducting RSM-based simulation optimization. These are more easily understood in the context of an example, which follows the strategy outlined in Table 2.

Consider a simple network design situation consisting of a communication system in which one must choose routing percentages to route 1000 randomly arriving messages to a particular destination. Suppose that there are three routes (networks) that might be used. One must choose P_1, the percent to network 1 and P_2, the percent of the remaining information packets that go to network 2, to minimize costs. Suppose that costs are composed of \$0.005/time unit each packet is in the system, plus a per-packet processing cost, c_i, that varies by network: \$0.03 for network 1, \$0.01 for network 2 and \$0.005 for network 3. In terms of the general simulation optimization problem, $\theta = (\,P_1\; P_2\,)'$ and $\Theta = [0, 100] \times [0, 100]$, and f is the expected total cost. An Arena model for this system is shown in Figure 6. The Arena simulation environment is described by Kelton et al. (2004). Suppose that packet interarrival times have an exponential distribution with mean $= 1/\lambda = 1$ time unit. Suppose that network transit times have triangular distributions with mean $\mathrm{E}(S)$ and limits $+/-0.5$ with $\mathrm{E}(S) = 1$, 2 and 3 for networks 1, 2 and 3, respectively.

Selecting the design region (L1)

Preliminary simulation runs have revealed that very low as well as very high percentages produce high traffic intensities and high cost. In addition, the initial local metamodel design region should be small enough so a linear approximation is adequate and large enough so that expected effects will be significant. Therefore, the initial region was chosen as $[37, 78] \times [37, 78]$.

Choosing local metamodel type (L2)

The objective of the queuing system analysis is to find values for $\theta = (P_1\ P_2)'$ that minimize the total cost of the system. Suppose that the current settings for P_1 and P_2 are 40% and 75% (the center of the initial design region), and we suspect that the minimum cost is not near the current settings. Therefore, the local surface could be approximated by a first-degree polynomial

$$Y = \beta_0 + \beta_1\theta_1 + \beta_2\theta_2 + \varepsilon, \quad \varepsilon \sim \mathrm{NID}(0, \sigma^2).$$

Designing local metamodel fitting experiment: Phase I (L3)

The design consisted of 10 simulation runs; a 2^2 factorial design with 2 replications at each point was chosen as a first-order design to fit this model. Replicates were used to provide an estimate of the experimental error. In addition, the inclusion of runs at a center point permits a check on the adequacy of the first-order model. The design as well as the data obtained from it are shown in Table 3.

Examining the standard deviations of the response (*cost*) at each of the design points shows some indication of nonhomogeneous standard homogeneous deviations (12.10, 3.26, 14.44, 9.60 and 4.13). A transformation may be appropriate to stabilize the variance. A maximum likelihood estimation of the power transformation parameter suggests a log transformation on the response. The transformed response values have also been included in Table 3. Subsequent analyses for this iteration are done for the transformed response.

Check model adequacy: Phase I (L5/L6/L7)

The analysis of variance and lack of fit for the least squares analysis is summarized in Figure 7. The first-order model is significant, indicated by a P-value of 0.0420 and there is no lack of fit, indicated by the P-value of 0.2049. For this situation, Figure 5(a) suggests continuing with a line search.

Table 3.
Data obtained from a first-order factorial design (centered around $P_1 = 40$, $P_2 = 75$)

Run	Design variables		Coded variables		Response	
	$\theta_1 = P_1$	$\theta_2 = P_2$	x_1	x_2	*cost*	ln(*cost*)
1	37	72	−1	−1	54.40	4.00
2	37	72	−1	−1	37.29	3.62
3	43	72	+1	−1	39.90	3.69
4	43	72	+1	−1	35.29	3.56
5	37	78	−1	+1	64.87	4.17
1	37	78	−1	+1	85.29	4.45
2	43	78	+1	+1	52.83	3.97
3	43	78	+1	+1	39.25	3.67
4	40	75	0	0	66.14	4.19
5	40	75	0	0	60.30	4.10

```
                Response Surface for Variable log(cost)
                Response Mean                          3.942000
                Root MSE                               0.213286
                R-Square                                 0.5958
                Coefficient of Variation                 5.4106

                          Type I Sum
        Regression     DF  of Squares   R-Square   F Value   Pr > F
        Covariates     2   0.469325     0.5958     5.16      0.0420
        Linear         0          0     0.0000     .         .
        Quadratic      0          0     0.0000     .         .
        Crossproduct   0          0     0.0000     .         .
        Total Model    2   0.469325     0.5958     5.16      0.0420

                         Sum of
        Residual      DF  Squares    Mean Square   F Value   Pr > F
        Lack of Fit   2   0.149535   0.074768      2.21      0.2049
        Pure Error    5   0.168900   0.033780
        Total Error   7   0.318435   0.045491
                                                                Parameter
                                                                Estimate
                                  Standard                      from Coded
   Parameter  DF   Estimate    Error      t Value   Pr > |t|   Data
   Intercept  1     3.942000   0.067447   58.45     <.0001      3.942000
   x1         1    -0.168750   0.075408   -2.24     0.0603     -0.168750
   x2         1     0.173750   0.075408    2.30     0.0547      0.173750
```

Fig. 7. Phase I analysis of first-order regression model with transformed response.

Conducting simulation runs in the search direction: Phase I (L8/L9)

With the coefficient estimates for the first-order model the equation of the first-order model can be written as

$$\ln\bigl(g(x)\bigr) = 3.942 - 0.16875x_1 + 0.17375x_2.$$

Rounding the coefficients for the gradient gives $(\,-0.17\ 0.17\,)'$ which can be scaled to $(\,-1\ \ 1\,)'$. Because the objective is to minimize the response, we take a series of steepest descent steps, starting at the center of the initial experimental region and moving $b_2/b_1 = 1/(-1) = -1.0$ units in x_2 for every 1 unit in x_1. Table 4 illustrates this process. Note how the response initially decreases, but then increases again in run 15. As recommended in Section 5.4, three additional steps were done to verify that the increase was not due to variability in the process.

A new 2^2 factorial design can be constructed in the vicinity of run 14; the design is shown in Table 5. This time, there is much less variability in the standard deviations of the responses and we proceed with the analysis of variance and lack of fit for the least squares analysis without a response transformation. This analysis is summarized in Figure 8.

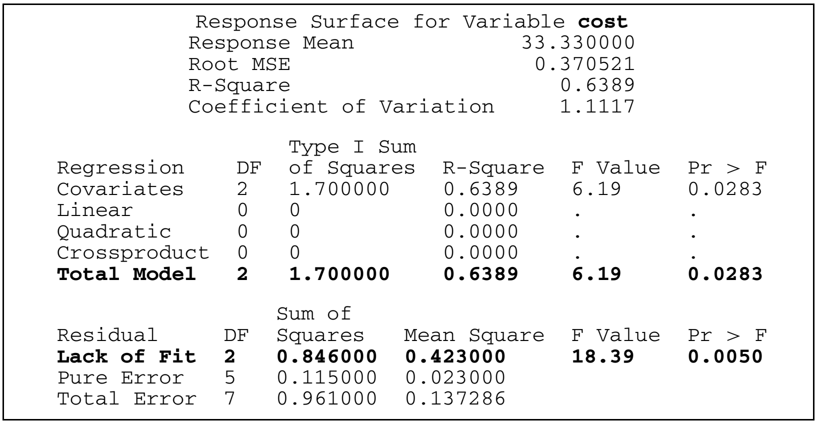
In this case, the model is significant, with a P-value of 0.0283, but the lack of fit test is also significant, with a P-value of 0.005. Following the path suggested in Figure 5(a), the first-order design is augmented to construct a central composite design with axial points at $\delta = \sqrt{2}$, as shown in Table 6.

Table 4.
Steepest descent path for queuing system design

Run	Coded variables		Design variables		Total cost
	x_1	x_2	P_1	P_2	y
Center run conditions	0	0	40	75	63.2 (average)
11	1	−1	43	72	38.3
12	2	−2	46	69	36.2
13	3	−3	49	66	33.3
14	**4**	**−4**	**52**	**63**	**32.6**
15	5	−5	55	60	33.5
16	6	−6	58	57	34.4
17	7	−7	61	54	34.2

Table 5.
Data obtained from a first-order factorial design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
18	49	60	−1	−1	34.3
19	49	60	−1	−1	34.0
20	55	60	+1	−1	33.5
21	55	60	+1	−1	33.5
22	49	66	−1	+1	33.3
23	49	66	−1	+1	33.5
24	55	66	+1	+1	32.8
25	55	66	+1	+1	32.9
26	52	63	0	0	32.9
27	52	63	0	0	32.6

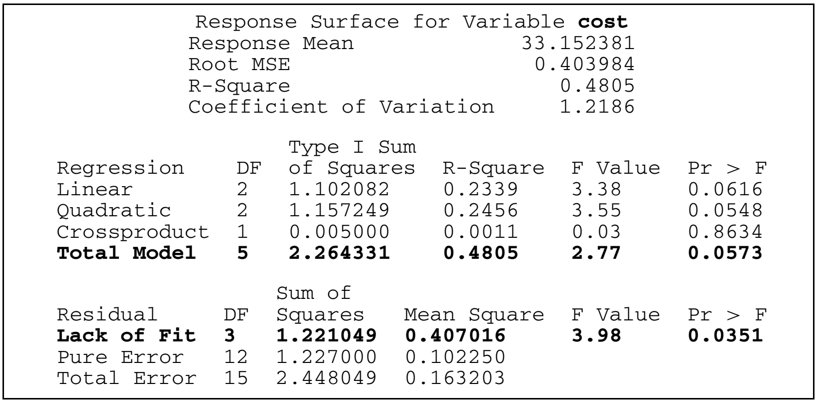


```
                  Response Surface for Variable cost
                 Response Mean                 33.330000
                 Root MSE                       0.370521
                 R-Square                         0.6389
                 Coefficient of Variation         1.1117

                        Type I Sum
Regression     DF   of Squares    R-Square   F Value   Pr > F
Covariates     2    1.700000      0.6389     6.19      0.0283
Linear         0    0             0.0000     .         .
Quadratic      0    0             0.0000     .         .
Crossproduct   0    0             0.0000     .         .
Total Model    2    1.700000      0.6389     6.19      0.0283

                      Sum of
Residual      DF    Squares    Mean Square   F Value   Pr > F
Lack of Fit   2     0.846000   0.423000      18.39     0.0050
Pure Error    5     0.115000   0.023000
Total Error   7     0.961000   0.137286
```

Fig. 8. Phase I analysis of first-order regression model, after the line search.

Table 6.
Data obtained from an augmented design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
28	48	63	−1.41	0	32.7
29	48	63	−1.41	0	33.7
30	56	63	+1.41	0	33.4
31	56	63	+1.41	0	32.8
32	52	59	0	−1.41	32.8
33	52	59	0	−1.41	33.4
34	52	67	0	+1.41	33.2
35	52	67	0	+1.41	32.7
36	52	63	0	0	33.0
37	52	63	0	0	32.7
38	52	63	0	0	32.5



```
                    Response Surface for Variable cost
                   Response Mean              33.152381
                   Root MSE                    0.403984
                   R-Square                      0.4805
                   Coefficient of Variation      1.2186

                             Type I Sum
Regression          DF     of Squares    R-Square    F Value     Pr > F
Linear              2      1.102082      0.2339      3.38        0.0616
Quadratic           2      1.157249      0.2456      3.55        0.0548
Crossproduct        1      0.005000      0.0011      0.03        0.8634
Total Model         5      2.264331      0.4805      2.77        0.0573

                          Sum of
Residual           DF     Squares     Mean Square    F Value     Pr > F
Lack of Fit        3      1.221049    0.407016       3.98        0.0351
Pure Error         12     1.227000    0.102250
Total Error        15     2.448049    0.163203
```

Fig. 9. Phase II analysis of second-order regression model (based on CCD).

Check Model Adequacy: Phase II (L5/L6/L7)

The analysis of the second-order response surface model is summarized in Figure 9.

The analysis of the second-order model shows that there is lack of fit, with *P*-value 0.0351, and marginal model significance, with *P*-value 0.0573. Referring back to Figure 5(b), this is a case that requires choosing a smaller range for a new second-order design. In an attempt to reduce the cost of exercising the simulation code, we first reduce the range of the second-order design by placing the axial points of the CCD at the center of the faces (with $\delta = 1$), creating a face-centered design. The four new points, with 2 replications each, are shown in Table 7.

The analysis of the second-order model based on the face-centered design is shown in Figure 10.

Table 7.
Data for a face centered design (centered around $P_1 = 52$, $P_2 = 63$)

Run	Design variables		Coded variables		Total cost
	P_1	P_2	x_1	x_2	y
39	49	63	−1	0	33.3
40	49	63	−1	0	33.4
41	55	63	+1	0	32.4
42	55	63	+1	0	32.8
43	52	60	0	−1	33.4
44	52	60	0	−1	33.4
45	52	66	0	+1	33.2
46	52	66	0	+1	32.9

```
                  Response Surface for Variable cost
                 Response Mean                33.227778
                 Root MSE                      0.168874
                 R-Square                        0.9108
                 Coefficient of Variation        0.5082

                        Type I Sum
Regression     DF    of Squares    R-Square   F Value    Pr > F
Linear          2    2.288333      0.5965     40.12      <.0001
Quadratic       2    1.200556      0.3130     21.05      0.0001
Crossproduct    1    0.005000      0.0013     0.18       0.6828
Total Model     5    3.493889      0.9108     24.50      <.0001

                      Sum of
Residual       DF    Squares    Mean Square   F Value    Pr > F
Lack of Fit    3     0.097222   0.032407      1.19       0.3672
Pure Error     9     0.245000   0.027222
Total Error    12    0.342222   0.028519
```

Fig. 10. Phase II analysis of second-order regression model.

The adjustment to the design range improved the second-order model with a model significance P-value of less than 0.0001 and a lack of fit P-value of 0.3672. Using Figure 5(b) as a guide, we continue with a line search.

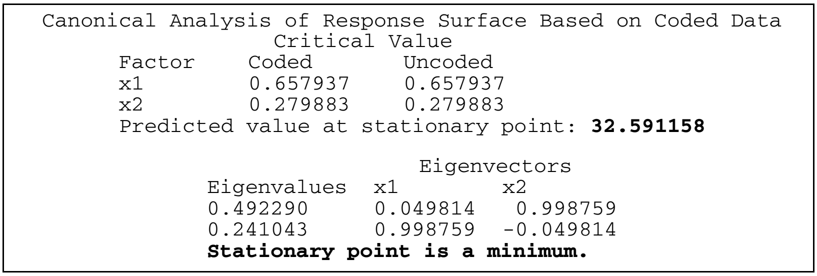
Conducting simulation runs in the search direction: Phase II (L8/L9)

Figure 11 summarizes the results for the canonical analysis, as discussed in Section 4.5.

The stationary point found is a minimum point, with $(\,\theta_1\ \theta_2\,) = (\,54.0\ 63.8\,)$, after translating the coded variables $(\,x_1\ x_2\,) = (\,0.66\ 0.28\,)$ to the original units. The estimated cost at the optimal operating conditions was \$32.59.

Conducting simulation runs in the search direction: Phase II (L8/L9)

To confirm the results from the analysis, 16 additional runs were made at $P_1 = 54.0$ and $P_2 = 63.8$. In practice, the number of confirming runs may depend on how many runs one can afford. The 95% confidence interval for the cost based on 16 replications was [\$32.77, \$33.31]. Compared with the



```
Canonical Analysis of Response Surface Based on Coded Data
                  Critical Value
      Factor    Coded       Uncoded
      x1        0.657937    0.657937
      x2        0.279883    0.279883
      Predicted value at stationary point: 32.591158

                              Eigenvectors
             Eigenvalues  x1          x2
             0.492290     0.049814    0.998759
             0.241043     0.998759   -0.049814
             Stationary point is a minimum.
```

Fig. 11. Phase II canonical analysis for the second-order regression model.

average cost at the starting point, $54.97, the response has improved significantly. We performed a 1000-replication validation for the same point, not likely to be practical in many settings, which provided a confidence interval of [$33.03, $33.10] for the cost.

4.8 RSM for simulation optimization

RSM has been used successfully for over 40 years for processes with stochastic variation. It has been successfully applied to stochastic simulation problems for approximately half that time. The advantage of the method is that it is robust. The disadvantages of the method are that automated versions of the algorithm are not readily available, and manual implementation of the method for more than a few cycles is tedious, complex, and prone to error. In Section 5, we use the network design example to illustrate a global metamodel approach and also provide an analytic solution to this problem.

5 Global metamodel-based optimization

5.1 Motivation and strategy

The developments in global approximation model technology present an opportunity for optimization using a single metamodel, rather than a sequence of fitted local metamodels. There are several potential advantages to this approach. First, a relatively flexible global metamodel may be able to provide a high-fidelity approximation for the response surface with relatively few experimental points, while a polynomial (RSM) metamodel using the same experimental data would fail.

Second, the overall process is simplified: there can be a single experiment design, and a single model-fitting step. This removes the need for sequential decisions on the type of metamodel to be fit and the kind of experiment design to be used for fitting.

Of course, more complex global metamodel-based optimization methods could be designed, for example, to update the fit by selecting additional

Table 8.
General and specific global strategy for metamodel-based optimization

General global strategy	Network design example strategy
G1: Determine global region	Smaller than the feasible region: elimination of obvious nonoptimal regions
G2: Choose a global metamodel type	Smoothing spline
G3: Choose global metamodel fitting DOE	Full factorial (4^2) plus center for smoothing spline
G5/G6/G7: Fit global metamodel and check fit for adequacy. Change model if necessary	Leave-one-out cross-validation sum of squared error. Confirmation runs
G8: Apply global optimization algorithm	Grid search
G10: Check the performance of the simulation at the metamodel-predicted optimum	Confirmation runs

simulation runs as the optimization progressed (Alexandrov et al., 1998; Booker et al., 1999). Such strategies are actually a mix of global and local strategies and will not be considered in this chapter. We will assume that the strategy follows that shown in Figure 4, and that the metamodel-fitting step occurs once, rather than having a sequential update of the design and refitting of the model.

In this section we illustrate a global metamodel-based optimization strategy for the network routing example described in the previous section. Table 8 shows the general global strategy from Figure 2 and the specific implementation that will be used for the network routing optimization example. The process is described in detail in the following sections.

5.2 Selecting the design region (G1)

Recall that for the network design example the design variables $\theta = (P_1 \quad P_2)$. The objective is to find routing percentages that minimize total cost (network use plus transit time) for 1000 messages. The feasible region is $\Theta = [0, 100] \times [0, 100]$, since the routing percentages can be set to any value. Setting a percentage near 100 means that the subsequent network route(s) will not be used, and so the traffic intensity might be too high on the used network(s). On the other hand, a percentage near zero again means that a network will not be used, and that traffic intensities on the remaining networks may be excessive. Some exploratory simulation runs showed that percentages less than 40 or greater than 80 tended to produce high traffic intensities and high costs. For this reason, the global metamodel fitting region was reduced from $[0, 100] \times [0, 100]$ to $[40, 80] \times [40, 80]$.

5.3 Global metamodel type (G2)

Section 2 describes many possible types for the global metamodel. We selected smoothing splines for several reasons. First, they are widely used for response functions of one or two variables. Second, there is publicly available code for fitting and prediction using smoothing splines (Dierckx, 1981, 1993; NETLIB, 2005). Third, they allow for a weighted fit based on observed standard deviations of responses.

5.4 Experiment design (G3)

Bivariate smoothing splines can be used with a variety of design types. We believed the response would be well-behaved (but not quadratic), with substantial increases in cost near the boundaries of the design region. Beyond that, we had no special knowledge of the likely location of the optimum, and chose a 4^2 factorial design to cover the design space, plus a center point ($P_1 = 60$, $P_2 = 60$).

The design consisted of 34 simulation runs, two runs at each of the 17 design points. Table 9 shows the average cost and standard deviation of cost at each design point. Note that there were significant variations in the observed standard deviations. This problem is more likely to occur when fitting global metamodels than when fitting local RSM-type metamodels.

Table 9.
Means and standard deviations for the 17 point, 34 run 4^2 factorial design

Design point	P_1	P_2	Avg. cost	S.D. cost
1	40	40	92.527	22.335
2	40	53	41.200	5.547
3	40	67	34.350	0.776
4	40	80	109.445	40.906
5	53	40	40.485	3.222
6	53	53	34.413	1.572
7	53	67	33.418	0.242
8	53	80	35.275	1.089
9	67	40	35.483	0.327
10	67	53	35.063	0.715
11	67	67	34.350	0.776
12	67	80	34.661	1.250
13	80	40	43.248	3.612
14	80	53	40.711	0.308
15	80	67	39.953	0.676
16	80	80	39.967	1.320
17	60	60	33.293	0.379

5.5 Checking model adequacy (G5, G6, G7)

We expected that a poor model fit might occur since there was substantial variation in standard deviations across the design points. This is a common characteristic for queueing simulations: means and standard deviations of time in system are typically related. We would like to limit the influence of high-variance observations on the overall fit by considering a variance-stabilizing transformation for the response.

Figure 12 shows the log standard deviation vs. the log mean for the 17 design points, which has a slope of approximately 3. This suggests a variance stabilizing transformation of $1/(cost^2)$, as discussed in Montgomery (2001). Figure 13 shows the fitted global metamodel for the untransformed data, using a weight function equivalent to the inverse standard deviation as well as metamodel fitted to the transformed data and using a constant standard devia-

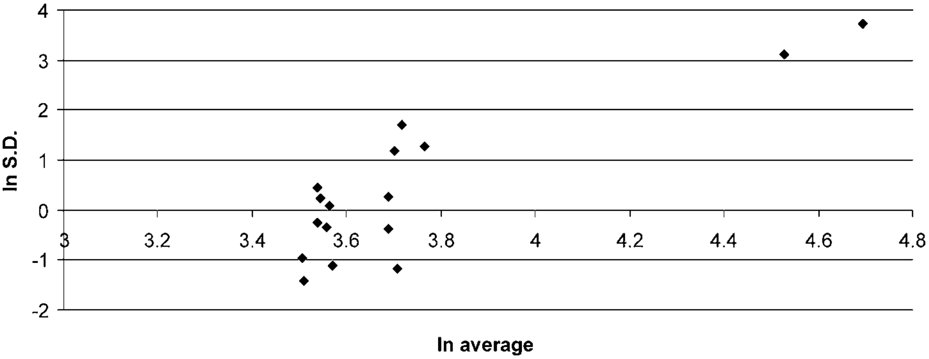
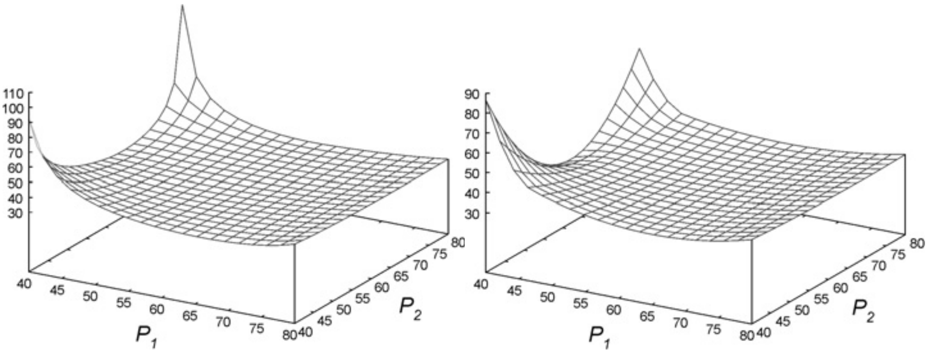


Fig. 12. Log standard deviation vs. log mean for the 17-point 4^2 design.



(a)

(b)

Fig. 13. Fitted global metamodels for the 17-point 4^2 design: (a) $1/y^2$ transformation, (b) no transformation.

tion assumption. The order of magnitude of the smoothing parameter in each case was chosen to provide good leave-one-out cross-validation results. For both the transformed (T-Model) and untransformed (U-Model) response, the smoothing parameter was set to 1.7. Although leave-one-out cross-validation has been shown useful in metamodel assessment (Meckesheimer et al., 2002), it has some drawbacks (Shao, 1993; Tibshirani, 1996). For that reason the two model fits were checked against 10 replications of four randomly selected confirmation design points.

The cross-validation and confirmation run results are summarized in Table 10. While the untransformed metamodel provides better cross-validation results, the transformed model provides lower error (though not statistically significant) for the confirmation runs. More than three times as many confirmation runs would be required to detect the observed differences as statistically significant. There is not a strong reason to select one model over the other, but we continue with the transformed response metamodel to parallel the analysis in the RSM section.

Table 10.
Validation results for metamodels with transformed (T-model) and untransformed (U-model) responses

P_1	P_2	Cost	U-Model	U-Error^2	T-Model	T-Error^2
Cross-validation results						
40	67	34.35	37.46	9.69	41.18	46.72
40	80	109.45	46.55	3956.12	35.10	5527.71
40	53	41.20	39.87	1.77	46.34	26.41
40	40	92.53	85.31	52.03	10.70	6695.82
80	67	39.95	39.62	0.11	41.30	1.82
80	80	39.97	41.43	2.15	34.78	26.90
80	53	40.71	41.25	0.29	48.88	66.77
80	40	43.25	42.18	1.13	35.35	62.36
53	67	33.42	32.43	0.98	32.21	1.46
53	80	35.28	39.74	19.98	40.21	24.34
53	53	34.41	35.15	0.54	34.11	0.09
53	40	40.48	38.71	3.15	43.36	8.27
67	67	34.35	34.40	0.00	34.09	0.07
67	80	34.66	33.35	1.71	36.71	4.19
67	53	35.06	34.61	0.20	34.02	1.08
67	40	35.48	36.38	0.81	37.56	4.33
60	60	33.29	33.59	0.09	33.62	0.11
Average cross-validation squared error			238.28		735.20	
Confirming run validation results						
53	72	33.89	33.80	0.008	34.18	0.084
40	80	34.71	36.33	2.624	35.82	1.232
40	53	34.18	34.30	0.014	34.57	0.152
40	40	36.02	35.16	0.740	35.47	0.302
Average confirming run squared error			0.85		0.44	

5.6 Global optimization results (G8)

The global optimization might proceed with a multistart gradient-based optimizer, identifying the global optimum as in Boender and Rinooy Kan (1987). For a two-variable optimization, a simpler grid search strategy is possible. The optimal operating conditions based on this search are $P_1 = 54.1$ and $P_2 = 64$, with an estimated cost of \$33.34 (for the untransformed response metamodel the results were $P_1 = 57.1$ and $P_2 = 65.6$, with an estimated cost of \$33.14).

5.7 Validation: confirming runs (G9)

Confirming runs of the simulation were made at $P_1 = 54.1$ and $P_2 = 64$. In practical situations it may be cost-prohibitive to conduct many replications. For this example we have chosen a middle ground for replications at 16. This provides a 95% confidence interval of [\$32.75, \$33.22] for the cost. We performed a 1000-replication validation for the same point, not likely to be practical in many settings, which provided a confidence interval of [\$33.05, \$33.11] for the cost. For comparison, the untransformed candidate for optimum produced a 1000-replication confidence interval of [\$33.18, \$33.24], inferior to the transformed response metamodel solution. With 16 replications, the two candidate solutions were indistinguishable.

5.8 An alternative global 'metamodel' based on steady-state behavior

An alternative 'model of the simulation model' is to take the steady-state approximation, which can be solved analytically. This is equivalent to assuming that the network queues begin in steady state, rather than empty and idle.

Suppose that the system to be studied terminates after processing 1000 jobs. The total cost can be approximated using the analytical solution for the steady-state problem. It is possible to decompose the system into three $M/G/1$ queues. In steady state, the average transit time on the ith network will be w_i. If p_i is the probability that a packet is routed through network i and w_i is its average transit time, then the total cost in the steady-state approximation will be

$$\text{Expected total cost for 1000 steady-state customers} \\ = \sum_{i=1}^{3} 1000 p_i (c_i + w_i). \tag{14}$$

The routing probabilities are computed as $p_1 = P_1/100$, $p_2 = (P_2/100)(1 - P_1/100)$ and $p_3 = 1 - [P_1/100 + (P_2/100)(1 - P_1/100)]$. For an $M/G/1$ system in steady state, the average time in the system for an entity is

$$w = \mathrm{E}(S) + \lambda \frac{(\mathrm{E}(S))^2 + \sigma^2}{2(1 - \lambda \mathrm{E}(S))}.$$

The queue for machine 1 has $\lambda_1 = \lambda(P_1/100) = P_1/100$, $\mu_1 = 1/\mathrm{E}(S_1) = 1$ and $\sigma_1^2 = [0.5^2 + 1^2 + 1.5^2 - (0.5)(1) - (0.5)(1.5) - (1)(1.5)]/18 = 1/24$, which gives

$$w_1 = 1 + \frac{(P_1/100)(1 + (1/24))}{2(1 - (P_1/100))}.$$

Similarly,

$$w_2 = 2 + \frac{(P_2/100)(1 - (P_1/100))(4 + (1/24))}{2(1 - ((P_2/100)(1 - (P_1/100))2))}$$

and

$$w_3 = 3 + \frac{(1 - ((P_1/100) + (P_2/100)(1 - P_1/100)))(9 + (1/24))}{2(1 - (1 - ((P_1/100) + (P_2/100)(1 - P_1/100))3))}.$$

By solving these equations for the values of P_1 and P_2 that minimize the expected total cost in Equation (14), we find that the optimum is at \$33.10 with $P_1 = 53.5$ and $P_2 = 63.1$.

The validation runs for this design point give a 1000-replication average of \$33.04 and a 95% confidence interval for the mean of [\$33.01, \$33.07]. One would expect the true optimal routing values to transfer a bit more traffic to routes 2 and 3 than the optimal steady-state solution, with slightly lower cost, since at the start the routes are not congested. This implies that $(\,53.5 \quad 63.1\,)$ may overestimate the percentages. The differences in performance are very small in this neighborhood, however, and not much more can be gained over the design identified by the analytical metamodel. For example the slight reduction of $P_1 = 53.1$ and $P_2 = 62.7$ drops less than \$0.02 in observed average cost for a 1000-replication validation, statistically indistinguishable from the expected cost for the simulation at the analytic solution.

6 Summary

Optimization of computationally costly simulations can be approximated by optimizing metamodels as surrogates for the costly simulation response functions. Local metamodels can be used within an iterative optimization strategy, developed or updated as the optimization progresses. Alternatively, global metamodels can be fitted once, based on a set of simulation runs from a global experiment design, and then the optimization can proceed iteratively using the same metamodel. In either case, it is important to (i) choose the metamodel form carefully, (ii) choose an experiment design appropriate for fitting that type of metamodel, and (iii) validate the metamodel fit and the predicted optimal operating conditions.

Metamodel-based optimization has two distinct benefits over other simulation optimization approaches: a reduction in prediction error that comes from an aggregation of error across many design points, and a representation that often permits insight into the behavior of the response function.

Metamodels can also be used for optimization in a robust design context. In that setting, the objective is to seek an ideal mean response while minimizing its variance. Ramberg (Ramberg et al., 1991), Sanchez (2000) and others cited by these authors discuss the metamodel approach to robust design.

Nearly thirty years after the coining of the term, metamodel-based optimization continues to be an active area of research. A number of researchers are studying how Bayesian methods can improve the fitting and prediction processes (Cheng and Currie, 2004; Chick, 2004), others examine the sequential design of fitting experiments for Bayesian and other metamodels (Kleijnen and van Beers, 2004; Santner et al., 2000, 2003; van Beers and Kleijnen, 2004). Barton (2005) examines alternatives to metamodel-based optimization when this optimization serves as a proxy for having an inverse function metamodel. Myers (1999) gives a fairly recent summary of research issues for RSM.

Finally, general purpose metamodel functions are not always the best approach: when the simulation models are relatively simple, analytic approximations may be used, and can serve as equally effective metamodels.

Acknowledgements

This work was partially supported by NSF Grant No. DMI-0084918. The authors appreciate the suggestions made by the editors, Barry Nelson and Shane Henderson, and by Averill Law. The authors also acknowledge helpful discussions with Andrew J. Booker and Stephen P. Jones from Applied Statistics, Phantom Works, the Boeing Company.

References

Alexandrov, N., Dennis Jr., J.E., Lewis, R.M., Torczon, V. (1998). A trust region framework for managing the use of approximate models in optimization. *Structural Optimization* 15, 16–23.

Andradóttir, S. (1998). A review of simulation optimization techniques. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 151–158.

April, J., Glover, F., Kelly, J.P. (2003). Practical introduction to simulation optimization. In: Chick, S., Sánchez, P.J., Ferrin, D., Morrice, D.J. (Eds.), *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 71–78.

Banks, J., Carson II, J.S., Nelson, B.L., Nicol, D.M. (2005). *Discrete-Event System Simulation*, 4th edition. Prentice Hall, Upper Saddle River, NJ.

Barton, R.R. (1992). Metamodels for simulation input–output relations. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 289–299.

Barton, R.R. (1998). Simulation metamodels. In: Medeiros, D.J., Watson, E.F., Carson, J.S., Manivannan, M.S. (Eds.), *Proceedings of the 1998 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 167–174.

Barton, R.R. (2005). Issues in development of simultaneous forward-inverse metamodels. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 209–217.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M. (1993). *Nonlinear Programming: Theory and Algorithms*, 2nd edition. Wiley, New York.

Bertsekas, D.P. (1999). *Nonlinear Programming*, 2nd edition. Athena, Nashua, NH.

Biles, W.E. (1974). A gradient regression search procedure for simulation experimentation. In: Highland, H.J., Steinberg, H., Morris, M.F. (Eds.), *Proceedings of the 1974 Winter Simulation Conference*. Association for Computing Machinery, New York, pp. 491–497.

Boender, C.G.E., Rinooy Kan, A.H.G. (1987). Bayesian stopping rules for multistart global optimization methods. *Mathematical Programming* 37, 59–80.

Booker, A.J., Dennis, J.E., Frank, P.D., Serafini, D.B., Torczon, V., Trosset, M.W. (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17, 1–13.

Box, G.E.P. (1954). The exploration and exploitation of response surfaces. Some general considerations and examples. *Biometrics* 10, 16–60.

Box, G.E.P., Behnken, D.W. (1960). Some new three-level designs for the study of qualitative variables. *Technometrics* 2, 455–475.

Box, G.E.P., Cox, D.R. (1964). An analysis of transformations (with discussion). *Journal of the Royal Statistical Society B* 26, 211–246.

Box, G.E.P., Draper, N.R. (1987). *Empirical Model-Building and Response Surfaces*. Wiley, New York.

Box, G.E.P., Wilson, K.B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B* 13, 1–45.

Breiman, L. (1991). The *Π* method for estimating multivariate functions from noisy data (with discussion). *Technometrics* 33, 125–160.

Charnes, A., Cooper, W.W. (1977). Goal programming and multiple objective optimization – part I. *European Journal of Operational Research* 1, 39–54.

Cheng, R.C.H., Currie, C.S.M. (2004). Optimization by simulation metamodelling methods. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 485–490.

Chick, S.E. (2004). Bayesian methods for discrete event simulation. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 89–100.

Craven, P., Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik* 31, 377–403.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, New York.

del Castillo, E. (1997). Stopping rules for steepest ascent in experimental optimization. *Communications in Statistics: Simulation and Computation* 26, 1599–1615.

del Castillo, E., Cahya, S. (2001). A tool for computing confidence regions on the stationary point of a response surface. *The American Statistician* 55, 358–365.

Dierckx, P. (1981). An algorithm for surface fitting with spline functions. *IMA Journal of Numerical Analysis* 1, 267–283.

Dierckx, P. (1993). *Curve and Surface Fitting with Splines*. *Monographs on Numerical Analysis*. Oxford University Press.

Donohue, J.M. (1995). The use of variance reduction techniques in the estimation of simulation metamodels. In: Alexopoulos, C., Kang, K., Goldsman, D., Lilegdon, W. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 194–200.

Donohue, J.M., Houck, E.C., Myers, R.H. (1993a). A sequential experimental design procedure for the estimation of first- and second-order simulation metamodels. *ACM Transactions on Modeling and Computer Simulation* 3, 190–224.

Donohue, J.M., Houck, E.C., Myers, R.H. (1993b). Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces. *Operations Research* 41, 880–902.

Donohue, J.M., Houck, E.C., Myers, R.H. (1995). Simulation designs for the estimation of quadratic response surface gradients in the presence of model misspecification. *Management Science* 41, 244–262.

Draper, N.R., Guttman, I. (1986). Response surface designs in flexible regions. *Journal of the American Statistical Association* 81, 1089–1094.

Draper, N.R., Lin, D.K.J. (1990). Small response-surface designs. *Technometrics* 32 (2), 187–194.

Eubank, R.L. (1988). *Spline Smoothing and Nonparametric Regression*. Dekker, New York.

Floudas, C.A., Pardalos, P.M. (Eds.) (1996). *State of the Art in Global Optimization*. Kluwer, Dordrecht, The Netherlands.

Franke, R. (1982). Scattered data interpolation: Tests of some methods. *Mathematics of Computation* 38, 181–200.

Friedman, J.H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics* 19, 1–141.

Fu, M.C. (1994). Optimization via simulation: A review. *Annals of Operations Research* 53, 199–248.

Fu, M.C. (2001). Simulation optimization. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 53–61.

Fu, M.C. (2003). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14, 192–215.

Glover, F., Kelly, J.P., Laguna, M. (1996). New advances and applications of combining simulation and optimization. In: Charnes, J.M., Morrice, D.J., Brunner, D.T., Swain, J.J. (Eds.), *Proceedings of the 1996 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 144–152.

Grossman, I.E. (Ed.) (1996). *Global Optimization in Engineering Design*. Kluwer, Dordrecht, The Netherlands.

Hardy, R.L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research* 76, 1905–1915.

Hussain, M.F., Barton, R.R., Joshi, S.B. (2002). Metamodeling: Radial basis functions versus polynomials. *European Journal of Operational Research* 138, 142–154.

Jin, R., Chen, W., Simpson, T.W. (2000). Comparative studies of metamodeling techniques under multiple modeling criteria. In: *Proceedings of the 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA*. AIAA, St. Louis, MO (CD-ROM).

Joshi, S., Sherali, H.D., Tew, J.D. (1998). An enhanced response surface methodology (RSM) algorithm using gradient deflection and second-order search strategies. *Computers and Operations Research* 25, 531–541.

Kelton, W.D., Sadowski, R.P., Sturrock, D.T. (2004). *Simulation with Arena*, 3rd edition. McGraw-Hill, New York.

Khuri, A.I., Cornell, J.A. (1987). *Response Surfaces, Designs, and Analyses*. Dekker, New York.

Kleijnen, J.P.C. (1975a). A comment on Blanning's metamodel for sensitivity analysis: The regression metamodel in simulation. *Interfaces* 5, 21–23.

Kleijnen, J.P.C. (1975b). *Statistical Techniques in Simulation, part II*. Dekker, New York.

Kleijnen, J.P.C. (1987). *Statistical Tools for Simulation Practitioners*. Dekker, New York.

Kleijnen, J.P.C. (2005). An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research* 164, 287–300.

Kleijnen, J.P.C., Sargent, R.G. (2000). A methodology for fitting and validating metamodels in simulation. *European Journal of Operational Research* 120, 14–29.

Kleijnen, J.P.C., van Beers, W.C.M. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society* 55, 876–883.

Kleijnen, J.P.C., van Beers, W.C.M. (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research* 165, 826–834.

Kleijnen, J.P.C., van Groenendaal, W. (1995). Two-stage versus sequential sample-size determination in regression analysis of simulation experiments. *American Journal of Mathematical and Management Sciences* 15 (1/2), 83–114.

Kolmogorov, A.N. (1961). On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. *American Mathematical Society Translations Series 2* 17, 369–373.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

Law, A., McComas, M.G. (2002). Simulation-based optimization. In: Yücesan, E., Chen, H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 41–44.

Másson, E., Wang, Y.-J. (1990). Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research* 47, 1–28.

MATLAB (2005). Neural network toolbox for MATLAB. Available at *http://www.mathworks.com/products/neuralnet/*.

McAllister, C.D., Altuntas, B., Frank, M., Potoradi, J. (2001). Implementation of response surface methodology using variance reduction techniques in semiconductor manufacturing. In: Peters, B.A., Smith, J.S., Medeiros, D.J., Rohrer, M.W. (Eds.), *Proceedings of the 2001 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1225–1230.

Meckesheimer, M., Booker, A.J., Barton, R.R., Simpson, T.W. (2002). Computationally inexpensive metamodel assessment strategies. *AIAA Journal* 40, 2053–2060.

Miró-Quesada, G., del Castillo, E. (2004). An enhanced recursive stopping rule for steepest ascent searches in response surface methodology. *Communications in Statistics: Simulation and Computation* 33, 201–228.

Mitchell, T.J., Morris, M.D. (1992). The spatial correlation function approach to response surface estimation. In: Swain, J.J., Goldsman, D., Crain, R.C., Wilson, J.R. (Eds.), *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 565–571.

Montgomery, D.C. (2001). *Design and Analysis of Experiments*, 5th edition. Wiley, New York.

Morris, M.D. (2000). A class of three-level experimental designs for response surface modeling. *Technometrics* 42, 111–121.

Myers, R.H. (1999). Response surface methodology – Current status and future directions. *Journal of Quality Technology* 31, 30–44.

Myers, R.H., Khuri, A.I. (1979). A new procedure for steepest ascent. *Communications in Statistics: Theory and Methods* 8, 1359–1376.

Myers, R.H., Montgomery, D.C. (2002). *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd edition. Wiley, New York.

Neddermeijer, H.G., van Oortmarssen, G.J., Piersma, N., Dekker, R. (2000). A framework for response surface methodology for simulation optimization. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 129–136.

Netlab (2005). How to use Netlab. Available at *http://homepages.cae.wisc.edu/~ece539/software/netlab/intro.htm*.

NETLIB (2005). DIERCKX: Fortran subroutines for calculating smoothing splines for various kinds of data and geometries, with automatic knot selection. Available at *http://www.netlib.org/dierckx/*.

Nicolai, R.P., Dekker, R., Piersma, N., van Oortmarssen, G.J. (2004). Automated response surface methodology for stochastic optimization models with unknown variance. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 491–499.

Nozari, A., Arnold, S.F., Pegden, C.D. (1987). Statistical analysis for use with the Schruben and Margolin correlation induction strategy. *Operations Research* 35, 127–139.

Peterson, J.J., Cahya, S., del Castillo, E. (2002). A general approach to confidence regions for optimal factor levels of response surfaces. *Biometrics* 58, 422–431.

Ramberg, J.S., Sanchez, S.M., Sanchez, P.J., Hollick, L.J. (1991). Designing simulation experiments: Taguchi methods and response surface metamodels. In: Nelson, B.L., Kelton, W.D., Clark, G.M. (Eds.), *Proceedings of the 1991 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 167–176.

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P. (1989). Design and analysis of computer experiments. *Statistical Science* 4, 409–435.

Sanchez, S. (2000). Robust design: Seeking the best of all possible worlds. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), *Proceedings of the 2000 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 69–76.

Santner, T.J., Williams, B.J., Notz, W.I. (2000). Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica* 10, 1133–1152.

Santner, T.J., Williams, B.J., Notz, W.I. (2003). *The Design and Analysis of Computer Experiments*. Spinger-Verlag, New York. PeRK code available at *http://www.stat.ohio-state.edu/~comp_exp/index.html*, June 30, 2005.

Schruben, L.W., Margolin, B.H. (1978). Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. *Journal of the American Statistical Association* 73, 504–525.

Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association* 88, 486–494.

Shin, M., Sargent, R.G., Goel, A.L. (2002). Gaussian radial basis functions for simulation metamodeling. In: Yücesan, E., Chen, C.-H., Snowdon, J.L., Charnes, J.M. (Eds.), *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 483–488.

Simpson, T.W., Mauery, T.M., Korte, J.J., Mistree, F. (1998). Comparison of response surface and kriging models for multidisciplinary design optimization. In: *7th Symposium on Multidisciplinary Analysis and Optimization*. *AIAA-98-4755*. AIAA, St. Louis, MO, pp. 381–391.

Simpson, T.W., Frecker, M., Rothrock, L., Stump, G., Barton, R.R., Ligetti, C. (2003). Assessing the impact of graphical design interfaces on design efficiency and effectiveness. *ASME Journal of Computer and Information Science and Engineering* 3, 144–154.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society B* 36, 111–133.

Tew, J.D., Wilson, J.R. (1992). Validation of simulation analysis methods for the Schruben–Margolin correlation-induction strategy. *Operations Research* 40, 87–103.

Tew, J.D., Wilson, J.R. (1994). Estimating simulation metamodels using combined correlation-based variance reduction techniques. *IIE Transactions* 26, 2–16.

Thurston, D.L., Carnahan, J.V., Liu, T. (1994). Optimization of design utility. *ASME Journal of Mechanical Design* 116, 801–807.

Tibshirani, R. (1996). A comparison of some error estimates for neural network models. *Neural Computation* 8, 152–163.

Tomick, J., Arnold, S., Barton, R.R. (1995). Sample size selection for improved Nelder–Mead performance. In: Alexopolous, C., Kang, K., Lilegdon, W.R., Goldsman, D. (Eds.), *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 341–345.

Tu, C.-H., Barton, R.R. (1997). Production yield estimation by the metamodel method with a boundary-focused experiment design. In: *Proceedings of DETC'97, 1997 ASME Design Engineering Technical Conference*. *DETC97/DTM3870*. ASME, Fairfield, NJ.

van Beers, W.C.M., Kleijnen, J.P.C. (2004). Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. CentER Discussion Paper No. 2004-63, Tilburg University.

Watlington, J.A. (2005). Polynomial and radial basis function example code. Available at *http://web.media.mit.edu/~wad/mas864/src/polynomial.html*.

Yesilyurt, S., Patera, A.T. (1995). Surrogates for numerical simulations; optimization of eddy-promoter heat exchangers. *Computer Methods in Applied Mechanics and Engineering* 121, 231–257.

Zionts, S. (1992). The state of multiple criteria decision making: Past, present, and future. In: Goicoechea, A., Duckstein, L., Zionts, S. (Eds.), *Multiple Criteria Decision Making*. Springer-Verlag, New York, pp. 33–43.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*

DOI: 10.1016/S0927-0507(06)13019-4

Chapter 19

Gradient Estimation

Michael C. Fu

University of Maryland, College Park, USA
E-mail: *mfu@rhsmith.umd.edu*

Abstract

This chapter considers the problem of efficiently estimating gradients from stochastic simulation. Although the primary motivation is their use in simulation optimization, the resulting estimators can also be useful in other ways, e.g., sensitivity analysis. The main approaches described are finite differences (including simultaneous perturbations), perturbation analysis, the likelihood ratio/score function method, and the use of weak derivatives.

1 Introduction

For optimization problems with *continuous-valued* controllable parameters, the availability of gradients is clearly helpful in obtaining improved solutions based on an iterative scheme, and can play a critical role in making a particular problem tractable. This is true in stochastic optimization, as well, especially for such problems based on an underlying simulation model. However, in this stochastic setting, since the outputs are themselves random, finding or deriving *stochastic* gradient estimators can itself be a challenging problem, which constitutes the subject of this chapter.

We write the general simulation optimization problem as follows:

$$\min_{\theta\in\Theta} J(\theta), \tag{1}$$

where $\theta \in \Theta$ is the controllable parameter and $\Theta \subset \mathcal{R}^d$ is the feasible region, i.e., θ is a d-dimensional vector. For the objective function, we use the variable J instead of f as in the preceding and succeeding chapters, because we will use f for the probability density functions (p.d.f.s) of the input random variables required to generate estimates for J.

We describe three examples that will be used in the chapter for illustrative purposes. The first example is the stochastic activity network introduced in

Chapter 1. The input random variables are the individual activity times, and the objective function is the total project duration. The parameters are the mean activity times, i.e., $\theta = (\theta_1, \ldots, \theta_d)$, where θ_i is the mean of the ith activity, and the dimension of the parameter vector is equal to the number of activities. The optimization problem is to minimize the expected project duration as a function of the mean activity times, where a cost is attached to each choice of mean activity time.

The second example is the familiar first-come, first-served (FCFS), single-server queue, with the input random variables being the interarrival and service times, and the output performance measure being time spent in the system. The objective function includes a cost on the service rate, and the optimization problem is

$$\min_{\theta \in (0,1/\lambda)} \mathrm{E}\big[T(\theta)\big] + \frac{c}{\theta}, \tag{2}$$

where c is the service rate cost, $T(\theta)$ is the average system time, θ is the mean service time (hence θ is scalar) and λ is the arrival rate. Usually the problem is considered in steady state, and often specializing to the $M/M/1$ queue, because this leads to an analytically tractable solution that can be compared with that obtained using the simulation optimization algorithm. Note that this is essentially a simplified version of the form of the optimization problem (9) posed at the end of Chapter 1 for the stochastic activity network, whereby the second term in the objective function (c/θ) can be viewed as the first constraint in (9) taken into the objective function using a Lagrange multiplier. The formulation here is a design problem that selects the parameter values "from scratch", as opposed to modifying nominal values of the parameters in an existing configuration.

The third example is an (s, S) inventory control system, which was treated in Chapter 17 as a discrete optimization problem where ranking and selection procedures can be applied. Here, the input random variables are related to demand (amount and possibly timing), with the objective function being a total cost function associated with inventory levels and order amounts, and the optimization problem being to minimize expected total cost by the selection of the inventory control parameters s and S, i.e., $\theta = (s, S)$ is a two-dimensional vector.

Although the main application of gradient estimation emphasized in this chapter is simulation-based optimization, derivative estimation has other important applications in simulation, most notably *sensitivity analysis*. This can be useful in many different contexts, e.g., factor screening to decide which factors are the most critical, and hedging of financial instruments and portfolios. The rest of this chapter is organized as follows. A brief overview of gradient-based simulation optimization is provided. Then the main approaches for stochastic gradient estimation are developed in some detail, including examples, some discussion of desirable theoretical properties and computational requirements. Specific applications in different areas are then described. Other than a few exceptions – such as acknowledging a specific key result – references to the literature will be provided in (deferred to) Sections 8 and 9 on applications and probing further.

We conclude this introduction by explicitly stating an implicit assumption that pervades stochastic gradient estimation research (as well as much of the research reported in this handbook).

Key Implicit Assumption. Each estimate of J at a given θ is expensive to generate.

This is not a very rigorous statement, as it is not even mathematical, but the gist of its implication is that because estimating J requires a nontrivial amount of effort, it behooves us to make use of its output more efficiently. The costliness can be due to a number of reasons:

(a) It is expensive to generate input random variables $\{X_i\}$ used to produce an estimate of J.
(b) A lot of input random variables need to be generated, either because each estimate involves a large number of input random variables, or because a lot of estimates (simulation replications) need to be generated to achieve a desired level of precision.
(c) It is a nontrivial task to go from the input random variables $\{X_i\}$ to the estimate of J.

In most stochastic discrete-event simulation models of practical interest, both (b) and (c) are true. If none of these conditions hold, then the simulation user should probably just use "brute force" finite difference techniques, which are described in Section 3.

2 Gradient-based simulation optimization

The two main approaches for conducting simulation-based optimization can be roughly characterized as follows:

- Carry out all of the simulations first – generally a very large number of replications – and then store things appropriately (random number seeds, the random numbers themselves, or realizations of random variates or possibly sample paths), converting the stochastic problem into a deterministic problem that is based on a large enough set of samples to well approximate the desired problem.
- Carry out a relatively small set of simulations and iteratively improve upon the current solution (or set of solutions in a population-based approach) until a sufficiently good solution is reached or found.

Previously, the first approach might have been hindered for large problems by constraints in computer memory, but that has become less of an issue over the past decade, making it a more attractive option, due to its conceptual simplicity and ability to use the arsenal of deterministic nonlinear optimization algorithms available. Since the topic of this chapter and handbook is not deterministic optimization, we will not delve further along this line; see Section 9 for some references to the development of the theory and application of this approach, which has a number of different names: sample average approximation, sample path optimization, and stochastic counterpart. We note that the gradient estimates discussed in this chapter can also be incorporated into, and often play a critical role in, these algorithms. Often this involves "freezing" the random numbers, to be able to use them to generate a value of the performance measure at any value of the parameter, and this value is treated as a deterministic quantity rather than a sample estimate.

The second approach, as related to the context of this chapter, uses the stochastic analog of gradient-based optimization, which is converted to a zero-finding problem (for the gradient of the objective function) and then addressed using stochastic approximation, which we now describe.

2.1 Stochastic approximation

A natural adaptation of "steepest descent" in deterministic nonlinear optimization is stochastic approximation (SA), an iterative update scheme on the parameter that takes the following general form for finding a zero of the objective function gradient:

$$\theta_{n+1} := \Pi_\Theta\bigl(\theta_n - a_n \widehat{\nabla} J(\theta_n)\bigr), \tag{3}$$

where the "hat" notation denotes an estimate of the gradient $\nabla J(\theta_n)$, $\{a_n\}$ denotes the "gain" (step-size multiplier) sequence, and Π_Θ denotes a projection back into the feasible region Θ when the update (3) takes θ out of Θ. Whereas in second-order Newton–Raphson schemes, a key enhancer is to use the inverse Hessian to estimate the optimal step size, this is much less of a concern in SA, especially in the early stages of the algorithm. In fact, to guarantee almost sure (a.s.) convergence, the gain sequence must vanish in the limit, but not too quickly. The usual condition is that

$$\sum_n a_n = \infty, \qquad \sum_n a_n^2 < \infty.$$

However, in practice, one often decreases the step size to some value at which it is kept constant. Theoretically, this leads to weak convergence (in distribution), at best, which might be unsatisfactory. Note that the sequence need not be deterministic either, in which case the conditions above have to be modified accordingly. There are also Central Limit Theorem results for the asymptotic behavior of θ_n, but these are beyond the scope of this chapter.

When $\widehat{\nabla} J(\theta_n)$ is an unbiased estimator of $\nabla J(\theta_n)$, the SA algorithm is generally referred to as being of the Robbins–Monro type, whereas if $\widehat{\nabla} J(\theta_n)$ is only asymptotically unbiased, e.g., using a finite difference estimate with the difference going to zero at an appropriate rate, then the algorithm is referred to being of the Kiefer–Wolfowitz type. The Robbins–Monro SA algorithm generally has a canonical asymptotic convergence rate of $n^{-1/2}$, in contrast to $n^{-1/3}$ for the Kiefer–Wolfowitz SA algorithm, which takes the following form in the scalar parameter case:

$$\theta_{n+1} := \Pi_{\Theta}\left(\theta_n + a_n \frac{\widehat{J}(\theta_n + c_n) - \widehat{J}(\theta_n - c_n)}{2c_n}\right), \tag{4}$$

where $\widehat{J}$ denotes an estimate of J and $\{c_n\}$ denotes a difference sequence that must also decrease to zero at an appropriate rate satisfying

$$\sum_n a_n c_n < \infty, \qquad \sum_n \frac{a_n^2}{c_n^2} < \infty.$$

Thus, in addition to having a slower canonical asymptotic convergence rate, a Kiefer–Wolfowitz SA algorithm involves the additional selection of an appropriate difference sequence. In certain special cases involving common random numbers, however, the best $n^{-1/2}$ rate can also be achieved in practice. A common form for the two sequences is $a_n = an^{-1}$ for some positive a (harmonic series) and $c_n = cn^{-1/6}$ for positive c. If

$$\frac{\widehat{J}(\theta_n + c_n) - \widehat{J}(\theta_n)}{c_n} \tag{5}$$

is used instead for the gradient estimator in the Kiefer–Wolfowitz SA algorithm (i.e., one-sided forward difference gradient estimator), then $c_n = cn^{-1/4}$ is commonly used.

Perhaps one of the key drawbacks of using an SA algorithm, especially for the new or inexperienced user, is the sensitivity of the early "transient" convergence rate to the choices of these sequences. For example, if the common sequences just mentioned are used, then the behavior will depend on the choices of a and c. If a is too small, then the algorithm will "crawl" towards the optimum, even at the $1/\sqrt{n}$ asymptotic rate. On the other hand, if a is chosen too large, then extreme oscillations may occur, resulting in an "unstable" progression. As far as we know, all of the theoretical results relating to convergence rate of these types of algorithms are *asymptotic* results, which may be of little practical use in some applications where all of the action occurs in a relatively short time.

Another drawback is that in general the SA algorithm converges only to a local optimum, although this can also be strengthened by the appropriate introduction of "noise" into the algorithm. Finally, in some settings, it may not be obvious how to project onto the feasible region Θ, which might be specified indirectly (e.g., in a mathematical programming formulation) and possibly involve “noisy” constraints that also have to be estimated along with the objective function. These are practical issues that still have not been adequately addressed for simulation optimization.

3 Indirect gradient estimation

We divide the approaches to stochastic gradient estimation into two main categories – indirect and direct – which we now define. An indirect gradient estimator usually has two characteristics: (i) it only estimates an *approximation* of the true gradient value, e.g., via a secant approximation in the scalar case; and (ii) it uses only function evaluations (performance measure output samples) from the original (unmodified) system of interest. A direct gradient estimator tries to estimate the true gradient using some additional analysis of the underlying stochastics of the model. More specifically, we will refer to the indirect gradient estimation approach as one in which the simulation output is treated as coming out of a given black box, by which we mean it satisfies two assumptions: (i) no knowledge of the underlying mechanics of the simulation model is used in deriving the estimators, such as knowing the input probability distributions; and (ii) no changes are made in the execution of the simulation model itself, such as changing the input distribution for importance sampling. Note that this entails satisfying *both* assumptions; many of the direct gradient estimation techniques can be *implemented* without changing anything in the underlying simulation, but they may require some knowledge of the simulation model, such as the input distributions or some of the system dynamics. In the case of stochastic simulation, as opposed to online estimation based on an actual system, it could be argued that to carry out the simulation most of these mechanics need to be known, i.e., one cannot carry out a stochastic simulation without specifying the input distributions. Here, we simply use the two assumptions to distinguish between the two categories of approaches and not to debate whether an estimator is “model” dependent or not. In terms of stochastic approximation algorithms, indirect and direct gradient estimators generally correspond to Kiefer–Wolfowitz and Robbins–Monro algorithms, respectively.

We describe two indirect gradient estimators: finite differences and simultaneous perturbations. Following our definition, these approaches require no knowledge of the workings of the simulation model, which is treated as a black box.

3.1 Finite differences

The “brute force” or “naïve” method for estimating a gradient at θ is simply to use finite differences, i.e., perturbing the value of each component of θ separately while holding the other components at the nominal value. If the value of the perturbation is too small, the resulting difference estimator could be extremely noisy, because the output is stochastic; hence there is a trade-off between bias and variance in making this selection, and unless all components of the parameter vector are suitably "standardized" a priori, this choice must be done for each component separately, which could be a burdensome task for high-dimensional problems. If the goal is sensitivity analysis rather than optimization, then one would generally err on the side of selecting a relatively larger value for the perturbation.

The ith component of the one-sided forward difference gradient estimator is given by

$$\frac{\widehat{J}(\theta + c_i e_i) - \widehat{J}(\theta)}{c_i}, \tag{6}$$

where c is the vector of differences (c_i the perturbation in the ith direction) and e_i denotes the unit vector in the ith direction.

The ith component of the two-sided symmetric (or central) difference gradient estimator is given by

$$\frac{\widehat{J}(\theta + c_i e_i) - \widehat{J}(\theta - c_i e_i)}{2c_i}, \tag{7}$$

which corresponds to the estimator used in the original Kiefer–Wolfowitz SA algorithm given by (4).

Again, it should be noted that in stochastic simulation, using common random numbers can reduce the variance of the gradient estimators substantially, although in practice synchronization is clearly an issue, since merely using the same random number seeds is typically not effective. The symmetric difference estimator given by (7) is more accurate, but it requires $2d$ objective function estimates (simulation replications) per gradient estimate, as opposed to $d + 1$ function estimates (simulation replications) for the one-sided estimator given by (6). For example, in the stochastic activity network, the symmetric difference estimator would involve performing simulations by varying the mean of each activity i individually by $\pm c_i$ while holding the other activity means at their nominal values, i.e., at parameter settings $\{\theta_i \pm c_i, \theta_j, j \neq i\}$ for $i = 1, \ldots, d$.

3.2 Simultaneous perturbations

This approach has the advantage that the number of simulation replications needed to form an estimator of the gradient is independent of the dimension of the parameter vector.

The ith component of the simultaneous perturbations (SP) gradient estimator is given by

$$\frac{\widehat{J}(\theta + c\Delta) - \widehat{J}(\theta - c\Delta)}{2c_i\Delta_i}, \tag{8}$$

where $\Delta = (\Delta_1, \ldots, \Delta_d)$ is a d-dimensional vector of perturbations, which are generally assumed i.i.d. as a function of iteration and independent across components. In this case, c is again the set of differences for each component, but it is a diagonal matrix with the differences $\{c_i\}$ on the diagonal. The key difference between this estimator and a finite difference estimator is that the *numerator* of (8) – corresponding to a difference in the function estimates – is the *same* for all components (i.e., independent of i), whereas the numerator in the symmetric difference estimator given by (7) involves a different pair of function estimates for each component (i.e., is a function of i). Thus, the full gradient estimator requires only *two* function estimates, regardless of the size of the parameter dimension d. However, since d random numbers must be generated to produce the perturbation sequence Δ at each iteration, if generating the function estimates $\widehat{J}$ is relatively "cheap", then this procedure is likely to be inferior to the previous finite difference approaches. However, our key implicit assumption is that function estimates are expensive, which is generally true in the context of stochastic discrete-event simulation. Furthermore, this procedure may also be of benefit in deterministic situations where J is expensive to generate, e.g., requires a computationally intensive finite element analysis program.

The key requirement on the perturbation sequence to guarantee a.s. convergence when used in a simultaneous perturbation stochastic approximation (SPSA) algorithm is that each term have mean zero and finite inverse second moments. Thus, the normal (Gaussian) distribution is prohibited, and the most commonly used distribution is the symmetric Bernoulli, whereby the perturbation takes the positive and negative (equal in magnitude, e.g., ± 1) value w.p. 0.5. Intuitively, convergence comes about from the averaging property of the random directions selected at each iteration, i.e., in the long-run, each component will converge to the correct gradient even if at any particular iteration the estimator may appear odd. This estimator was designed for optimization via stochastic approximation, and is of limited use in sensitivity analysis, although perhaps averaging over a large number of replications might make it more applicable.

A very similar gradient estimator for use in SA algorithms is the random directions gradient estimator, whose ith component is given by

$$\frac{(\widehat{J}(\theta + c\Delta) - \widehat{J}(\theta - c\Delta))\Delta_i}{2c_i}. \tag{9}$$

Instead of dividing by the perturbation component, the difference term multiplies the component. Thus, normal distributions can be used for the perturbation sequence, and convergence requirements translate the moment condition to a bound on the second moment, as well as zero mean. Of course, a correspondence to the SP estimator can be made by simply taking the componentwise inverse, but in practice the performance of the two resulting SA algorithms differs substantially.

A more recent development in the application of SPSA is to use deterministic sequences for the perturbation sequences $\{\Delta\}$. The idea is analogous to the use of quasi-Monte Carlo, whereby the gradient averaging is the critical factor.

There are also relevant connections to literature in the design of experiments methodology. However, more theoretical work explaining their effectiveness and more numerical examples establishing their advantage over stochastic sequences is needed.

4 Direct gradient estimation

Although indirect gradient estimation offers greater generality, direct gradient estimation has the following advantages:

- It usually provides an unbiased estimator, which leads to faster convergence rates when implemented in a simulation optimization algorithm, e.g., stochastic approximation.
- It eliminates the need to determine appropriate values for the finite difference perturbations – c in the estimators given by (4)–(9) – which influences the accuracy of the estimator. Smaller values lead to lower bias but generally at the cost of increased variance, to the point of possibly giving the wrong sign for small enough values.
- The resulting estimators are usually more computationally efficient. This is almost universally true when compared to high-dimensional brute force finite differences, but not necessarily the case when compared with SP estimators. When used in stochastic approximation, this can result in faster convergence rates, as discussed earlier.

Potential challenges in applying direct gradient estimation include the following:

- They require more "off-line" work, which might be as simple as computing some derivatives of density functions, but could involve quite a bit of problem-specific analysis requiring sophisticated applied probability tools.
- The implementation usually requires some coding inside the simulation model, and sometimes involves some changes in the way the simulation is actually carried out.

For expositional purposes, we will assume that the objective function is an expectation, specifically

$$J(\theta) = \mathrm{E}\big[Y(\theta)\big] = \mathrm{E}\big[Y(X_1, \ldots, X_T)\big], \tag{10}$$

where Y is the (univariate) output performance measure, $\{X_i\}$ are the input random variables and T is a fixed finite number. This covers performance measures of type PM1 and PM2 discussed in Chapter 1, as well as distributional performance measures such as the tail distribution PM3, handled using indicator functions as the performance measure. However, the "dual" performance measure involving quantiles (e.g., median), as in PM4, and also measures such as the mode, cannot be absorbed into this framework. The cases where T is random (a stopping time) or as T goes to infinity (steady state), covering PM5 and PM6, can also be handled, but require some additional technicalities that are beyond the scope of the discussion in this chapter.

In the right-most expression for the performance measure given in (10), the dependence on θ has not been displayed, because where θ appears influences which direct gradient estimation technique is most applicable. In particular, we distinguish between two main dependencies:

- sample (pathwise);
- measure (distributional).

It is important to note that for many performance measures of interest, Y can be expressed such that either dependency can be used. We will see examples of this in the discussion that follows.

To derive direct gradient estimators, we write the expectation using what is sometimes called the law of the unconscious statistician:

$$\mathrm{E}\big[Y(\mathbf{X})\big] = \int y\,\mathrm{d}F_Y(y) = \int Y(\mathbf{x})\,\mathrm{d}F_{\mathbf{X}}(\mathbf{x}). \tag{11}$$

In fact, one of our chief views of stochastic simulation is a way of carrying out this relationship, i.e., coming into the simulation are input random variables, for which we know the distribution; coming out of the simulation are output random variables, for which we would like to know the distributions. However, what we have is a way to generate samples of the output random variables as a function of the input random variables via the simulation model. Of course, if we really knew the distribution of the sample performance output r.v. Y, then there would probably be little reason to have to simulate the system.

For simplicity in discussion, we will assume henceforth that the parameter θ is scalar, because the vector case can be handled by taking each component individually. In view of the right-hand side of (11), we revisit the question as to the location of the parameter in a stochastic setting. Putting it in the sample performance $Y(\cdot;\theta)$ corresponds to the view of perturbation analysis (PA), whereas if it is absorbed in the distribution $F(\cdot;\theta)$, then the approach follows that of the likelihood ratio (LR) method (also known as the score function (SF) method) or weak derivatives (WD) (also known as measure-valued differentiation). In the general setting where the parameter is a vector, it is possible that some of the components would be most naturally located in the sample performance, while others would be easily retained in the distributions, giving rise to a mixed approach. For example, in the (s, S) inventory control case with random order arrival times, it might be most effective to use PA for the control parameters, WD for demand parameters, and LR/SF for order parameters.

Naming the parameter "Spot" for good measure, we pose a question and offer several observations:

(i) Where is Spot?
This may determine which approach is most appropriate. LR/SF and

WD estimators consider distributional parameters, so for them the parameter should appear in the input distribution(s).

(ii) Spot can move!
For the same system, the problem may be formulated such that Spot appears in either spot (location). We will demonstrate this in the examples. In the first two examples, it is simply a matter of interpretation of the underlying stochastics (probability measures), whereas in the (s, S) inventory system, a change of variables is necessary.

(iii) Spot can make repeat appearances.
For example, in the single-server queue example, where the parameter appears in the service time distribution, the parameter must be considered every time a service time is generated.

(iv) Spot can be in two places at once!
It is possible that the parameter appears in both the distribution and in a sample pathwise manner. Also, the parameter could appear in more than one distribution (as opposed to being in the same distribution multiple times, as in the previous item).

Let f denote the joint p.d.f. of all of the input random variables. Differentiating (11), and assuming an interchange of integration and differentiation is permissible, we write two cases:

$$\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} = \int_{-\infty}^{\infty} Y(x)\frac{\mathrm{d}f(x;\theta)}{\mathrm{d}\theta}\,\mathrm{d}x, \tag{12}$$

and

$$\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} = \int_{0}^{1} \frac{\mathrm{d}Y(X(\theta;u))}{\mathrm{d}\theta}\,\mathrm{d}u, \tag{13}$$

where x and u and the integrals are all T-dimensional. For notational simplicity, throughout these T-dimensional multiple integrals are written as a single integral, and we also assume one random number u produces one random variate x. In (12), the parameter appears in the distribution directly, whereas in (13), the underlying uncertainty is considered the uniform random numbers. These correspond to what we called the distributional (measure) and pathwise (sample) dependencies, respectively.

For expositional ease in introducing the approaches, we begin by assuming that the parameter only appears in X_1, which is generated *independently* of the other input random variables. So for the case of (13), we use the chain rule to write

$$\begin{aligned}\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} &= \int_{0}^{1} \frac{\mathrm{d}Y(X_1(\theta;u_1), X_2, \ldots)}{\mathrm{d}\theta}\,\mathrm{d}u \\ &= \int_{0}^{1} \frac{\partial Y}{\partial X_1}\frac{\mathrm{d}X_1(\theta;u_1)}{\mathrm{d}\theta}\,\mathrm{d}u. \end{aligned}\tag{14}$$

In other words, the estimator takes the form

$$\frac{\partial Y(X)}{\partial X_1}\frac{\mathrm{d}X_1}{\mathrm{d}\theta}, \tag{15}$$

where the parameter appears in the transformation from random number to random variate, and the derivative is expressed as the product of a sample path derivative and derivative of a random variable. The issue of what constitutes the latter will be taken up shortly, but this approach is called infinitesimal perturbation analysis (IPA). For the $M/M/1$ queue, the sample path derivative could be derived using Lindley's equation, relating the time in system of a customer to the service times (and interarrival times, which are not a function of the parameter).

Assume that X_1 has marginal p.d.f. $f_1(\cdot;\theta)$ and that the joint p.d.f. for the remaining input random variables $(X_2,\ldots)$ is given by f_{-1}, which has no (functional) dependence on θ. Then the assumed independence gives $f = f_1 f_{-1}$, and the expression (12) involving differentiation of a density (measure) can be further manipulated using the product rule of differentiation to yield the following:

$$\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} = \int_{-\infty}^{\infty} Y(x)\frac{\partial f_1(x_1;\theta)}{\partial\theta} f_{-1}(x_2,\ldots)\,\mathrm{d}x \tag{16}$$

$$= \int_{-\infty}^{\infty} Y(x)\frac{\partial \ln f_1(x_1;\theta)}{\partial\theta} f(x)\,\mathrm{d}x. \tag{17}$$

In other words, the estimator takes the form

$$Y(X)\frac{\partial \ln f_1(X_1;\theta)}{\partial\theta}. \tag{18}$$

Since the term $\frac{\partial \ln f_1(\cdot;\theta)}{\partial\theta}$ is the well-known (efficient) score function in statistics, this approach has been called the score function (SF) method. The other name given to this approach – the likelihood ratio (LR) method – comes from the closely related likelihood ratio function given by

$$\frac{f_1(\cdot;\theta)}{f_1(\cdot;\theta_0)},$$

which when differentiated with respect to θ gives

$$\frac{\partial f_1(\cdot;\theta)/\partial\theta}{f_1(\cdot;\theta_0)},$$

which is equal to the score function upon setting $\theta_0 = \theta$.

On the other hand, if instead of expressing the right-hand side of (16) as (17), the density derivative is written as

$$\frac{\partial f_1(x_1;\theta)}{\partial\theta} = c(\theta)\bigl(f_1^{(2)}(x_1;\theta) - f_1^{(1)}(x_1;\theta)\bigr),$$

it leads to the following relationship:

$$\begin{aligned}\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} &= \int_{-\infty}^{\infty} Y(x)\frac{\partial f(x;\theta)}{\partial\theta}\,\mathrm{d}x \\ &= c(\theta)\biggl(\int_{-\infty}^{\infty} Y(x)f_1^{(2)}(x_1;\theta)f_{-1}(x_2,\ldots)\,\mathrm{d}x \\ &\qquad - \int_{-\infty}^{\infty} Y(x)f_1^{(1)}(x_1;\theta)f_{-1}(x_2,\ldots)\,\mathrm{d}x\biggr).\end{aligned}$$

The triple $(c(\theta), f_1^{(1)}, f_1^{(2)})$ constitutes a weak derivative (WD) for f_1, which is in general not unique, as we shall shortly see. The corresponding WD estimator is of the form

$$c(\theta)\bigl(Y\bigl(X_1^{(2)}, X_2,\ldots\bigr) - Y\bigl(X_1^{(1)}, X_2,\ldots\bigr)\bigr), \tag{19}$$

where $X_1^{(1)} \sim f_1^{(1)}$ and $X_1^{(2)} \sim f_1^{(2)}$. The term "weak" derivative comes about from the possibility that $\frac{\partial f_1(\cdot;\theta)}{\partial\theta}$ in (16) may not be proper, but its *integral* may be well defined, e.g., it might involve delta-functions (impulses), corresponding to mass functions of discrete distributions.

If in the expression (13), the interchange of expectation and differentiation does not hold (e.g., if Y is an indicator function), then as long as there is more than one input random variable, appropriate conditioning will often allow the interchange as follows:

$$\frac{\mathrm{dE}[Y(X)]}{\mathrm{d}\theta} = \int_0^1 \frac{\mathrm{dE}[Y(X(\theta;u))|Z(u)]}{\mathrm{d}\theta}\,\mathrm{d}u, \tag{20}$$

where $Z \subset \{X_1,\ldots,X_T\}$. This conditioning is known as smoothed perturbation analysis (SPA), because it is intended to "smooth" out a discontinuous function. It leads to an estimator of the following form

$$\frac{\partial\mathrm{E}[Y(X)|Z]}{\partial X_1}\frac{\mathrm{d}X_1}{\mathrm{d}\theta}. \tag{21}$$

Note that taking Z as the entire set leads back to (15).

Remark. For SPA, the conditioning in (20) was done with respect to a subset of the input random variables only. Further conditioning can be done on events in the system, which leads to an estimator of the following general form:

$$\frac{\mathrm{d}Y}{\mathrm{d}\theta} + \mathrm{E}_Z[\delta Y|\mathcal{B}]\frac{\mathrm{dP}_Z(\mathcal{B})}{\mathrm{d}\theta}, \tag{22}$$

where the subscript indicates a corresponding conditional expectation/probability, $\mathcal{B}$ is an appropriately selected event and δY represents a change in the performance measure under the conditioned (usually called "critical") event. In this case, if the probability rate $\frac{\mathrm{dP}_Z(\mathcal{B})}{\mathrm{d}\theta}$ is 0, the estimator (22) also reduces to IPA. On the other hand, if the IPA term $\frac{\mathrm{d}Y}{\mathrm{d}\theta}$ is zero, the estimator may coincide with the WD estimator in certain cases, with correspondences between $c(\theta)$ and the probability rate, and between the difference term in (19) and the conditional expectation in (22).

4.1 Desirable properties

Direct gradient estimators are estimators like any other estimators; they just happen to be estimating derivatives. Thus, like other estimators, we would like them to be unbiased.

Definition. The stochastic gradient estimator $\widehat{\nabla}J(\theta)$ is *unbiased* if $\mathrm{E}[\widehat{\nabla}J(\theta)] = \nabla_\theta J(\theta)$.

We would also like the estimators to have low variance. In other words, we want them to be both correct and precise, as opposed to being wrong and noisy. Strong consistency is another desirable property. For finite horizon performance measures, we simply invoke the strong law of large numbers. For infinite horizon or steady-state performance measures, the usual ergodicity considerations come into play, and just as in steady-state output analysis methodology, this can involve a lot of theoretical tools from applied probability.

Basically, unbiasedness requires the exchange of the operations of differentiation (limit) and integration (expectation), as was assumed in deriving (12) and (13). Mathematically, this boils down to whether or not the dominated convergence theorem can be applied (see Section 6). In the case of PA, the bounding involves properties of the performance measure, whereas in LR/SF and WD, the bounding involves the distribution functions. In either case, the primary difficulty is in being able to implement the derivative/gradient, just as building a simulation model is a nontrivial task in implementation, although conceptually there may be little difficulty.

The applicability of IPA may depend on how the input processes are constructed/generated (see the next section). In applying SPA, there is the choice of conditioning quantities (cf. (20) and (21)), which affects how easily the resulting conditional expectation can be estimated from sample paths. There are also a multitude of choices of WD triples for a given input distribution, and this determines both the amount of additional simulation required and the variance of the resulting WD output gradient estimator. For LR/SF estimators, the variance of the estimator could also be a problem if care is not taken in implementation, e.g., the naïve estimator will lead to a linear increase in variance with respect to the simulation horizon.

4.2 Derivatives of random variables

PA estimators – e.g., those shown in (15), (21) and (22) – require the notion of derivatives of random variables. The mathematical problem for defining such derivatives consists of constructing a family of random variables parameterized by θ on a *common* probability space, with the point of departure being a set of parameterized distribution functions $\{F(\cdot;\theta)\}$. We wish to construct $X(\theta) \sim F(\cdot;\theta)$ s.t. $\forall\theta \in \Theta$, $X(\theta)$ is differentiable w.p.1. The sample derivative is then defined in the intuitive manner as

$$\frac{\mathrm{d}X(\theta,\omega)}{\mathrm{d}\theta} = \lim_{\Delta\theta\to 0} \frac{X(\theta+\Delta\theta,\omega) - X(\theta,\omega)}{\Delta\theta},$$

where ω denotes a sample point in the underlying probability space. If the distribution of X is known, we have

$$\frac{\mathrm{d}X(\theta)}{\mathrm{d}\theta} = -\frac{\partial F(X;\theta)/\partial\theta}{\partial F(X;\theta)/\partial X}, \tag{23}$$

where we use the (slightly abusive) notation

$$\frac{\partial F(X;\theta)}{\partial X} = \left.\frac{\partial F(x;\theta)}{\partial x}\right|_{x=X}.$$

Definition. For a distribution function $F(x;\theta)$, θ is said to be a *location* parameter if $F(x+\theta;\theta)$ does not depend on θ; θ is said to be a *scale* parameter if $F(x\theta;\theta)$ does not depend on θ; and θ is said to be a *generalized scale* parameter if $F(\bar{\theta}+x\theta;\theta)$ does not depend on θ, for some fixed $\bar{\theta}$ (usually a location parameter) not dependent on θ.

In these special cases, one can use the following sample derivatives for the three respective cases (location, scale, generalized scale):

$$\frac{\mathrm{d}X}{\mathrm{d}\theta} = 1, \qquad \frac{\mathrm{d}X}{\mathrm{d}\theta} = \frac{X}{\theta}, \qquad \frac{\mathrm{d}X}{\mathrm{d}\theta} = \frac{X-\bar{\theta}}{\theta}.$$

The most well-known example is the normal distribution, with the mean being a location parameter and the standard deviation a generalized scale parameter. Similarly, the two parameters in the Cauchy, Gumbel (extreme value), and logistic distributions also correspond to location and generalized scale parameters. Other examples include the mean of the exponential being a scale parameter; and the mean of the uniform distribution being a location parameter, with the half-width being a generalized scale parameter. In the special case $\mathrm{U}(0,\theta)$, the single parameter is an ordinary scale parameter. Also, for $\mathrm{N}(\theta,(\theta\sigma)^2)$, θ is an ordinary scale parameter.

Lastly, note that for a given distribution, there may be multiple ways to generate a random variate (see Chapter 4), which leads to different derivatives, some of which may be unbiased and some of which may not. This is called the role of representations, which we illustrate with a simple example.

Example 1. Let

$$X \sim \begin{cases} \mathrm{U}(1,2) & \text{w.p.}\theta,\ \theta\in(0,1), \\ \mathrm{U}(0,1) & \text{w.p.}(1-\theta), \end{cases}$$

a mixture of two uniform distributions, with $\mathrm{dE}[X]/\mathrm{d}\theta = 1$. A straightforward construction/representation using two random numbers is

$$X = \mathbf{1}\{U_1 \leqslant \theta\}(1 + U_2) + \mathbf{1}\{U_1 > \theta\}U_2, \tag{24}$$

where $U_1,\ U_2 \sim \mathrm{U}(0, 1)$ are *independent* and $\mathbf{1}\{\cdot\}$ denotes the indicator function. However, since

$$\frac{\mathrm{d}X}{\mathrm{d}\theta} = 0 \quad \text{w.p.1},$$

this clearly leads to a biased estimator. Note that viewed as a function of θ, X jumps from U_2 to $1 + U_2$ at $\theta = U_1$. However, an unbiased estimator can be obtained by using the following construction in which the “coin flipping” and uniform generation are correlated:

$$\begin{aligned} X = {} & \mathbf{1}\{U \leqslant \theta\}\left(1 + \frac{\theta - U}{\theta}\right) \\ & + \mathbf{1}\{U > \theta\}\left(\frac{1 - U}{1 - \theta}\right), \quad \text{where } U \sim \mathrm{U}(0, 1), \\ \Longrightarrow \quad & \frac{\mathrm{d}X}{\mathrm{d}\theta} = \mathbf{1}\{U \leqslant \theta\}\frac{U}{\theta^2} + \mathbf{1}\{U > \theta\}\frac{1 - U}{(1 - \theta)^2}, \end{aligned}$$

which is unbiased (has expectation equal to $\mathrm{dE}[X]/\mathrm{d}\theta = 1$). This construction is based on the property that the distributions of the random variable $(\theta - U)/\theta$ under the condition $\{U < \theta\}$ and the random variable $(1 - U)/(1 - \theta)$ under the condition $\{U \geqslant \theta\}$ are both $\mathrm{U}(0, 1)$. From a simulation perspective, this representation has the additional advantage of requiring only a single random number to generate X instead of two as in the previous construction. In terms of the derivative, the crucial property is that X is continuous across $\theta = U$. One can easily construct other single random number representations that do not have this desirable characteristic, e.g.,

$$\begin{aligned} X = {} & \mathbf{1}\{U \leqslant \theta\}\left(1 + \frac{U}{\theta}\right) + \mathbf{1}\{U > \theta\}\frac{U - \theta}{1 - \theta}, \quad \text{where } U \sim \mathrm{U}(0, 1), \\ \Longrightarrow \quad & \frac{\mathrm{d}X}{\mathrm{d}\theta} = \mathbf{1}\{U \leqslant \theta\}\frac{-U}{\theta^2} + \mathbf{1}\{U > \theta\}\left[-\frac{1 - U}{(1 - \theta)^2}\right], \end{aligned}$$

which is biased (has expectation -1), the intuitive reason being the discontinuity of X at $U = \theta$.

For the first representation, where the “natural” construction leads to a biased estimator, we demonstrate the use of conditioning (SPA) to rectify the situation. First, we derive an alternative estimator by conditioning on U_2, i.e., let

$$X = \mathrm{E}[X_1 | U_2] = (1 + U_2)\theta + U_2(1 - \theta) = \theta + U_2,$$

leading to the obviously unbiased estimator $\mathrm{d}X/\mathrm{d}\theta = 1$.

Alternatively, consider the event

$$\mathcal{B}(\Delta\theta) = \bigl\{U_1 \in (\theta, \theta + \Delta\theta]\bigr\}, \quad \Delta\theta > 0.$$

Intuitively, this event corresponds to those samples such that a perturbation $\Delta\theta > 0$ causes a "jump" in the sample performance from U_2 (when $U_1 > \theta$) to $1 + U_2$ (when $U_1 \leqslant \theta + \Delta\theta$). The complement of this event corresponds to the "smooth" case of IPA. Thus, we write for the representation defined by (24):

$$\begin{aligned}
\frac{\mathrm{dE}[X]}{\mathrm{d}\theta} &= \lim_{\Delta\theta\to 0} \frac{\mathrm{E}[X(\theta+\Delta\theta) - X(\theta)]}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \frac{\mathrm{E}[(X(\theta+\Delta\theta) - X(\theta))\mathbf{1}(\mathcal{B}^{\mathrm{c}}(\Delta\theta))]}{\Delta\theta} \\
&\quad + \lim_{\Delta\theta\to 0} \frac{\mathrm{E}[(X(\theta+\Delta\theta) - X(\theta))\mathbf{1}(\mathcal{B}(\Delta\theta))]}{\Delta\theta} \\
&= \mathrm{E}\left[\frac{\mathrm{d}X}{\mathrm{d}\theta}\right] \\
&\quad + \lim_{\Delta\theta\to 0} \mathrm{E}\bigl[X(\theta+\Delta\theta) - X(\theta)|\mathcal{B}(\Delta\theta)\bigr] \lim_{\Delta\theta\to 0} \frac{\mathrm{P}(\mathcal{B}(\Delta\theta))}{\Delta\theta} \\
&= \lim_{\Delta\theta\to 0} \bigl(\mathrm{E}\bigl[(1+U_2) - U_2\bigr]\bigr) \lim_{\Delta\theta\to 0} \frac{\Delta\theta}{\Delta\theta} = 1, \qquad (25)
\end{aligned}$$

where $\mathcal{B}^{\mathrm{c}}$ denotes the complement of $\mathcal{B}$. In this case, because we can evaluate the difference $(1+U_2) - U_2$ analytically, the final "estimator" is a deterministic quantity equal to the value to be estimated. In more complicated systems, the challenge is usually in estimating this difference of the sample performance under two different conditions on the sample path, given by the limiting conditional expectation in (25).

4.3 Derivatives of measures

As we have seen already, both the LR/SF and WD estimators rely on differentiation of the underlying measure, so the parameters of interest should appear in the underlying (input) distributions. If this is not the case, then one approach is to try to "push" the parameter out of the performance measure into the distribution, so that the usual differentiation can be carried out. This is achieved by a change of variables, which is problem dependent.

Recall that we introduced the idea of a weak derivative by expressing the derivative of a density as the difference of two measures, i.e.,

$$\frac{\partial f(x;\theta)}{\partial\theta} = c(\theta)\bigl(f^{(2)}(x;\theta) - f^{(1)}(x;\theta)\bigr).$$

This idea can be generalized without the need for a differentiable density, as long as the integral exists with respect to a certain set of (integrable) "test" functions, say $\mathcal{L}$, e.g., the set of continuous bounded functions.

Definition. The triple $(c(\theta), F^{(1)}, F^{(2)})$ is called a *weak derivative* for distribution (c.d.f.) F if for all functions $g \in \mathcal{L}$ (not a function of θ),

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\theta}\int g(x)\,\mathrm{d}F(x;\theta)\\ = c(\theta)\biggl(\int g(x)\,\mathrm{d}F^{(2)}(x;\theta) - \int g(x)\,\mathrm{d}F^{(1)}(x;\theta)\biggr).\end{aligned}$$

Remarks. As mentioned earlier, the derivative is "weak" in the sense that the density derivative may not be defined in the usual sense, but in terms of generalized functions integrable with respect to the functions in $\mathcal{L}$, as in the "definition" of a delta function in terms of its integral, i.e., they could include discrete mass functions, as well. The concept of a weak derivative need not be restricted to probability measures, but any finite signed measures. Lastly, note that a WD gradient estimate may require as many as $2d$ additional simulations for the vector case (a pair for each component), unlike LR/SF and IPA estimators, which will always require just a single simulation.

One choice for the weak derivative (density) that is readily available is

$$\frac{\mathrm{d}f}{\mathrm{d}\theta} = c(f_2 - f_1), \tag{26}$$

where

$$f_1 = \frac{1}{c}\biggl(\frac{\mathrm{d}f}{\mathrm{d}\theta}\biggr)^-, \qquad f_2 = \frac{1}{c}\biggl(\frac{\mathrm{d}f}{\mathrm{d}\theta}\biggr)^+, \tag{27}$$

$x^+ \equiv \max(x,0)$, $x^- \equiv \max(-x,0)$ and $c = \int(\frac{\mathrm{d}f}{\mathrm{d}\theta})^+\,\mathrm{d}x = \int f_2\,\mathrm{d}x = \int(\frac{\mathrm{d}f}{\mathrm{d}\theta})^-\,\mathrm{d}x$, using the fact that

$$\int f(x)\,\mathrm{d}x = 1 \quad\Longrightarrow\quad \int \frac{\mathrm{d}f}{\mathrm{d}\theta}\,\mathrm{d}x = 0.$$

The representation given by (26) and (27) is the Hahn–Jordan decomposition, which will always exist for probability measures, and results in a decomposition involving two measures with complementary support.

Remarks. The representation is clearly not unique. In fact, for any nonnegative integrable function h, we have

$$\frac{\mathrm{d}f}{\mathrm{d}\theta} = c\bigl([f_1 + h] - [f_2 + h]\bigr) = c'\biggl(\frac{f_1 + h}{1 + \int h} - \frac{f_2 + h}{1 + \int h}\biggr),$$

where $c' = c(1 + \int h)$. Thus, one way to obtain the estimator using the original simulation is to choose a representation in which both measures have the same support as the original measure. Then importance sampling can be applied, so that the original simulation can be used to generate the estimator without the need for simulating the system under alternative input distributions. Perhaps the most direct way to achieve this is to add the original measure itself to both f_1 and f_2 and renormalize appropriately, i.e., choose $h = f$ above:

$$\frac{\mathrm{d}f}{\mathrm{d}\theta} = 2c\left(\frac{f_1 + f}{2} - \frac{f_2 + f}{2}\right).$$

4.4 Input distribution examples

We now demonstrate some of these concepts on a single random variable. Section 5 will consider the examples introduced at the beginning of the chapter.

Example 2. Let $X \sim \exp(\theta)$, an exponential random variable with mean θ, with p.d.f. given by

$$f(x;\theta) = \frac{1}{\theta}\mathrm{e}^{-x/\theta}\mathbf{1}\{x > 0\}.$$

The usual construction of the r.v. is as follows:

$$X(\theta; u) = -\theta \ln u,$$

where u represents a random number. Differentiating, we get

$$\begin{aligned}
\frac{\mathrm{d}f(x;\theta)}{\mathrm{d}\theta} &= \left[\frac{x}{\theta^2}\frac{1}{\theta}\mathrm{e}^{-x/\theta} - \frac{1}{\theta^2}\mathrm{e}^{-x/\theta}\right]\mathbf{1}\{x > 0\} \\
&= f(x;\theta)\left[\frac{x}{\theta^2} - \frac{1}{\theta}\right] \\
&= \frac{1}{\theta}\left[\frac{x}{\theta^2}\mathrm{e}^{-x/\theta}\mathbf{1}\{x > 0\} - f(x;\theta)\right] \\
&= \frac{1}{\theta\mathrm{e}}\left[\frac{\mathrm{e}}{\theta}\left(\frac{x}{\theta} - 1\right)\mathrm{e}^{-x/\theta}\mathbf{1}\{x > \theta\}\right. \\
&\qquad \left. - \frac{\mathrm{e}}{\theta}\left(1 - \frac{x}{\theta}\right)\mathrm{e}^{-x/\theta}\mathbf{1}\{0 < x \leqslant \theta\}\right], \\
\frac{\mathrm{d}X(\theta;u)}{\mathrm{d}\theta} &= -\ln u = \frac{X(\theta;u)}{\theta}.
\end{aligned}$$

In the third and fourth lines above, the density derivative (which is itself *not* a density) has been expressed as the difference of two densities multiplied by a constant. This demonstrates that the weak derivative representation is not unique, with the last decomposition being the Hahn–Jordan decomposition, noting that $x = \theta$ is the point at which $\mathrm{d}f(x;\theta)/\mathrm{d}\theta$ changes sign. The following correspond to the LR/SF, WD (a) and (b), and IPA estimators, respectively:

$$\begin{aligned}
&Y(X)\frac{1}{\theta}\left(\frac{X_1}{\theta}-1\right),\\
&\frac{1}{\theta}\left[Y\left(X_{1a}^{(2)},\ldots\right)-Y\left(X_{1a}^{(1)},\ldots\right)\right],\\
&\frac{1}{\theta \mathrm{e}}\left[Y\left(X_{1b}^{(2)},\ldots\right)-Y\left(X_{1b}^{(1)},\ldots\right)\right],\\
&\frac{\mathrm{d}Y}{\mathrm{d}X_1}\frac{X_1}{\theta},
\end{aligned}$$

where $X_{1a}^{(1)}$ and $X_{1a}^{(2)}$ are random variables distributed as $\exp(\theta)$ and $\mathrm{Erl}(2, \theta)$, respectively, and $X_{1b}^{(1)}$ and $X_{1b}^{(2)}$ are random variables with densities

$$\frac{\mathrm{e}}{\theta}\left(1-\frac{x}{\theta}\right)\mathrm{e}^{-x/\theta}, \quad 0 < x \leqslant \theta, \quad \text{and} \quad \frac{\mathrm{e}}{\theta}\left(\frac{x}{\theta}-1\right)\mathrm{e}^{-x/\theta}, \quad x > \theta,$$

respectively.

The following is a simple example that demonstrates that the WD estimator is more broadly applicable than the LR/SF estimator.

Example 3. Let $X \sim \mathrm{U}(0, \theta)$. Then we have the following:

$$\begin{aligned}
f(x;\theta) &= \frac{1}{\theta}\mathbf{1}\{0 < x < \theta\},\\
X(\theta;u) &= u\theta,\\
\frac{\mathrm{d}f(x;\theta)}{\mathrm{d}\theta} &= \frac{1}{\theta}\left[\delta(\theta-x)-\frac{1}{\theta}\mathbf{1}\{0<x<\theta\}\right]\\
&= \frac{1}{\theta}\left[\delta(\theta-x)-f(x;\theta)\right],\\
\frac{\mathrm{d}X(\theta;u)}{\mathrm{d}\theta} &= u = \frac{X(\theta;u)}{\theta},
\end{aligned} \tag{28}$$

where we define the Dirac δ-function as the “derivative” of a step function by

$$\mathbf{1}\{x \geqslant \theta\} = \int_{-\infty}^{x} \delta(y-\theta)\,\mathrm{d}y. \tag{29}$$

On the right-hand side of Equation (28), we have the difference of densities for a mass at θ and the original $\mathrm{U}(0, \theta)$ distribution, respectively, i.e., the weak derivative representation $(1/\theta, \theta, F)$, where θ indicates a deterministic distribution with mass at θ. So, for example, the estimator in (19) would be given by

$$\frac{1}{\theta}\left(Y(\theta, X_2, \ldots) - Y(X_1, X_2, \ldots)\right).$$

This is a case where the LR/SF estimator is ill-defined, due to the δ-function.

Another example is the following one.

Example 4. Let $X \sim \mathrm{Par}(\alpha, \theta)$, which represents the Pareto distribution with shape parameter α and scale parameter θ, and p.d.f. given by

$$f(x) = \theta^\alpha \alpha x^{-(\alpha+1)} \mathbf{1}\{x \geqslant \theta\}.$$

Once again the LR/SF estimator does not exist, due to the appearance of the parameter in the indicator function that controls the support of the distribution, whereas WD estimators can be derived (see Table 1 at the end of the section).

However, the very general exponential family of distributions leads to a nice form for the LR/SF estimator.

Example 5. Let θ denote the vector of parameters in a p.d.f. that can be written in the following form:

$$f(x; \theta) = k(\theta) \exp\left(\sum_i v_i(\theta) t_i(x)\right) h(x),$$

where the functions h and $\{t_i\}$ are independent of θ, and the functions k and $\{v_i\}$ do not involve the argument. Then it is straightforward to derive

$$\frac{\partial \ln f(x; \theta)}{\partial \theta} = \frac{\nabla k(\theta)}{k(\theta)} + \sum_i \nabla v_i(\theta) t_i(x).$$

Examples include the normal, gamma, Weibull and exponential, for the continuous case, and the binomial, Poisson and geometric for the discrete case.

Discrete distributions present separate challenges for the two approaches. Basically, when the parameter appears in the support *probabilities*, then LR/SF can be easily applied, whereas IPA is in general not applicable. The reverse is true, however, if the parameter appears instead in the support *values*. Here are two examples to demonstrate this, where we work directly with probability mass functions $p(x; \theta) = \mathrm{P}(X = x)$, instead of densities with δ-functions. Let $\mathrm{Ber}(p; a, b)$ denote a Bernoulli distribution that takes value a with probability p and b with probability $1 - p$.

Example 6. Let $X \sim \mathrm{Ber}(\theta; a, b)$, which has mass function

$$p(x; \theta) = \theta \mathbf{1}\{x = a\} + (1 - \theta) \mathbf{1}\{x = b\},$$

so

$$\frac{\partial p}{\partial \theta} = \mathbf{1}\{x = a\} - \mathbf{1}\{x = b\},$$

which can be viewed as the difference of two (deterministic) masses at a and b (with $c(\theta) = 1$), and is the Hahn–Jordan decomposition in this case. For the LR/SF estimator, we have

$$\frac{\partial \ln p}{\partial \theta} = \frac{\mathbf{1}\{x = a\} - \mathbf{1}\{x = b\}}{\theta \mathbf{1}\{x = a\} + (1-\theta)\mathbf{1}\{x = b\}} = \frac{1}{\theta}\mathbf{1}\{x = a\} - \frac{1}{1-\theta}\mathbf{1}\{x = b\}.$$

In this case, there is no way to construct X such that it will be differentiable w.p.1. For example, the natural construction/representation

$$X = a\mathbf{1}\{U \leqslant \theta\} + b\mathbf{1}\{U > \theta\}$$

yields $\mathrm{d}X/\mathrm{d}\theta = 0$ w.p.1, so IPA is not applicable.

Example 7. Let $X \sim \mathrm{Ber}(p; \theta; 0)$, which has mass function

$$p(x; \theta) = p\mathbf{1}\{x = \theta\} + (1-p)\mathbf{1}\{x = 0\},$$

which is not differentiable with respect to θ, so LR/SF and WD estimators cannot be derived.

The natural random variable construction

$$X = \theta \mathbf{1}\{U \leqslant p\}$$

leads to the unbiased

$$\frac{\mathrm{d}X}{\mathrm{d}\theta} = \mathbf{1}\{U \leqslant p\} = \mathbf{1}\{X = \theta\}$$

(which in this example also happens to equal X/θ). The IPA estimator $\mathrm{d}X/\mathrm{d}\theta = \mathbf{1}\{X = \theta\}$ holds even if any number of additional values are added to the underlying support, if all of them do not involve θ. If θ enters into them, then the estimator can be easily modified to reflect that.

It is straightforward to verify the corresponding derivatives of some common distributions displayed in Table 1, where $\mathrm{geo}(p)$ indicates a geometric distribution with mass function

$$(1-p)^{n-1}p, \quad x = 1, 2, \ldots;$$

$\mathrm{bin}(n, p)$ indicates a binomial distribution with mass function

$$\binom{n}{p} p^x(1-p)^{n-x}, \quad x = 0, 1, \ldots;$$

$\mathrm{negbin}(n, p)$ indicates a negative binomial distribution with mass function

$$\binom{x-1}{n-1}(1-p)^{x-n}p^n, \quad x = n, n+1, \ldots;$$

Table 1.
Derivatives for some common/simple input distributions ("NA" = not applicable)

Input dist	WD	IPA	LR/SF
$X \sim F$	$(c, F^{(2)}, F^{(1)})$	$\frac{\mathrm{d}X}{\mathrm{d}\theta}$	$\frac{\mathrm{d}\ln f(X)}{\mathrm{d}\theta}$
$\mathrm{Ber}(\theta; a, b)$	$(1, a, b)$	NA	$\frac{1}{\theta}\mathbf{1}\{X = a\} - \frac{1}{1-\theta}\mathbf{1}\{X = b\}$
$\mathrm{Ber}(p; \theta, b)$	NA	$\mathbf{1}\{X = \theta\}$	NA
$\mathrm{geo}(\theta)$	$\bigl(\frac{1}{\theta}, \mathrm{geo}(\theta), \mathrm{negbin}(2, \theta)\bigr)$	NA	$\frac{1}{\theta} + \frac{1-X}{1-\theta}$
$\mathrm{bin}(n, \theta)$	$(n, 1 + \mathrm{bin}(n-1, \theta), \mathrm{bin}(n-1, \theta))$	NA	$\frac{X}{\theta} - \frac{n-X}{1-\theta}$
$\mathrm{Poi}(\theta)$	$(1, 1 + \mathrm{Poi}(\theta), \mathrm{Poi}(\theta))$	NA	$\frac{X}{\theta} - 1$
$\mathrm{N}(\theta, \sigma^2)$	$(\frac{1}{\sqrt{2\pi}\sigma}, \theta + \mathrm{Wei}\bigl(2, \frac{1}{2\sigma^2}\bigr), \theta - \mathrm{Wei}\bigl(2, \frac{1}{2\sigma^2}\bigr)\bigr)$	1	$\frac{X-\theta}{\sigma^2}$
$\mathrm{N}(\mu, \theta^2)$	$\bigl(\frac{1}{\theta}, \mathrm{Mxw}(\mu, \theta^2), \mathrm{N}(\mu, \theta^2)\bigr)$	$\frac{X-\mu}{\theta}$	$\frac{1}{\theta}\bigl[\bigl(\frac{x-\mu}{\theta}\bigr)^2 - 1\bigr]$
$\mathrm{U}(0, \theta)$	$\bigl(\frac{1}{\theta}, \theta, \mathrm{U}(0, \theta)\bigr)$	$\frac{X}{\theta}$	NA
$\mathrm{U}(\theta - \gamma, \theta + \gamma)$	$\bigl(\frac{1}{2\gamma}, \theta + \gamma, \theta - \gamma\bigr)$	1	NA
$\mathrm{U}(\mu - \theta, \mu + \theta)$	$\bigl(\frac{1}{\theta}, \mathrm{Ber}(0.5; \mu - \theta, \mu + \theta), U(\mu - \theta, \mu + \theta)\bigr)$	$\frac{X-\mu}{\theta}$	NA
$\exp(\theta)$	$\bigl(\frac{1}{\theta}, \mathrm{Erl}(2, \theta), \exp(\theta)\bigr)$	$\frac{X}{\theta}$	$\frac{1}{\theta}\bigl(\frac{X}{\theta} - 1\bigr)$
$\mathrm{Wei}(\alpha, \theta)$	$\bigl(\frac{\alpha}{\theta}, F^*(\alpha, \theta), \mathrm{Wei}(\alpha, \theta)\bigr)$	$\frac{X}{\theta}$	$\frac{1}{\theta}\bigl[\bigl(\frac{X}{\theta}\bigr)^\alpha - \alpha\bigr]$
$\mathrm{gam}(\alpha, \theta)$	$\bigl(\frac{\alpha}{\theta}, \mathrm{gam}(\alpha + 1, \theta), \mathrm{gam}(\alpha, \theta)\bigr)$	$\frac{X}{\theta}$	$\frac{1}{\theta}\bigl(\frac{X}{\theta} - \alpha\bigr)$
$\mathrm{Par}(\alpha, \theta)$	$\bigl(\frac{\alpha}{\theta}, \mathrm{Par}(\alpha, \theta), \theta\bigr)$	$\frac{X}{\theta}$	NA

$\mathrm{Poi}(\lambda)$ indicates a Poisson distribution (mean/variance λ) with mass function

$$\frac{\mathrm{e}^{-\lambda}\lambda^x}{x!}, \quad x = 0, 1, \ldots;$$

$\mathrm{Erl}(n, \beta)$ indicates an Erlang distribution with p.d.f.

$$\frac{\beta^{-n}x^{n-1}\mathrm{e}^{-x/\beta}}{(n-1)!}\mathbf{1}\{x > 0\};$$

$\mathrm{Mxw}(\mu, \sigma^2)$ indicates a double-sided Maxwell distribution with p.d.f.

$$\frac{(x-\mu)^2}{\sqrt{2\pi}\sigma^3}\mathrm{e}^{-(x-\mu)^2/(2\sigma)};$$

$\mathrm{Wei}(\alpha, \beta)$ indicates a Weibull distribution with shape parameter α and scale parameter β, and p.d.f.

$$\alpha\beta^{-\alpha}x^{\alpha-1}\mathrm{e}^{-(x/\beta)^\alpha}\mathbf{1}\{x > 0\};$$

$\mathrm{gam}(\alpha, \beta)$ indicates a gamma distribution with shape parameter α and scale parameter β, and p.d.f.

$$\frac{\beta^{-\alpha}x^{\alpha-1}\mathrm{e}^{-x/\beta}}{\Gamma(\alpha)}\mathbf{1}\{x > 0\},$$

where

$$\Gamma(\alpha) = \int_0^\infty t^{\alpha-1}\mathrm{e}^{-t}\,\mathrm{d}t = (\alpha-1)\Gamma(\alpha-1);$$

recalling that $\mathrm{Erl}(n,\beta) = \mathrm{gam}(n,\beta)$ for positive integer n; $\mathrm{Wei}(1,\beta) = \mathrm{gam}(1,\beta) = \exp(\beta)$; and the special distribution $F^*(\alpha,\theta)$ has p.d.f.

$$\alpha\beta^{-2\alpha}x^{2\alpha-1}\mathrm{e}^{-(x/\beta)^\alpha}.$$

Note that our definition of the second parameter β in the gamma and Weibull distributions (also corresponding to the only parameter in the exponential distribution) is the inverse of what is usually found in the literature (Law and Kelton (2000) being a notable exception), but leads to it being a scale parameter under our definition.

5 Examples

5.1 Stochastic activity network

Let X_i have p.d.f. f_i, and assume all of the activity times are *independent*. Let P^* denote the set of activities on the optimal path corresponding to the project duration (e.g., shortest or longest path, depending on the problem), so we can write

$$Y = \sum_{j\in P^*} X_j,$$

where P^* itself is a random variable. We wish to estimate $\mathrm{d}E[Y]/\mathrm{d}\theta$. Let θ be a parameter in the distribution of X_1, i.e., in f_1 only. Then, the IPA estimator is given by

$$\frac{\mathrm{d}Y}{\mathrm{d}\theta} = \frac{\mathrm{d}X_1}{\mathrm{d}\theta}\mathbf{1}\{1 \in P^*\}.$$

The LR/SF estimator is given by

$$Y\frac{\partial \ln f_1(X_1;\theta)}{\partial\theta}.$$

The WD estimator is of the form

$$c(\theta)\bigl(Y\bigl(X_1^{(2)}, X_2, \ldots, X_T\bigr) - Y\bigl(X_1^{(1)}, X_2, \ldots, X_T\bigr)\bigr),$$

where $X_1^{(j)} \sim F_1^{(j)}$, $j = 1, 2$, and $(c(\theta), F_1^{(2)}, F_1^{(1)})$ is a weak derivative for F_1.

If we allow the parameter to possibly appear in all of the distributions, then the IPA estimator is found by applying the chain rule:

$$\frac{\mathrm{d}Y}{\mathrm{d}\theta} = \sum_{i\in P^*}\frac{\mathrm{d}X_i}{\mathrm{d}\theta};$$

whereas the LR/SF and WD estimators are derived by applying the product rule of differentiation to the underlying input distribution, applying the independence that allows the joint distribution to be expressed as a product of marginals. In particular, the LR/SF estimator is given by

$$Y(X)\left(\sum_{i=1}^{d} \frac{\partial \ln f_i(X_i;\theta)}{\partial \theta}\right).$$

The WD estimator is of the form

$$\sum_{i=1}^{T} c_i(\theta)\bigl(Y\bigl(X_1,\ldots,X_i^{(2)},\ldots,X_T\bigr) - Y\bigl(X_1,\ldots,X_i^{(1)},\ldots,X_T\bigr)\bigr),$$

where $X_i^{(j)} \sim F_i^{(j)}$, $j = 1, 2$, $i = 1, \ldots, T$, and $(c_i(\theta), F_i^{(2)}, F_i^{(1)})$ is a weak derivative for F_i.

If instead we were interested in estimating $\mathrm{P}(Y > y)$ for some fixed y, the WD and LR/SF estimators would simply replace Y with the indicator function $\mathbf{1}\{Y > y\}$. However, IPA will not apply, since the indicator function is inherently discontinuous, so an extension of IPA such as SPA is required. On the other hand, if the performance measure were $\mathrm{P}(Y > \theta)$, then since the parameter does not appear in the distribution of the input random variables, WD and LR/SF estimators cannot be derived without first carrying out an appropriate change of variables. These cases are addressed in Fu (2006).

5.2 *Single-server queue*

We illustrate all of the gradient estimators for the queueing example. Let A_n be the interarrival time between the $(n-1)$st and nth customer (i.i.d. with p.d.f. f_1 and c.d.f. F_1), X_n the service time of the nth customer (i.i.d. with p.d.f. f_2 and c.d.f. F_2) and T_n the system time (in queue plus in service) of the nth customer. We consider the case where θ is a parameter in the service time distribution, and the sample performance of interest is the average system time over the first N customers $\overline{T}_N = \frac{1}{N}\sum_{n=1}^{N} T_n$. Assume that the system starts empty, so that the times of the first N service completions are completely determined by the first N interarrival times and first N service times.

The system time of a customer for an FCFS single-server queue satisfies the well-known recursive Lindley equation

$$T_{n+1} = X_{n+1} + (T_n - A_{n+1})^{+}. \tag{30}$$

The IPA estimator is obtained by differentiating (30):

$$\frac{\mathrm{d}T_{n+1}}{\mathrm{d}\theta} = \frac{\mathrm{d}X_{n+1}}{\mathrm{d}\theta} + \frac{\mathrm{d}T_n}{\mathrm{d}\theta}\mathbf{1}\{T_n \geqslant A_{n+1}\}. \tag{31}$$

Using the above recursion, the IPA estimator for the derivative of average system time is given by

$$\frac{\mathrm{d}\overline{T}_N}{\mathrm{d}\theta} = \frac{1}{N}\sum_{n=1}^{N}\frac{\mathrm{d}T_n}{\mathrm{d}\theta} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=n_{m-1}+1}^{n_m}\sum_{j=n_{m-1}+1}^{i}\frac{\mathrm{d}X_j}{\mathrm{d}\theta}, \tag{32}$$

where M is the number of busy periods observed and n_m is the index of the last customer served in the mth busy period ($n_0 = 0$ and $n_M = N$ for M complete busy periods). Implementation of the estimator involves keeping track of two running quantities, one for (31) and another for the summation in (32); thus, the additional computational overhead is minimal, and *no alteration of the underlying simulation is required*.

The implicit assumption used in deriving an IPA estimator is that small changes in the parameter will result in small changes in the sample performance. Thus, in the above, this means that the boundary condition in (31) is unchanged by differentiation. In general, the interchange (13) will typically hold if the sample performance is continuous with respect to the parameter. For the Lindley equation, although T_{n+1} in (30) has a "kink" at $T_n = A_{n+1}$, it is still continuous at that point. This intuitively explains why IPA works. Unfortunately, the "kink" means that the derivative given by (31) has a discontinuity at $T_n = A_{n+1}$, so that IPA will fail for the second derivative.

An unbiased SPA second derivative estimator can be derived by conditioning on all *previous* interarrival and service times at each departure, which determines the system time, say T_n, with the corresponding next interarrival time, A_{n+1}, unconditioned. We provide a brief informal derivation based on sample path intuition (refer to Figure 1). For the right-hand estimator, in which we assume $\Delta T_n > 0$ (technically it should refer to $\Delta\theta$), the only "critical" events are those departures that terminate a busy period, with the possibility that two busy periods coalesce (idle period disappears) due to a perturbation. The corresponding probability rate (conditional on T_n) is then calculated as follows:

$$\lim_{\Delta\theta\to 0}\frac{\mathrm{P}(T_n + \Delta T_n \geqslant A_{n+1} | T_n < A_{n+1})}{\Delta T_n} = \frac{f_1(T_n)}{1 - F_1(T_n)}\frac{\mathrm{d}T_n}{\mathrm{d}\theta},$$

and the corresponding effect would be that the ΔT_n perturbation would be propagated to the next busy period. The complete SPA estimator is given by

$$\left(\frac{\mathrm{d}^2\overline{T}_N}{\mathrm{d}\theta^2}\right)_{\mathrm{SPA}} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=n_{m-1}+1}^{n_m}\sum_{j=n_{m-1}+1}^{i}\frac{\mathrm{d}^2X_j}{\mathrm{d}\theta^2} + \frac{1}{M}\sum_{m=1}^{M}\frac{f_1(T_{n_m})}{1 - F_1(T_{n_m})}\left(\frac{\mathrm{d}T_{n_m}}{\mathrm{d}\theta}\right)^2,$$

where $\mathrm{d}^2X/\mathrm{d}\theta^2$ is well defined when $F_2(X;\theta)$ is twice differentiable, $\frac{\mathrm{d}^2X}{\mathrm{d}\theta^2} = 0$ for location, scale and generalized scale parameters.

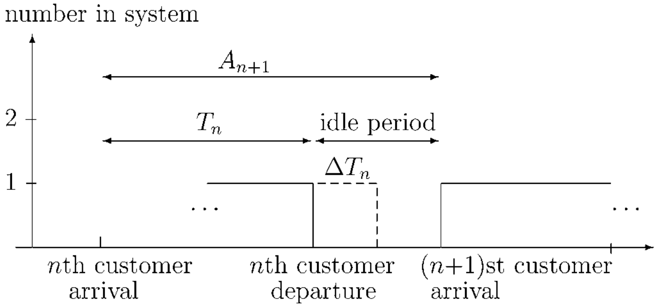


Fig. 1. Quantities used in deriving FCFS single-server queue SPA estimator.

To derive the LR/SF estimator, since all the interarrival and service times are independently generated, the joint p.d.f. f on X will simply be the product of the p.d.f.s of the interarrival and service time distributions given by

$$f(\theta, A_1, \dots, A_N, X_1, \dots, X_N) = \prod_{i=1}^{N} f_1(A_i) \prod_{i=1}^{N} f_2(X_i; \theta). \tag{33}$$

Thus, the straightforward estimator would be given by

$$\left(\frac{\mathrm{d}\overline{T}_N}{\mathrm{d}\theta}\right)_{\mathrm{LR}} = \frac{1}{N}\sum_{i=1}^{N} T_i \sum_{j=1}^{N} \frac{\partial \ln f_2(X_j; \theta)}{\partial \theta}, \tag{34}$$

where expressions for some common input distributions can be found in Table 1. However, for large N, the variance of the estimator, which increases linearly with N, will render it practically useless, so some sort of truncation is necessary, and in this particular example, the regenerative structure provides such a mechanism. Using regenerative theory (see Chapter 16) the mean steady-state system time can be written as a ratio of expectations:

$$\mathrm{E}[T] = \frac{\mathrm{E}[Q]}{\mathrm{E}[\eta]},$$

where η is the number of customers served in a busy period and Q is the sum of the system times of customers served in a busy period. Differentiation yields

$$\frac{\mathrm{dE}[T]}{\mathrm{d}\theta} = \frac{\mathrm{dE}[Q]/\mathrm{d}\theta}{\mathrm{E}[\eta]} - \frac{\mathrm{dE}[\eta]/\mathrm{d}\theta}{\mathrm{E}[\eta]}\mathrm{E}[T].$$

Now, use the natural LR/SF estimators for each of the terms separately to obtain the following regenerative estimator over M busy periods:

$$\begin{aligned}
&\left(\frac{\mathrm{d}\overline{T}_N}{\mathrm{d}\theta}\right)_{\mathrm{LR}} \\
&\quad= \frac{1}{N}\sum_{m=1}^{M}\left\{\sum_{i=n_{m-1}+1}^{n_m} T_i \sum_{i=n_{m-1}+1}^{n_m} \frac{\partial \ln f_2(X_i;\theta)}{\partial\theta}\right\} \\
&\qquad - \frac{1}{N}\sum_{m=1}^{M}\left\{(n_m - n_{m-1})\sum_{i=n_{m-1}+1}^{n_m} \frac{\partial \ln f_2(X_i;\theta)}{\partial\theta}\right\}\frac{1}{N}\sum_{j=1}^{N} T_j.
\end{aligned}$$

The advantage of these estimators is that the summations are bounded by the length of the busy periods, so as long as the busy periods are not too lengthy, the variance of the estimators should be acceptable. Furthermore, the second derivative estimator is relatively easy to derive, as well:

$$\begin{aligned}
&\left(\frac{\mathrm{d}^2\overline{T}_N}{\mathrm{d}\theta^2}\right)_{\mathrm{LR}} \\
&\quad= \frac{1}{N}\sum_{m=1}^{M}\left\{\sum_{i=1}^{n_m} T_i \sum_{i=n_{m-1}+1}^{n_m} \left[\frac{\partial^2 \ln f_2(X_i;\theta)}{\partial\theta^2} + \left(\frac{\partial \ln f_2(X_i;\theta)}{\partial\theta}\right)^2\right]\right\} \\
&\qquad - \frac{1}{N}\sum_{m=1}^{M}\left\{(n_m - n_{m-1})\sum_{i=n_{m-1}+1}^{n_m} \left[\frac{\partial^2 \ln f_2(X_i;\theta)}{\partial\theta^2}\right.\right. \\
&\qquad\qquad \left.\left. + \left(\frac{\partial \ln f_2(X_i;\theta)}{\partial\theta}\right)^2\right]\right\} \\
&\qquad \times \frac{1}{N}\sum_{j=1}^{N} T_j.
\end{aligned}$$

The WD estimator is also relatively straightforward, just incorporating the product rule of differentiation as before:

$$\begin{aligned}
\left(\frac{\mathrm{d}\overline{T}_N}{\mathrm{d}\theta}\right)_{\mathrm{WD}} &= c(\theta)\sum_{i=1}^{N}\bigl[\overline{T}_N\bigl(A_1,\dots,A_N,\dots,X_i^{(2)},\dots\bigr) \\
&\qquad - \overline{T}_N\bigl(A_1,\dots,A_N,\dots,X_i^{(1)},\dots\bigr)\bigr],
\end{aligned}$$

where $X_i^{(j)} \sim F_2^{(j)}$, $j = 1, 2$, $i = 1, \dots, N$, for $(c(\theta), F_2^{(1)}, F_2^{(2)})$ a weak derivative of F_2. A second derivative estimator would take exactly the same form, the only difference being that $(c(\theta), F_2^{(1)}, F_2^{(2)})$ should be a weak second derivative of F_2. Note that in general, implementation of the estimator requires $2T$ separate sample paths and resulting sample performance estimates, because the parameter appears in T input random variables. Although the variance of the estimator does not increase with T, implementation may not be practical for large T. However, in many cases, the expression can be simplified, making the computation more acceptable. The variance properties of the estimators depend heavily on the particular weak derivative(s) used.

What happens when T is random and when $T \to \infty$ is also of theoretical interest. In these settings, the estimators are extended in the natural way, but proving theoretical properties becomes a more challenging task.

5.3 (s, S) inventory system

We now consider the single-item periodic review (s, S) inventory system, in which once every period the inventory level is reviewed and, if necessary, orders are placed to replenish depleted inventory. An (s, S) ordering policy specifies that an order be placed when the level of inventory on hand plus that on order (called *inventory position*) falls below the level s, and that the amount of the order be the difference between S and the present inventory position, i.e., order amounts are placed "up to S". For expositional ease, we describe only the gradient estimate for average inventory level with respect to the policy parameters s and $q = S - s$, which do not enter through probability distributions as in the previous examples.

We consider the model where all excess demand is backlogged and eventually filled, and replenishment orders are immediately received (zero lead time), so that inventory level and inventory position coincide. We assume that during the period, demand is satisfied, and then the order replenishment decision is made at the end of the period. Let D_n be the demand in period n, which is assumed i.i.d. with respective density and distribution functions given by f and F, and let V_n be the inventory level in period n *after* demand satisfaction at the end of the period, but just *prior* to the order replenishment decision. This quantity satisfies a recursive equation somewhat analogous to the Lindley equation:

$$V_{n+1} = \begin{cases} V_n - D_{n+1} & \text{if } V_n \geqslant s, \\ S - D_{n+1} & \text{if } V_n < s. \end{cases} \tag{35}$$

The average inventory level over N periods is given by

$$\overline{V}_N = \frac{1}{N} \sum_{n=1}^{N} V_n.$$

From a sample path point of view, the key discrete event in the system is the ordering decision each period. A change in s, with q held fixed, has no effect on these decisions, so infinitesimal perturbations in s result in infinitesimal changes in the inventory level and in the sample performance function $\overline{V}_N$. In particular, for a perturbation of size Δs (of any size, not necessarily infinitesimal), $V_n(s + \Delta s) = V_n(s) + \Delta s$, and thus $\partial \overline{V}_N / \partial s = 1$ is an unbiased estimator for $\partial \mathrm{E}[\overline{V}_N]/\partial s$. Intuitively, the shape of the sample path is unaltered by changes in s if q is held constant; the entire sample path is merely shifted by the size of the change (assuming starting inventory of $V_0 = S = s + q$; else, the statement only holds starting from the first order point). The IPA estimator can also be obtained by simply differentiating the recursive relationship (35), noting that D_n does not depend on s or q:

$$\frac{\mathrm{d}V_{n+1}}{\mathrm{d}\theta} = \begin{cases} \frac{\mathrm{d}V_n}{\mathrm{d}\theta} & \text{if } V_n \geqslant s, \\ 1 & \text{if } V_n < s, \end{cases}$$

for either $\theta = s$ or $\theta = q$. Under the assumption that $V_0 = S = s + q$, the expression reduces to 1 for all n when $\theta = s$.

On the other hand, a change in q with s held fixed may cause a change in the set of ordering decisions, resulting in a drastic change in the sample path and hence in the performance measure $\overline{V}_N$. Thus, SPA is required to derive an unbiased gradient estimator for $\theta = q$. We provide a brief informal derivation for a right-hand SPA estimator, i.e., $\Delta q > 0$, in which a period where a replenishment order was originally placed could become one where an order is not placed, for sufficiently large Δq (refer to Figure 2). To calculate the probability rate for such an event requires knowing the quantity above the reorder point s prior to demand being realized in the period, given by $\xi_n = Y_n - s$ in Figure 2, where Y_n is the inventory position/level prior to demand being realized. Conditioning on all demands except the one in the period of interest (in which an order is placed), D_n, the corresponding probability rate is then given by

$$\lim_{\Delta q \to 0} \frac{\mathrm{P}(\xi_n + \Delta q \geqslant D_n | \xi_n < D_n)}{\Delta q} = \frac{f(\xi_n)}{1 - F(\xi_n)}.$$

The resulting conditional expected effect on the performance measure requires some further analysis, and is omitted here, and we simply give the final SPA estimator for $\partial \mathrm{E}[\overline{V}_N]/\partial q$, which can be easily and efficiently estimated

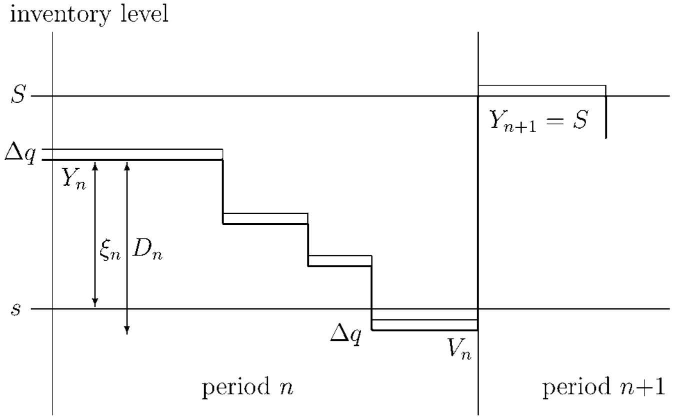


Fig. 2. Quantities used in (s, S) inventory system.

from the original sample path:

$$1+\frac{1}{N}\sum_{n\leqslant N:V_n<s}\frac{f(\xi_n)}{1-F(\xi_n)}\left[s-\mathrm{E}[D]-\overline{V}_N\right],$$

where E[D] is mean demand.

The LR/SF and WD methods require a change of variables in order to move the parameters into the distribution. This requires some nontrivial analysis, beyond the scope of this chapter.

6 Basic theoretical tools

The key result used in the theoretical proofs of unbiasedness is the (Lebesgue) dominated convergence theorem that allows for the exchange of limit and expectation operators required in Equations (12) or (13).

Theorem 1 (Dominated convergence theorem). *If* $\lim_{n\to\infty} g_n = g$ *a.s. and* $|g_n| \leqslant M$ $\forall n$ *a.s. with* $\mathrm{E}[M] < \infty$, *then* $\lim_{n\to\infty}\mathrm{E}[g_n] = \mathrm{E}[g]$.

Take $\Delta\theta \to 0$ instead of $n \to \infty$, and g is the gradient estimator, so $g_{\Delta\theta}$ is the limiting sequence that defines the sample (path) gradient. Verifying that an actual bound exists is often a nontrivial task in applications, especially in the case of perturbation analysis.

Looking at (12), we translate these conditions to

$$g_{\Delta\theta} = \frac{Y(\theta+\Delta\theta)-Y(\theta)}{\Delta\theta},$$
$$g_{\Delta\theta} = Y(x)\frac{f(x;\theta+\Delta\theta)-f(x;\theta)}{\Delta\theta},$$

for IPA and LR/SF, respectively.

For IPA, the following generalization of the mean value theorem is most useful (cf. Theorem 6 in Chapter 2).

Theorem 2 (Generalized mean value theorem). *Let Y be a continuous function that is differentiable on a compact set $\widetilde{\Theta} = \Theta \setminus \widetilde{D}$, where $\widetilde{D}$ is a set of countably many points. Then $\forall\theta$, $\theta+\Delta\theta \in \Theta$,*

$$\left|\frac{Y(\theta+\Delta\theta)-Y(\theta)}{\Delta\theta}\right| \leqslant \sup_{\theta\in\widetilde{\Theta}}\left|\frac{\mathrm{d}Y}{\mathrm{d}\theta}\right|.$$

If $Y(\theta)$ can be shown to be continuous and piecewise differentiable on Θ w.p.1, then it just remains to show

$$\mathrm{E}\left[\sup_{\theta\in\widetilde{\Theta}}\left|\frac{\mathrm{d}Y}{\mathrm{d}\theta}\right|\right] < \infty,$$

to satisfy the conditions required for unbiasedness via the dominated convergence theorem. Basically, in order for the chain rule to be applicable, the sample performance function needs to be continuous with respect to the underlying random variable(s). This translates into requirements on the form of the performance measure and on the dynamics of the underlying stochastic system.

For the LR/SF method, the bound is applied to the (joint) density (or mass) function. Note that the bound is for $f(x;\theta)$ with respect to the parameter θ and not its usual argument. For WD, the required interchange is guaranteed by the definition of the weak derivative, but the sample performance must be in the set of "test" functions $\mathcal{L}$ in the definition, which again generally requires the dominated convergence theorem (or uniform integrability, which is usually difficult to check directly).

The previous examples can be used to show in very simple cases where difficulties arise.

Consider the p.d.f.

$$f(x;\theta) = \frac{1}{\theta}\mathbf{1}\{0 < x < \theta\},$$

where the LR/SF method does not apply. In this case, f viewed as a function of θ for fixed x has a discontinuity at $\theta = x$.

Consider the function

$$\mathrm{P}(Y > y) = \mathrm{E}\big[\mathbf{1}\{Y > y\}\big],$$

in which IPA will not work. In this case, the performance measure is an indicator function, which is discontinuous in its argument.

Thus, in both simple examples, the dominated convergence theorem is not applicable as the required quantity cannot be bounded. However, it is only a sufficient condition, not necessary, so in some (very) special cases, unbiasedness may hold even without the boundedness (continuity).

7 Simple guidelines for the simulation practitioner

We summarize the most important considerations in applying the three direct gradient procedures (PA, LR/SF, WD):

- IPA – cannot handle "bad" performance measures, e.g., indicator functions, or nonsmooth systems; one way of checking the latter is the commuting condition, which checks event sequences in the system (cf. Glasserman, 1991); smoothness may depend on system representation (see discussion on role of representations);
- SPA – choice of what to condition on, and how to compute (estimate) resulting conditional expectation; may require many additional simulations;

- LR/SF and WD – may encounter difficulties in handling parameters not in distribution (so-called "structural" parameters); may need to try a change of variables;
- LR/SF – if the parameter appears in an input distribution that is reused frequently (parameter makes too many repeat appearances), e.g., i.i.d. service times in a queueing system, then may need to find a way to truncate the estimator to mitigate the linear increase in variance;
- WD – choice of which (nonunique) WD representation to use; also high-dimensional vectors may require many simulations;
- For discrete distributions, IPA works if the parameter occurs in the support *values*, whereas LR/SF and WD work if the parameter occurs in the support *probabilities*;
- Higher derivative estimates are usually easiest to derive using LR/SF, but even then the variance of the resulting estimators may be problematic.

The characteristics/choices in applying SPA are strikingly similar to the use of conditional Monte Carlo for variance reduction. The choice of what to condition on in applying SPA also has analogs to the WD representation choice. The simplest procedures to use are the LR/SF and IPA estimators. The WD estimator may be easier to apply than SPA, because one has a set of "standard" choices for a large class of distributions. However, there is no guarantee that these choices are necessarily good for a particular application.

It should be clear from this brief summary that the direct gradient estimation techniques may require some effort on the part of the simulation user, whereas the indirect techniques are straightforward to apply. To put it another way, for direct gradient estimation sometimes it takes some art to do the science.

8 Applications

In discussing applications, the focus is on the direct gradient estimators, since application of the indirect gradient estimates is generally more straightforward and domain independent. The most dominant application of direct gradient estimation in the research literature is clearly queueing systems; however, inventory management and financial engineering have employed many of these results in real-world practice.

For derivatives with respect to distributional parameters, IPA can be applied in any single-class Jackson-like network. Here, "Jackson-like" (also called a "generalized Jackson network") means that the network retains all of the usual characteristics of a Jackson network except that service times and interarrival times (in an open network) do not have to be exponentially distributed. This is also true for the LR/SF method and the WD method. In the latter case, there are often choices of the WD used. However, if the parameter is the routing probabilities, then IPA cannot be applied directly. Also, if the network is extended to multiple classes of customers, then IPA is often not applicable.

It is relatively straightforward to apply the LR/SF method to queueing systems, the only caveat being the potential variance problem mentioned in the previous section. It is also relatively straightforward to derive weak derivative estimators, but there are generally many choices of distributions, and often one or more additional simulations using different input distributions will be necessary.

Inventory systems is a domain in which direct gradient estimation has been successfully applied in real-world applications. Because the parameters of interest are usually those controlling replenishment decisions as in the (s, S) inventory control example of Section 5, PA is most relevant. For so-called base–stock policies, often IPA suffices. Such IPA estimators are being used to optimize inventory management in the worldwide supply chain of Caterpillar, and the success of the approach is reported in a *Fortune* magazine article, "New victories in the supply chain revolution" (by Philip Siekman, October 30, 2000); technical details of the approach can be found in Kapuscinski and Tayur (1999). For more complicated ordering policies such as an (s, S) policy that comes about with the inclusion of a fixed order cost, extensions of IPA are required; see Fu and Healy (1992, 1997), Fu (1994b), Bashyam and Fu (1994), Fu and Hu (1994), Zhang and Fu (2005), Pflug and Rubinstein (2002) for an LR/SF estimator using the push-out method with conditioning, and especially Chapter 7 of Fu and Hu (1997).

Because on Wall Street and elsewhere in the global financial world, Monte Carlo simulation is routinely used for pricing and hedging derivatives, it is now commonly included in any finance text that addresses numerical solution methods. Simulation can easily handle the pricing of high-dimensional derivatives, such as path-dependent claims or systems with a large number of underlying assets and/or uncertainties (e.g., stochastic volatility and interest rate models); see also the discussion in Chapter 1. In hedging, gradients are the most critical ingredients in determining what positions need to be taken in any portfolio. In equities, they are usually referred to as "Greeks", because they are represented by Greek letters. For example, the most common one is the Delta, which is the sensitivity of a derivative price with respect to the underlying security price, e.g., the price of a call option with respect to the underlying stock price on which the contract is written. Delta hedging and other types of hedging are described in most elementary finance textbooks on derivatives such as Hull (2005), and the text by Clewlow and Strickland (1999) includes simple IPA estimators for calculating Greeks using Monte Carlo simulation (though they do not use the term IPA nor stochastic gradient estimation). Fu and Hu (1995) and Broadie and Glasserman (1996) were the first to develop stochastic gradients in these settings; see also Glasserman (2004). Heidergott (2001b) considered weak derivatives in a similar setting as Fu and Hu (1995). In fixed income securities, the analogous quantities go by the name of duration and convexity. In Chen and Fu (2002), IPA is applied to the pricing and hedging of mortgage-backed securities, where both first and second derivatives are required. For just five parameters, if symmetric differences were used, this would require 236 (1+10+225) simulations for each set of performance measures and gradient estimates. In actual implementation, there was a resulting dramatic 97.2% reduction in computation time. Another important finance application is the pricing of American-style derivatives: contingent claims in which early exercise is permitted. One of the earliest proposed approaches to this problem was to parameterize the early exercise boundary, thus casting the optimal stopping problem as a stochastic optimization problem with respect to the parameters. The earliest example of applying PA and SA to such an option pricing problem is given in Fu and Hu (1995). Earlier editions of Hull (2005) and other finance texts claimed that simulation could be used only to price European options, so this is one of many approaches that dispelled that belief. See Glasserman (2004) for other approaches and references on this topic.

Other applications include stochastic activity networks (see Rubinstein and Shapiro (1993) for the LR/SF method, Bowman (1994) for IPA, and Fu (2006) for SPA and WD); preventive maintenance (Fu et al., 1993; Heidergott 1999, 2001a; L'Ecuyer et al., 1999); statistical process control (Fu and Hu, 1999); traffic light signal control (Howell and Fu, 2003; also mentioned in Rubinstein and Shapiro, 1993, p. 3, as an example).

9 Probing further

Disclaimer: The list here is meant to be representative, not comprehensive, and almost no attempt is made to provide a historical context to the results in this chapter.

Other approaches not treated in this chapter include frequency domain experimentation (Schruben and Cogliano, 1981; Schruben, 1986; Jacobson et al., 1991; Jacobson, 1994); and Malliavin calculus, which has been used primarily in financial applications (e.g., Fournié et al., 1999; Benhamou, 2002, and references therein).

For gradient-based simulation optimization, further discussion can be found in the papers of Fu (2002), Andradóttir (1998), Fu (1994a) and Jacobson and Schruben (1989); see also the books by Rubinstein and Shapiro (1993), Pflug (1996), Fu and Hu (1997) and Spall (2003), as well as Fu (2001b). Both of the well-known simulation textbooks by Law and Kelton (2000) and Banks et al. (2000) also devote sections to the topic, but the latter does not discuss gradient estimation, per se. Applications to the single-server queue example considered here include the theoretical convergence results of Fu (1990), Chong and Ramadge (1992) and L'Ecuyer and Glynn (1994), and the in-depth numerical comparisons of L'Ecuyer et al. (1994) and Andradóttir (1998). Andradóttir (1996) considers the more general setting of using the LR/SF estimators in SA algorithms, and Tang et al. (1999) analyze the asymptotic efficiency of an averaging version of SA using perturbation analysis estimators. Work on sample path optimization (called the stochastic counterpart method in Rubinstein and Shapiro, 1993) includes Plambeck et al. (1996), Robinson (1996), Gürkan et al. (1999), Homem-de-Mello et al. (1999); Dussault et al. (1997) combines the approach with stochastic approximation. See also the chapter by Shapiro (2003) on using Monte Carlo methods for stochastic programming. The original papers on stochastic approximation are Robbins and Monro (1951) and Kiefer and Wolfowitz (1952). For a more recent general book on stochastic approximation, see Kushner and Yin (1997); L'Ecuyer and Yin (1998) discusses convergence rates as a function of computational budget. Spall (2000) provides both indirect and direct gradient-based SA methods for obtaining near-optimal or optimal convergence rates via stochastic analogues to the deterministic Newton–Raphson algorithm, using Hessian matrix estimation. SPSA was introduced by Spall (1992), although the random directions method was proposed in Kushner and Clark (1978); see *http://www.jhuapl.edu/SPSA/* for an extensive annotated bibliography. Application to the settings of this handbook is described in Fu and Hill (1997), and using deterministic sequences for SPSA is considered in Bhatnagar et al. (2003) and Xiong et al. (2002). Andradóttir (1998) contains further useful discussion on the application of SA algorithms in simulation optimization, especially regarding the issues of choosing the gain sequence and modifying the algorithm in cases where the objective function is not well behaved. Most stochastic approximation algorithms involve known constraints; Bashyam and Fu (1998) consider the case of a noisy constraint that must also be estimated, in particular a service level constraint for an (s, S) inventory system. Studies on choosing the finite difference used in the Kiefer–Wolfowitz version of SA include Zazanis and Suri (1993) and L'Ecuyer and Perron (1994).

For perturbation analysis, the books by Glasserman (1991), Ho and Cao (1991), and Cao (1994) treat IPA extensively (in particular, see Glasserman (1991) for a complete treatment of the commuting condition; see also Cao (1985) and Heidelberger et al. (1988)) and SPA to some extent, whereas the book by Fu and Hu (1997) is a comprehensive treatment of SPA, which was introduced by Gong and Ho (1987) and Suri and Zazanis (1988); see also Fu and Hu (1992), Fu (2001a) and Fu and Hu (1991, 1993), where higher derivatives for multi-server queues are treated. Many of the examples in this chapter are adopted from Fu and Hu (1997). More recent work connecting IPA with Markov decision processes using the idea of potentials, which also relates to the LR/SF method, include Cao (2000, 2003a, 2003b). The field of perturbation analysis for gradient estimation includes numerous other extensions and variations on IPA not discussed here, including rare perturbation analysis (Brémaud and Vázquez-Abad, 1992); structural IPA (Dai and Ho, 1995); discontinuous perturbation analysis (Shi, 1996); and augmented IPA (Gaivoronski et al., 1992). Seminal papers applying perturbation analysis for estimating the effects of *finite* perturbations in the parameter include Ho and Li (1988), Cassandras and Strickland (1989) and Vakili (1991).

For the likelihood ratio or score function method, a good reference is the book by Rubinstein and Shapiro (1993), which also includes some discussion of IPA (Chapter 5 on the "push in" method), as well as both first and second derivative estimators for most of the entries in Table 1 (Section 2.2); see also Reiman and Weiss (1989), Rubinstein (1989), Glynn (1990) and Andradóttir (1996). The "push out" method for handling structural parameters was introduced in Rubinstein (1992); see also Section 2.5.4 in Rubinstein and Shapiro (1993). Using conditional expectation to reduce variance in the LR/SF method was considered in McLeish and Rollins (1992); see also Section 3.4 in Rubinstein and Shapiro (1993). A "unified" view of IPA and LR/SF is presented in L'Ecuyer (1990), which allows the parameter to appear in both the sample performance and the distribution (see also L'Ecuyer (1995) for further discussion and some technical corrections).

The weak derivative method was introduced by Pflug (1989, 1990, 1996). More recent work attempting to unify the approach with others, such as rare perturbation analysis and smooth perturbation analysis, includes Heidergott and Vazquez-Abad (20006a, 2006b), in the setting of general Markov processes, which may not provide as convenient a framework for the discrete-event simulation setting as generalized semi-Markov processes (GSMPs). Derivations for many of the entries in Table 1 can be found in Heidergott et al. (2003).

10 Future research directions

Stochastic gradient estimation research has matured over the last decade to the point that much of the analysis has become theoretical rather than algorithmic. This line of research addresses many issues that also arise in traditional steady-state output analysis – only now for stochastic gradient estimators – and can involve advanced probabilistic tools. In contrast, two possible directions for future research described briefly here are more motivated by simulation practice and have much algorithmic room left in them to grow.

How is gradient estimation best employed in simulation optimization? One recent development is the use of fluid models in this setting. A discrete-event simulation model for which application of the direct derivative estimation may be difficult is approximated by a stochastic fluid model, which is used to derive IPA estimators that are then implemented on either the (simulated) stochastic fluid model or sometimes on the original stochastic discrete-event simulation model. In the former case, where the IPA estimates are generally unbiased, the optimal settings often provide a good approximation for the optimum in the original model. In the latter case, the new IPA estimators do not yield unbiased estimators when implemented in the discrete-event model (so are generally not useful in sensitivity analysis, per se), but they often provide a good approximation of the zero gradient location when used in optimization; however, the theoretical basis for this good fortune is still not fully understood.

Some papers in this area include Wardi et al. (2002), Cassandras et al. (2002), Sun et al. (2004) and Panayiotou et al. (2005).

The LR/SF method is closely related to importance sampling (cf. Chapter 11). Investigating this connection more thoroughly for variance reduction purposes, and doing so in a more general stochastic gradient setting, would be of benefit to simulationists. Some work along this line in the setting of stochastic programming is contained in Shapiro and Homem-de-Mello (1998).

Acknowledgements

This work was supported in part by the National Science Foundation under Grants DMI 9988867 and DMI 0323220, and by the Air Force Office of Scientific Research under Grants F496200110161 and FA95500410210. The author thanks the co-Editors for their detailed comments and suggestions that have led to an improved exposition, and Jeff Heath, Jiaqiao Hu, Ying He, Scott Nestler, Chang Su, and Xiaoping Xiong for their reading of various drafts along the way.

References

Andradóttir, S. (1996). Optimization of the transient and steady-state behavior of discrete event systems. *Management Science* 42, 717–737.

Andradóttir, S. (1998). Simulation optimization. In: Banks, J. (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Wiley, New York, pp. 307–333, Chapter 9.

Banks, J., Carson, J.S., Nelson, B.L., Nicol, D.M. (2000). *Discrete Event Systems Simulation*, 3rd edition. Prentice Hall, Englewood Cliffs, NJ.

Bashyam, S., Fu, M.C. (1994). Application of perturbation analysis to a class of periodic review (s, S) inventory systems. *Naval Research Logistics* 41, 47–80.

Bashyam, S., Fu, M.C. (1998). Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science* 44, S243–S256.

Benhamou, E. (2002). Smart Monte Carlo: Various tricks using Malliavin calculus. *Quantitative Finance* 2, 329–336.

Bhatnagar, S., Fu, M.C., Marcus, S.I., Wang, I.J. (2003). Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. *ACM Transactions on Modeling and Computer Simulation* 13, 180–209.

Bowman, R.A. (1994). Stochastic gradient-based time-cost tradeoffs in PERT network using simulation. *Annals of Operations Research* 53, 533–551.

Brémaud, P., Vázquez-Abad, F.J. (1992). On the pathwise computation of derivatives with respect to the rate of a point process: The phantom RPA method. *Queueing Systems: Theory and Applications* 10, 249–270.

Broadie, M., Glasserman, P. (1996). Estimating security price derivatives using simulation. *Management Science* 42, 269–285.

Cao, X.R. (1985). Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Transactions on Automatic Control* 30, 834–843.

Cao, X.R. (1994). *Realization Probabilities: The Dynamics of Queuing Systems*. Springer-Verlag, New York.

Cao, X.R. (2000). A unified approach to Markov decision problems and performance sensitivity analysis. *Automatica* 36, 771–774.

Cao, X.R. (2003a). From perturbation analysis to Markov decision processes and reinforcement learning. *Discrete Event Dynamic Systems: Theory and Applications* 13, 9–39.

Cao, X.R. (2003b). Semi-Markov decision problems and performance sensitivity analysis. *IEEE Transactions on Automatic Control* 48, 758–769.

Cassandras, C.G., Strickland, S.G. (1989). On-line sensitivity analysis of Markov chains. *IEEE Transactions on Automatic Control* 34, 76–86.

Cassandras, C.G., Wardi, Y., Melamed, B., Sun, G., Panayiotou, C.G. (2002). Perturbation analysis for on-line control and optimization of stochastic fluid models. *IEEE Transactions on Automatic Control* 47, 1234–1248.

Chen, J., Fu, M.C. (2002). Hedging beyond duration and convexity. In: *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1593–1599.

Chong, E.K.P., Ramadge, P.J. (1992). Convergence of recursive optimization algorithms using infinitesimal perturbation analysis. *Discrete Event Dynamic Systems: Theory and Applications* 1, 339–372.

Clewlow, L.J., Strickland, C.R. (1999). *Implementing Derivative Models*. Wiley, New York.

Dai, L., Ho, Y.C. (1995). Structural infinitesimal perturbation analysis for derivative estimation in discrete event dynamic systems. *IEEE Transactions on Automatic Control* 40, 1154–1166.

Dussault, J.P., Labrecque, D., L'Ecuyer, P., Rubinstein, R.Y. (1997). Combining the stochastic counterpart and stochastic approximation methods. *Discrete Event Dynamic Systems: Theory and Applications* 7, 5–28.

Fournié, E., Lasry, J.M., Lebuchoux, J., Lions, P.L., Touzi, N. (1999). Applications of Malliavin calculus to Monte Carlo methods in finance. *Finance and Stochastics* 3, 391–412.

Fu, M.C. (1990). Convergence of a stochastic approximation algorithm for the $GI/G/1$ queue using infinitesimal perturbation analysis. *Journal of Optimization Theory and Applications* 65, 149–160.

Fu, M.C. (1994a). Optimization via simulation: A review. *Annals of Operations Research* 53, 199–248.

Fu, M.C. (1994b). Sample path derivatives for (s, S) inventory systems. *Operations Research* 42, 351–364.

Fu, M.C. (2001a). Perturbation analysis. In: Gass, S.I., Harris, C.M. (Eds.), *Encyclopedia of Operations Research and Management Science*, 2nd edition. Kluwer Academic Publishers, Boston, MA, pp. 608–611.

Fu, M.C. (2001b). Simulation optimization. In: Gass, S.I., Harris, C.M. (Eds.), *Encyclopedia of Operations Research and Management Science*, 2nd edition. Kluwer Academic Publishers, Boston, MA, pp. 756–759.

Fu, M.C. (2002). Optimization for simulation: Theory vs. practice (Feature Article). *INFORMS Journal on Computing* 14, 192–215.

Fu, M.C. (2006). Sensitivity analysis for stochastic activity networks. In: Alt, F.B., Fu, M.C., Golden, B.L. (Eds.), *Topics in Modeling, Optimization, and Decision Technologies: Honoring Saul Gass' Contributions to Operations Research*. Kluwer Academic Publishers, Boston, MA, in press.

Fu, M.C., Healy, K.J. (1992). Simulation optimization of (s, S) inventory systems. In: *Proceedings of the Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 506–514.

Fu, M.C., Healy, K.J. (1997). Techniques for simulation optimization: An experimental study on an (s, S) inventory system. *IIE Transactions* 29, 191–199.

Fu, M.C., Hill, S.D. (1997). Optimization of discrete event systems via simultaneous perturbation stochastic approximation. *IIE Transactions* 29, 233–243.

Fu, M.C., Hu, J.Q. (1991). On choosing the characterization for smoothed perturbation analysis. *IEEE Transactions on Automatic Control* 36, 1331–1336.

Fu, M.C., Hu, J.Q. (1992). Extensions and generalizations of smoothed perturbation analysis in a generalized semi-Markov process framework. *IEEE Transactions on Automatic Control* 37, 1483–1500.

Fu, M.C., Hu, J.Q. (1993). Second derivative sample path estimators for the $GI/G/m$ queue. *Management Science* 39, 359–383.

Fu, M.C., Hu, J.Q. (1994). (s, S) inventory systems with random lead times: Harris recurrence and its implications in sensitivity analysis. *Probability in the Engineering and Informational Sciences* 8, 355–376.

Fu, M.C., Hu, J.Q. (1995). Sensitivity analysis for Monte Carlo simulation of option pricing. *Probability in the Engineering and Information Sciences* 9, 417–446.

Fu, M.C., Hu, J.Q. (1997). *Conditional Monte Carlo: Gradient Estimation and Optimization Applications*. Kluwer Academic Publishers, Boston, MA.

Fu, M.C., Hu, J.Q. (1999). Efficient design and sensitivity analysis of control charts using Monte Carlo simulation. *Management Science* 45, 395–413.

Fu, M.C., Hu, J.Q., Shi, L. (1993). An application of perturbation analysis to replacement problems in maintenance. In: *Proceedings of the 1993 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 329–337.

Gaivoronski, A., Shi, L., Sreenivas, R.S. (1992). Augmented infinitesimal perturbation analysis: An alternate explanation. *Discrete Event Dynamic Systems: Theory and Applications* 2, 121–138.

Glasserman, P. (1991). *Gradient Estimation via Perturbation Analysis*. Kluwer Academic Publishers, Boston, MA.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

Glynn, P.W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communicatios of the ACM* 33, 75–84.

Gong, W.B., Ho, Y.C. (1987). Smoothed perturbation analysis of discrete-event dynamic systems. *IEEE Transactions on Automatic Control* 32, 858–867.

Gürkan, G., Özge, A.Y., Robinson, S.M. (1999). Sample-path solution of stochastic variational inequalities. *Mathematical Programming* 84, 313–333.

Heidelberger, P., Cao, X.R., Zazanis, M., Suri, R. (1988). Convergence properties of infinitesimal perturbation analysis estimates. *Management Science* 34, 1281–1302.

Heidergott, B. (1999). Optimization of a single-component maintenance system: A smoothed perturbation analysis approach. *European Journal of Operational Research* 119, 181–190.

Heidergott, B. (2001a). A weak derivative approach to optimization of threshold parameters in a multicomponent maintenance system. *Journal of Applied Probability* 38, 386–406.

Heidergott, B. (2001b). Option pricing via Monte Carlo simulation: A weak derivative approach. *Probability in Engineering and Informational Sciences* 15, 335–349.

Heidergott, B., Vázquez-Abad, F. (2006a). Measure-valued differentiation for random horizon problems. *Markov Processes and Related Fields*, in press.

Heidergott, B., Vázquez-Abad, F. (2006b). Measure-valued differentiation for Markov chains. *Journal of Optimization Theory and Applications*, in press.

Heidergott, B., Pflug, G., Vázquez-Abad, F. (2003). Measure-valued differentiation for stochastic systems: From simple distributions to Markov chains. Manuscript. Available at *http://staff.feweb.vu.nl/bheidergott/*.

Ho, Y.C., Cao, X.R. (1991). *Discrete Event Dynamics Systems and Perturbation Analysis*. Kluwer Academic Publishers, Boston, MA.

Ho, Y.C., Li, S. (1988). Extensions of infinitesimal perturbation analysis. *IEEE Transactions on Automatic Control* 33, 827–838.

Homem-de-Mello, T., Shapiro, A., Spearman, M.L. (1999). Finding optimal material release times using simulation based optimization. *Management Science* 45, 86–102.

Howell, W.C., Fu, M.C. (2003). Application of perturbation analysis to traffic light signal timing. In: *Proceedings of the 42nd IEEE Conference on Decision and Control*. IEEE Press, Piscataway, NJ, pp. 4837–4840.

Hull, J.C. (2005). *Options, Futures, and Other Derivative Securities*, 6th edition. Prentice Hall, Englewood Cliffs, NJ.

Jacobson, S.H. (1994). Convergence results for harmonic gradient estimators. *ORSA Journal on Computing* 6, 381–397.

Jacobson, S.H., Schruben, L.W. (1989). A review of techniques for simulation optimization. *Operations Research Letters* 8, 1–9.

Jacobson, S.H., Buss, A., Schruben, L.W. (1991). Driving frequency selection for frequency domain simulation experiments. *Operations Research* 39, 917–924.

Kapuscinski, R., Tayur, S.R. (1999). Optimal policies and simulation based optimization for capacitated production inventory systems. In: Tayur, S.R., Ganeshan, R., Magazine, M.J. (Eds.), *Quantitative Models for Supply Chain Management*. Kluwer Academic Publishers, Boston, MA. Chapter 2.

Kiefer, J., Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics* 23, 462–466.

Kushner, H.J., Clark, D.S. (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York.

Kushner, H.J., Yin, G.G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York.

Law, A.M., Kelton, W.D. (2000). *Simulation Modeling and Analysis*, 3rd edition. McGraw-Hill, New York.

L'Ecuyer, P. (1990). A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science* 36, 1364–1383.

L'Ecuyer, P. (1995). On the interchange of derivative and expectation for likelihood ratio derivative estimators. *Management Science* 41, 738–748.

L'Ecuyer, P., Glynn, P.W. (1994). Stochastic optimization by simulation: Convergence proofs for the $GI/G/1$ queue in steady-state. *Management Science* 40, 1562–1578.

L'Ecuyer, P., Perron, G. (1994). On the convergence rates of IPA and FDC derivative estimators. *Operations Research* 42, 643–656.

L'Ecuyer, P., Yin, G. (1998). Budget-dependent convergence rate of stochastic approximation. *SIAM Journal on Optimization* 8, 217–247.

L'Ecuyer, P., Giroux, N., Glynn, P.W. (1994). Stochastic optimization by simulation: Numerical experiments with a simple queue in steady-state. *Management Science* 40, 1245–1261.

L'Ecuyer, P., Martin, B., Vázquez-Abad, F. (1999). Functional estimation for a multicomponent age-replacement model. *American Journal of Mathematical and Management Sciences* 19, 135–156.

McLeish, D.L., Rollins, S. (1992). Conditioning for variance reduction in estimating the sensitivity of simulations. *Annals of Operations Research* 39, 157–173.

Panayiotou, C.G., Howell, W.C., Fu, M.C. (2005). Online traffic light control through gradient estimation using stochastic fluid models. *Proceedings of the IFAC 16th Triennial World Congress*. CD-ROM.

Pflug, G.C. (1989). Sampling derivatives of probabilities. *Computing* 42, 315–328.

Pflug, G.C. (1990). On-line optimization of simulated Markovian processes. *Mathematics of Operations Research* 15, 381–395.

Pflug, G.C. (1996). *Optimization of Stochastic Models*. Kluwer Academic Publishers, Boston, MA.

Pflug, G.C., Rubinstein, R.Y. (2002). Inventory processes: Quasi-regenerative property, performance evaluation and sensitivity estimation via simulation. *Stochastic Models* 18, 469–496.

Plambeck, E.L., Fu, B.-R., Robinson, S.M., Suri, R. (1996). Sample-path optimization of convex stochastic performance functions. *Mathematical Programming* 75, 137–176.

Robinson, S.M. (1996). Analysis of sample-path optimization. *Mathematics of Operations Research* 21, 513–528.

Reiman, M.I., Weiss, A. (1989). Sensitivity analysis for simulations via likelihood ratios. *Operations Research* 37, 830–844.

Robbins, H., Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics* 22, 400–407.

Rubinstein, R.Y. (1989). Sensitivity analysis of computer simulation models via the score efficient. *Operations Research* 37, 72–81.

Rubinstein, R.Y. (1992). Sensitivity analysis of discrete event systems by the 'push out' method. *Annals of Operations Research* 39, 229–251.

Rubinstein, R.Y., Shapiro, A. (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, New York.

Schruben, L.W. (1986). Simulation optimization using frequency domain methods. In: *Proceedings of the Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 366–369.

Schruben, L.W., Cogliano, V.J. (1981). Simulation sensitivity analysis: A frequency domain approach. In: *Proceedings of the Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 455–459.

Shapiro, A. (2003). Monte Carlo sampling methods. In: Ruszczynski, A., Shapiro, A. (Eds.), *Stochastic Programming. Handbook in Operations Research and Management Science*. Elsevier, New York. Chapter 6.

Shapiro, A., Homem-de-Mello, T. (1998). A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* 81, 301–325.

Shi, L.Y. (1996). Discontinuous perturbation analysis of discrete event dynamic systems. *IEEE Transactions on Automatic Control* 41, 1676–1681.

Spall, J.C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control* 37, 332–341.

Spall, J.C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control* 45, 1839–1853.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Hoboken, NJ.

Sun, G., Cassandras, C.G., Wardi, Y., Panayiotou, C.G., Riley, G.F. (2004). Perturbation analysis and optimization of stochastic flow networks. *IEEE Transactions on Automatic Control* 49, 2143–2159.

Suri, R., Zazanis, M.A. (1988). Perturbation analysis gives strongly consistent sensitivity estimates for the $M/G/1$ queue. *Management Science* 34, 39–64.

Tang, Q.-Y., L'Ecuyer, P., Chen, H.-F. (1999). Asymptotic efficiency of perturbation analysis-based stochastic approximation with averaging. *SIAM Journal on Control and Optimization* 37, 1822–1847.

Vakili, P. (1991). Using a standard clock technique for efficient simulation. *Operations Research Letters* 10, 445–452.

Wardi, Y., Melamed, B., Cassandras, C.G., Panayiotou, C.G. (2002). IPA gradient estimators in single-node stochastic fluid models. *Journal of Optimization Theory and Applications* 115, 369–406.

Xiong, X., Wang, I.J., Fu, M.C. (2002). Randomized-direction stochastic approximation algorithms using deterministic perturbation sequences. In: *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 285–291.

Zazanis, M.A., Suri, R. (1993). Convergence rates of finite-difference sensitivity estimates for stochastic systems. *Operations Research* 41, 694–703.

Zhang, H., Fu, M.C. (2005). Sample path derivatives for (s, S) inventory systems with price determination. In: Golden, B.L., Raghavan, S., Wasil, E.A. (Eds.), *The Next Wave in Computing, Optimization, and Decision Technologies*. Kluwer Academic Publishers, Boston, MA, pp. 229–246.

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13020-0

Chapter 20

An Overview of Simulation Optimization via Random Search

Sigrún Andradóttir

School of Industrial and Systems Engineering, Georgia Institute of Technology, USA
E-mail: sa@gatech.edu

Abstract

This chapter provides an overview of the use of random search methods for optimizing system performance via simulation. More specifically, we consider a broad class of optimization algorithms that sample a set of feasible system designs in each iteration, conduct simulations at the sampled designs in an effort to estimate the performance of these designs, and then use the simulation results to decide on what designs should be sampled in the next iteration and on the current estimate of the optimal system design. Consequently, the class of optimization algorithms under consideration is broad enough to include simulated annealing, tabu search, and genetic algorithms as special cases. We provide a discussion of when random search methods are guaranteed to converge almost surely to a globally optimal system design, as well as a description of desirable features such methods should have in order to yield attractive empirical performance.

1 Introduction

This chapter is concerned with the use of random search to solve simulation optimization problems of the form

$$\min_{\theta \in \Theta} f(\theta), \tag{1}$$

where Θ is the feasible region and $f : \Theta \to \mathbb{R}$ is the objective function. In practice, the feasible region Θ consists of all possible designs of a system under consideration and $f(\theta)$ denotes the expected performance of that system under the design $\theta \in \Theta$. We assume that the system under consideration is complex enough that the expected performance $f(\theta)$ of each system design $\theta \in \Theta$ cannot be determined exactly, but is instead estimated through simulation. The feasible region Θ is often embedded in a real space (i.e., $\Theta \subset \mathbb{R}^d$, where d is a positive integer), but this is not a requirement for applying random search methods to solve the optimization problem (1). We will outline the basic form of random search methods and discuss the convergence of such methods and other desirable features the methods should have. Other approaches for solving simulation optimization problems of the form (1) are described in Chapters 18, 19 and 21 of this volume.

Random search methods involve sampling points from the feasible region Θ of the underlying optimization problem (1) based on some sampling strategy, evaluating the performance of the objective function f at the chosen points, and then updating the sampling strategy based on the observed history (i.e., the feasible points that have been sampled so far and the associated objective function values), before proceeding to the next iteration. Hence, random search methods constitute a broad class of optimization techniques that can be applied to solve both deterministic and stochastic optimization problems with either discrete or continuous decision parameters (or both). Simulation optimization is a special case of stochastic optimization where the required objective function values $f(\theta)$ are estimated via computer simulation, and hence involve some noise.

Most existing random search methods were originally developed for solving deterministic optimization problems where there is no noise in the estimated objective function values. There are two primary ways in which simulation optimization techniques address the problem of having noise in the estimated objective values (it is also possible to combine ideas from the two approaches). One approach involves expending a significant amount of computer effort at each point visited by the algorithm to obtain a relatively precise estimate of the objective function values at these points (especially as the search for an optimal solution progresses), and then using a deterministic optimization approach to solve the underlying optimization problem. Kleywegt et al. (2001) study the application of this sample path approach to solve discrete optimization problems; see also Healy and Schruben (1991), Rubinstein and Shapiro (1993), Robinson (1996), Shapiro and Wardi (1996), Chen and Schmeiser (2001) and Homem-de-Mello (2003), among others, for related work.

The other approach does not involve obtaining highly precise estimates of the objective function values each time the algorithm visits a feasible point, so that the algorithm must at each step decide on how to proceed based only on limited information. This means that techniques that were originally intended for deterministic optimization generally need to be modified to yield good performance in the presence of noise. The basic issue is that the fact that a particular point *appears* at first to be good (bad), in that it has a small (large) estimated objective function value, see Equation (1), does not necessarily imply that this point is in fact good (bad). Consequently, the optimization technique must proceed cautiously based on the information that is available at the time, so that the method can recover quickly from errors made due to misleading information. For example, it is possible for the method to obtain misleading estimated objective function values suggesting that a bad subset of the feasible region Θ is preferable to a good subset of Θ. When this happens, the method should not completely lose progress made previously with respect to identifying the location of a global optimal solution. Stochastic approximation methods fall within this category of simulation optimization techniques, see, for example, Chapter 19 in this volume and Robbins and Monro (1951), Kiefer and Wolfowitz (1952), Kushner and Clark (1978), Benveniste et al. (1990), Pflug (1996), Kushner and Yin (1997) and Spall (2003). Currently existing random search methods for simulation optimization fall in both categories in that they may require highly precise estimates of the objective function values as the search progresses or not.

Throughout this chapter, we focus on the use of random search methods to optimize the expected performance of stochastic systems using simulation when the number of potential system configurations (i.e., the number of elements of the feasible region Θ) is either finite or countably infinite. This is consistent with the current use of random search methods for solving simulation optimization problems. The objective function values $f(\theta)$ can be estimated using either transient or steady-state simulation, depending on whether we are interested in optimizing the performance of the underlying system over a finite time horizon, or in the long run, respectively.

The outline of this chapter is as follows: In Section 2 we describe the basic structure of random search methods and briefly review the literature on such methods designed for solving simulation optimization problems. In Section 3 we discuss conditions under which random search methods can be shown to be almost surely convergent to the set of global optimal solutions of the optimization problem (1). In Section 4 we discuss properties that random search methods should have in addition to guaranteed convergence to perform well in practice. Finally, Section 5 contains a brief summary of this chapter.

2 A brief review of random search methods

In this section, we provide the general structure and a brief review of random search methods. We start by providing the general structure of random search methods applied to solve the simulation optimization problem (1).

Generic random search algorithm for simulation optimization:

Step 0 (initialize). Choose the initial sampling strategy $\mathcal{S}_1$ and let $n = 1$.

Step 1 (sample). Select $\theta_n^{(1)}, \ldots, \theta_n^{(M_n)} \in \Theta$ according to the sampling strategy $\mathcal{S}_n$.

Step 2 (simulate). Estimate $f(\theta_n^{(i)})$, for $i = 1, \ldots, M_n$, using simulation.

Step 3 (update). Use the simulation results obtained so far in Step 2 to compute an estimate of the optimal solution θ_n^* and to choose an updated sampling strategy $\mathcal{S}_{n+1}$. Let $n = n + 1$ and go to Step 1.

Note that the number M_n of feasible points sampled in iteration n of the algorithm above is a parameter of the sampling strategy $\mathcal{S}_n$ used in iteration n, and consequently does not need to be chosen in advance of executing the algorithm. Also, note that the statement of the algorithm above does not include a stopping criterion. This is typical of the random search literature, and is consistent with the fact that convergence results for random search methods are typically asymptotic in nature; i.e., the sequence $\{\theta_n^*\}$ will be shown to converge in some sense as n approaches infinity. In practice, it is of course necessary to augment this algorithm with a suitable stopping criterion (because otherwise the algorithm involves executing an infinite loop).

The generic random search algorithm given above clearly outlines a broad class of techniques for solving the simulation optimization problem (1). These techniques differ primarily in the choice of the sampling strategy $\{\mathcal{S}_n\}$. For example, the sampling strategy can be "point-based", leading to methods such as simulated annealing and tabu search that involve sampling points in the neighborhood of the current point that the algorithm is visiting, where the neighborhood of a feasible point $\theta \in \Theta$ is a set $N(\theta) \subset \Theta \setminus \{\theta\}$, and the neighborhood structure $N = \{N(\theta)\colon\ \theta \in \Theta\}$ is connected in the sense that it has the feature that for all $\theta, \theta' \in \Theta$, $\theta \neq \theta'$, there exist $\theta_1, \ldots, \theta_k \in \Theta$ such that $\theta_{i+1} \in N(\theta_i)$ for $i = 0, \ldots, k$, where $\theta_0 = \theta$ and $\theta_{k+1} = \theta'$. The sampling strategy can also be "set-based", leading to methods such as (stochastic) branch-and-bound and nested partitions (see Chapter 21) that sample points in a particular subset of the feasible space Θ. Finally, the sampling strategy can be "population-based", leading to methods such as genetic algorithms that generate a new collection of points using properties of the current collection of points. Random search methods also differ in the choice of the sequence of estimates $\{\theta_n^*\}$ of the optimal solution. This is discussed in Section 3.

A detailed discussion of specific random search methods is outside the scope of this chapter. Instead, we focus on desirable properties of random search methods (i.e., guaranteed convergence and attractive empirical performance) and how these properties may be achieved. Nevertheless, to assist readers who are interested in using random search methods to solve simulation optimization problems, we now briefly describe some existing random search methods. Tabu search, nested partitions, and genetic algorithms are reviewed in Chapter 21 of this volume, and hence will not be discussed here.

Stochastic ruler methods constitute one class of random search methods, see Yan and Mukai (1992) and Alrefaei and Andradóttir (2001, 2005). These are point-based methods in which the quality of different feasible points and future movements of the algorithm are decided by comparing estimated objective function values to observations of the stochastic ruler, which is a uniform random variable covering the range of values the objective function f can take. Stochastic comparison and descent algorithms form another class of random search methods. This class would include the point-based methods of Andradóttir (1995, 1996, 1999), Gong et al. (1999) and Prudius and Andradóttir (2004, 2006a), and also the set-based COMPASS method of Hong and Nelson

(2005). These methods involve comparing estimated objective function values at different points with each other, rather than with a stochastic ruler. Algorithm movements take place when a point is found that appears to be better (i.e., has a smaller objective function value, see Equation (1)) than previously visited points. Simulated annealing methods constitute a third class of random search methods, see Gelfand and Mitter (1989), Gutjahr and Pflug (1996), Fox and Heine (1996), Alrefaei and Andradóttir (1999) and Prudius and Andradóttir (2005, 2006b). These point-based methods are designed to escape from locally optimal solutions (and hence be globally convergent), and consequently will sometimes move from the current point to an apparently inferior candidate point in the hope that this will allow the algorithm to eventually find even better points elsewhere in the feasible region Θ. Other related work includes the branch-and-bound method of Norkin et al. (1998a, 1998b), and also the results of Homem-de-Mello (2003) on the convergence of random search.

3 Convergence

One desirable feature of random search methods for discrete simulation optimization is that such methods can be shown to converge almost surely to the set of global optimal solutions of the underlying (discrete) simulation optimization problem under very general conditions. We now review such convergence results. The material in this section is based on Andradóttir (1999, 2006); see also Andradóttir (2000).

In the process of using the generic random search algorithm described in Section 2 to solve a simulation optimization problem of the form (1), simulations will be conducted at any point θ in the feasible region Θ each time the algorithm visits that point (i.e., each time when $\theta \in \{\theta_n^{(1)}, \dots, \theta_n^{(M_n)}\}$). These simulation results are then aggregated to obtain estimates of the objective function value $f(\theta)$, for each $\theta \in \Theta$, that improve with each visit the algorithm makes to θ. We will use the notation $C_n(\theta)$ to refer to the number of times the feasible point θ has been visited in the first n iterations, and $f_n(\theta)$ to refer to the aggregated estimate of $f(\theta)$ available after n iterations have been completed.

We first consider the case when the feasible region Θ is finite, see Andradóttir (1999). In this case, the estimated optimal solution θ_n^* after n iterations have been completed will be chosen from the subset of the feasible points visited so far by the method whose elements have the best (smallest, see Equation (1)) estimated objective function values. Hence,

$$\theta_n^* \in \arg\min_{\theta \in \widetilde{\Theta}_n} f_n(\theta) \tag{2}$$

for all $n \geqslant 1$, where

$$\widetilde{\Theta}_n = \bigl\{\theta \in \Theta\colon\ C_n(\theta) \geqslant 1\bigr\}.$$

If the estimates $f_n(\theta)$ of the objective function are strongly consistent, for all $\theta \in \Theta$, then almost sure convergence to the set of global optimal solutions

$$\Theta^* = \{\theta \in \Theta\colon f(\theta) \leqslant f(\theta'), \forall \theta' \in \Theta\}$$

of the simulation optimization problem (1) is guaranteed. Assuring the strong consistency of the objective function estimates $f_n(\theta)$, for all $\theta \in \Theta$, would typically require that the underlying random search method be guaranteed to visit each feasible point $\theta \in \Theta$ infinitely often with probability one as the number of iterations grows.

For all $\theta \in \Theta$ and $t \geqslant 0$, let $X_\theta(t)$ denote the state of the simulated system at time t when the value of the decision parameter that we want to optimize is given by θ. In transient simulation optimization, the objective function values take the form $f(\theta) = \mathrm{E}\{Y_\theta\}$, where the random variable Y_θ only depends on the system behavior $X_\theta(t)$ until some finite time τ_θ, where τ_θ is a stopping time with respect to the stochastic process $\{X_\theta(t)\}$ (so that $f(\theta)$ can be estimated using transient simulation). For example, if for all $\theta \in \Theta$, $X_\theta(t)$ denotes the number of customers in a $G/G/s$ queue at time t, for all $t \geqslant 0$, $\tau_\theta = T \in (0, \infty)$, and $Y_\theta = \int_0^T X_\theta(t)\,\mathrm{d}t/T$, then $f(\theta)$ represents the average number of customers in the system over the period $[0, T]$. The strong consistency of the estimated objective function values $f_n(\theta)$ in transient simulation optimization is usually achieved by generating one or more observations of Y_θ at each feasible point $\theta \in \{\theta_n^{(1)}, \ldots, \theta_n^{(M_n)}\}$ visited in each iteration n of the algorithm, averaging these observations to obtain an estimate of $f(\theta)$, and then letting $f_n(\theta)$ be the average of the $C_n(\theta)$ estimates of $f(\theta)$ that have been obtained so far by the algorithm. The strong law of large numbers would then be used to prove the strong consistency of the estimated objective function values $f_n(\theta)$, assuming that each feasible point $\theta \in \Theta$ is visited infinitely often by the underlying algorithm with probability one.

In steady-state simulation optimization, we usually have that

$$f(\theta) = \lim_{t \to \infty} \int_0^t h_\theta(X_\theta(u))\,\mathrm{d}u \Big/ t$$

(assuming that the limit exists and equals a constant almost surely), where h_θ is a deterministic function. For example, in the $G/G/s$ example given in the previous paragraph, if for all $\theta \in \Theta$, $h_\theta(x) = x$ for all $x \in \mathbb{R}$, then $f(\theta)$ represents the long-run average number of customers in the system. The strong consistency of the estimators $f_n(\theta)$ in steady-state simulation optimization is usually proved using renewal or regenerative process theory, and can be achieved in various ways, see Andradóttir (2000). One approach involves saving enough information about the simulation at each point $\theta \in \widetilde{\Theta}_n$ visited so far by the algorithm, so that the simulation can be restarted when the algorithm revisits the point θ. Then if the process $\{X_\theta(t)\}$ is simulated for T time units each time when $\theta \in \{\theta_n^{(1)}, \ldots, \theta_n^{(M_n)}\}$, the estimate $f_n(\theta)$ will be an observation of

$$\frac{1}{TC_n(\theta)} \int_0^{TC_n(\theta)} h_\theta\big(X_\theta(u)\big)\,\mathrm{d}u$$

(we could also truncate some observations at the beginning of the run). Another way involves starting simulations from scratch each time the algorithm visits a point $\theta \in \Theta$, letting the length T_k of the simulation run conducted the kth time the algorithm visits a feasible point grow to infinity with k and averaging all the resulting $C_n(\theta)$ estimates, so that

$$f_n(\theta) = \frac{1}{C_n(\theta)} \sum_{k=1}^{C_n(\theta)} \frac{1}{T_k} \int_0^{T_k} h_\theta\big(X_{\theta,k}(u)\big)\,\mathrm{d}u$$

for all $n \geqslant 1$ and $\theta \in \widetilde{\Theta}_n$, where $\{X_{\theta,k}(t)\}$ denotes the sample path of $\{X_\theta(t)\}$ generated the kth time the algorithm visits $\theta \in \Theta$ for all $k \geqslant 1$ (as before, we could truncate some observations at the beginning of each replication).

In both cases (transient and steady-state simulation optimization), one can ensure that each point $\theta \in \Theta$ is visited infinitely often by including a pure search component in the random search method (e.g., in each iteration n, generate a point at random from the feasible region with a small probability). Moreover, many existing random search methods have the property that $M_n = 1$ for all n, and the sequence of points $\{\theta_n^{(1)}\}$ visited by the algorithm is a Markov chain. If this Markov chain is time-homogeneous, irreducible and positive recurrent, then each feasible point $\theta \in \Theta$ will be visited infinitely often with probability one.

We now consider the case when the feasible region Θ is countably infinite, see (Andradóttir, 2006). In this case, we constrain the estimated optimal solution θ_n^* after n iterations of the random search method under consideration have been completed to be chosen from the subset of the feasible points visited *sufficiently often* so far by the algorithm whose elements have the best (smallest) estimated objective function values. (This approach for estimating the optimal solution may also perform better than the approach (2) even when Θ is finite, see Prudius and Andradóttir, 2006a.) Moreover, it is often useful to also constrain θ_n^* to lie in a deterministic set Θ_n, where $\Theta_n \subset \Theta_{n+1}$ for all n and $\bigcup_{n=1}^{\infty} \Theta_n = \Theta$. Consequently, we let

$$\theta_n^* \in \arg\min_{\theta \in \widetilde{\Theta}'_n} f_n(\theta), \tag{3}$$

where

$$\widetilde{\Theta}'_n = \big\{\theta \in \Theta_n\colon\, C_n(\theta) \geqslant K_n\big\}$$

and $\{K_n\}$ is a nondecreasing sequence of positive integers that converges to infinity at a sublinear rate. For example, the sequence $\{\Theta_n\}$ would typically be chosen so that the number of elements $|\Theta_n|$ in Θ_n increases at a polynomial rate with n (e.g., if $\Theta \subset \mathbb{R}^d$, then we could choose $\Theta_n = \Theta \cap \prod_{i=1}^{d}[\tilde{\theta}_i - P_i(n), \tilde{\theta}_i + P_i(n)]$ for all $n \geqslant 1$, where $(\tilde{\theta}_1, \ldots, \tilde{\theta}_d)$ is the user's best initial guess of an optimal solution $\theta^* \in \Theta^*$ and $P_1(n), \ldots, P_d(n)$ are polynomial functions that increase with n), and one could choose $K_n = [an^\alpha]$ for all $n \geqslant 1$, where $a > 0$, $0 < \alpha < 1$, and $[x]$ is the integer closest to x for all $x \in \mathbb{R}$. (The sequence $\{K_n\}$ is constrained to grow at a sublinear rate in n because it is typically possible to show that the number of visits $C_n(\theta)$ to the feasible point θ in the first n iterations of the algorithm grows at a linear rate with n for all θ in some subset of interest of the feasible region. Such feasible points θ would then be included in the set $\widetilde{\Theta}'_n$ with probability one for large n. If the number M_n of points sampled in iteration n grows superlinearly with n, then it is possible that $C_n(\theta)$ will also grow superlinearly with n for some interesting set of feasible points θ, in which case the sequence K_n could be chosen to grow at a linear or superlinear rate with n.)

Suppose that Θ is countably infinite, Equation (3) is used to estimate the optimal solution, the underlying random search method is guaranteed to visit each feasible point sufficiently often as the number of iterations grows (so that the intersection between $\widetilde{\Theta}'_n$ and Θ^* is nonempty for sufficiently large n), and the estimates of the objective function value $f(\theta)$ obtained after the feasible point θ has been visited k times by the algorithm, denoted $\hat{f}_k(\theta)$, are "sufficiently close to" the true objective function value $f(\theta)$ for large k and all $\theta \in \Theta$. Then almost sure convergence to the set Θ^* of global optimal solutions to the simulation optimization problem (1) is guaranteed, see Theorem 3.1 and Section 4 of Andradóttir (2006) for the details. Assuring that each point $\theta \in \Theta$ is visited sufficiently often can typically be achieved in the same manner as when Θ is finite (i.e., by incorporating a pure search component in the method or ensuring that the sequence of points visited by the algorithm constitutes a time-homogeneous, irreducible, and positive recurrent Markov chain). Ensuring that the estimated objective function values $\hat{f}_k(\theta)$ are sufficiently close to the true objective function values $f(\theta)$ for large k and all $\theta \in \Theta$ can usually be ascertained using large deviations theory, see Andradóttir (2006) for more details and Dembo and Zeitouni (1993) for an introduction to large deviations theory.

4 Efficiency

In Section 3 we reviewed under what conditions a random search method is guaranteed to converge almost surely to the set of global optimal solutions Θ^*. These conditions are quite general, and consequently leave a lot of flexibility to develop simulation optimization algorithms that are both provably globally convergent and also highly efficient. Consequently, we believe that there is no reason to consider simulation optimization algorithms in practice that are not guaranteed to converge almost surely to the set of global optimal solutions under weak conditions. Fortunately, many existing random search methods that do not have this property at present can be easily modified in a way that assures almost sure and global convergence (e.g., by adding a pure search component to the method).

However, knowing that a simulation optimization algorithm is globally convergent with probability one does not necessarily imply that this algorithm will perform well in practice. In fact, it is easy to see that certain algorithms that are obviously undesirable are nevertheless guaranteed to converge almost surely to the set of global optimal solutions Θ^* with probability one. For example, when Θ is finite, then repeated enumeration (to repeatedly visit all the points in the state space in a particular order) will converge almost surely to Θ^* as long as the point with the best estimated objective function value is used to estimate the optimal solution (see Equation (2)) and the estimated objective function values obtained after k visits to each feasible point are strongly consistent as k grows.

Moreover, no-free-lunch theorems for deterministic optimization (see Wolpert and Macready, 1997) show that without any knowledge about the structure of the underlying optimization problem, all optimization algorithms will exhibit the same average performance (with the average taken over all possible objective functions). More specifically, suppose that we want to optimize a function $f:\mathcal{X} \to \mathcal{Y}$, where both $\mathcal{X}$ and $\mathcal{Y}$ are finite (e.g., because of finite precision). Suppose furthermore that $d_m = \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\}$ is a time ordered set of m distinct points visited by an optimization algorithm (not counting times when feasible points are revisited), with $d_m^x = \{d_m^x(1), \ldots, d_m^x(m)\}$ being the successive (distinct) feasible points and $d_m^y = \{d_m^y(1), \ldots, d_m^y(m)\}$ being the corresponding objective function values. Then Theorem 1 of Wolpert and Macready (1997) states that if a_1 and a_2 are two optimization algorithms, then

$$\sum_f \mathrm{P}\{d_m^y|f, m, a_1\} = \sum_f \mathrm{P}\{d_m^y|f, m, a_2\}.$$

In other words, the average probability of observing a particular value of d_m^y (averaged over all possible objective functions f) does not depend on the choice of optimization algorithm. This result immediately implies that for any measure of algorithm performance that is a function of d_m^y, the average performance of all optimization algorithms (averaged over all possible f) is the same. Consequently, for an optimization algorithm to do better than repeated enumeration, the underlying optimization problem needs to have some known structure that the algorithm can exploit. Ideas about how such "landscape" structure can be measured can be found in the literature on evolutionary computing, see for example Vassilev et al. (2000) and Reidys and Stadler (2002).

In contrast with the literature on deterministic optimization, which is dominated by methods designed to solve highly structured optimization problems (e.g., linear programs), simulation optimization problems frequently have very little structure that the optimization algorithm can exploit, and even if the underlying problem has a lot of structure, the user may not be aware of this fact. Moreover, since most simulation practitioners use simulation languages to implement their models, for a simulation optimization technique to find widespread use it is desirable that the method be suitable for incorporation in simulation languages. This requires the method to perform well on a broad class of problems having very different characteristics. Consequently, most random search methods are designed to exploit the structure that good points (i.e., points with low objective function values, see (1)) tend to be clustered together. Of course, if more structure is available (e.g., convexity), then this can be exploited to make the search more efficient by narrowing down the location of the global optimal solution(s) as the search progresses.

When a method that is designed to exploit the structure that good points tend to be clustered together has identified a relatively good feasible point (or collection of feasible points), it is sensible to focus the search near that point. Point-based methods do this by focusing the search in a neighborhood of the current point (see Section 3), set-based methods do it by focusing the search in a set containing the current point, and population-based methods focus the search on points that can be generated by manipulating the current collection of points. Additionally, for good performance, these methods must have a mechanism for finding attractive points quickly. This generally translates into ensuring that the method can move rapidly around within the feasible region. A mechanism that accomplishes this needs to be incorporated in the sampling strategy used by the algorithm, regardless of whether it is point-based, set-based, or population-based.

Achieving the goals described in the previous paragraph depends on the set of feasible points under consideration by the sampling strategy $\mathcal{S}_n$ in each iteration n (including the size of this set), and also on how this set is searched. For point-based methods, this translates into the choice of the neighborhoods $N(\theta)$ and the decision about how these neighborhoods are searched. Based on the discussion in the previous paragraph, the neighborhood $N(\theta)$ of a feasible point $\theta \in \Theta$ should include points close to θ (to allow the method to focus the search near θ if desired) and also allow for rapid movements within the feasible region Θ (this would generally involve including some points that are far from θ in $N(\theta)$). Since the choice of the neighborhood structure $N = \{N(\theta)\colon\ \theta \in \Theta\}$ can have a substantial effect on the performance of the random search method under consideration, it is important to try to capture the essential features of the underlying optimization problem (1) in this choice, especially with respect to deciding what points should be considered to be close to each other.

With respect to the size of the neighborhoods, at one extreme one can use very large neighborhoods (e.g., $N(\theta) = \Theta \setminus \{\theta\}$ for all $\theta \in \Theta$), while at the other extreme the neighborhoods can be very localized around the current point (e.g., $N(\theta) = \Theta \cap \{\theta' \in \mathbb{R}^d\colon\ \|\theta' - \theta\| \leqslant 1\}$ when $\Theta \subset \mathbb{R}^d$). Clearly, large neighborhoods allow rapid movements within the feasible region and facilitate achieving global convergence, but they can be less effective than small neighborhoods in focusing the search around desirable points. On the other hand, small neighborhoods can lead to good performance if the algorithm is started in a good point, but can perform poorly if the feasible region Θ is large and many steps are required to move from the starting point(s) $\theta_1^{(1)}, \ldots, \theta_1^{(M_1)}$ to the global optimal solutions (e.g., if Θ is the set of integers, $N(\theta) = \{\theta - 1, \theta + 1\}$ for all $\theta \in \Theta$, $M_1 = 1$, and $\inf_{\theta^* \in \Theta^*} |\theta_n^{(1)} - \theta^*|$ is large).

When it comes to deciding how the neighborhoods $N(\theta)$ should be searched, at one extreme one could sample one point from the neighborhood of the current point and compare it with the current point, while at the other extreme one could search for the best point in the neighborhood and compare that point with the current point. Procedures that search very large neighborhoods constitute an active area of research in the combinatorial optimization community, see Ahuja et al. (2002) for a recent review. However, doing this effectively requires the exploitation of some underlying structure that the optimization problem has. Since such structure is often either not present or unknown in simulation optimization problems, this approach does not appear to be promising for developing simulation optimization algorithms that are suitable for widespread use, for example through being incorporated in a simulation language. Hence, the use of large neighborhood structures for simulation optimization would typically be accompanied by a sampling mechanism over the neighborhoods, rather than an optimization approach.

On the other hand, when the neighborhoods are relatively small in size, then it is possible to either sample from them or optimize over them. In the latter case, there is an opportunity to combine random search methods with ranking-and-selection methods, see Chapter 17 of this volume for a review. However, care needs to be taken to avoid expending too much computer time on optimizing over neighborhoods early in the search when the neighborhoods may not include any desirable points, or on choosing between several points exhibiting very similar performance (except perhaps near the conclusion of the search). Other opportunities for combining random search methods and ranking-and-selection techniques in the simulation optimization setting include using ranking and selection to ensure that the method will only change the estimate of the optimal solution or focus the search in a different subset of the feasible region Θ if there is strong evidence that the change is in fact an improvement over the status quo, and for "clean up" at the end of the search (i.e., to ensure that the final estimate of the optimal solution is in fact the best point visited by the algorithm with high probability). These issues are discussed by Pichitlamken and Nelson (2003) and Boesel et al. (2003), respectively.

The discussion above suggests that it is desirable for a random search method to incorporate both global and local search components, and to maintain the right balance between the two as the search progresses. The global search component is useful for quickly identifying desirable subsets of the feasible region Θ (with small objective function values). Once such subsets have been found, then it is beneficial to search locally within these areas to identify improved solutions. More details about how both global and local search can be included in random search methods, and how appropriate balance can be maintained between the two, are discussed by Prudius and Andradóttir (2004, 2006a). A related issue involves deciding on the appropriate balance between the number of points sampled by the algorithm and how much effort is spent on estimating the objective function value at each sampled point, see for example Yakowitz et al. (2000).

Note that if a random search method is guaranteed to be globally convergent with probability one, then this method does include a mechanism for escaping from feasible points that are locally, but not globally optimal (in other words, such a mechanism should be included in the method). Another desirable feature a random search algorithm may have (in addition to almost sure and global convergence, the ability to focus the search in desirable regions, and the ability to move quickly within Θ) is the ability to focus the search in areas of the feasible region that either appear to be desirable (i.e., have relatively low objective function values) or have not been sampled much previously. This is accomplished in tabu search by excluding certain points from consideration (e.g., points that have been visited recently by the algorithm), and in COMPASS by using a neighborhood structure that focuses the search in the region whose elements are closer to the best point seen so far than they are to other points that the algorithm has already visited.

Furthermore, it is desirable for random search methods to be highly adaptive in order to take as much advantage as possible of the information that it gathers throughout the search to guide the search. In our generic random search algorithm for simulation optimization given in Section 2, we incorporate this by allowing the sampling strategies $\{\mathcal{S}_n\}$ to adapt to all information collected by the algorithm. For example, the neighborhoods used by point-based methods can be chosen adaptively, and the same is true of how these neighborhoods are searched. The only restrictions on this are that adapting to the information collected by the algorithm should in fact improve performance (e.g., the algorithm should not be misled by the noise in the estimated objective function values), this should be done in a way that maintains the guarantee of almost sure convergence, and it is preferable for the algorithm not to require the intervention of the user as it adapts to the available information (e.g., for the method to be suitable for inclusion in a simulation language).

One way of adapting the search to the information collected over time by the random search method is to use the aggregated function estimates $f_n(\theta)$ to guide the search, rather than simply for computing the sequence $\{\theta_n^*\}$ of estimates of the optimal solution, see Equations (2) and (3). The most commonly used alternative to this approach is to use only the simulation results obtained in the current iteration of the algorithm to decide on the next action taken by the algorithm (this is often done to ensure that the progression of the algorithm can be modeled as a Markov chain, which is useful for proving results about the behavior of the algorithm). Since for all $\theta \in \Theta$, the aggregated estimate $f_n(\theta)$ of the objective function value $f(\theta)$ is more precise than an estimate of $f(\theta)$ obtained in a single replication (as long as the feasible point θ has been visited more than once by the random search method), and since the presence of noise in the estimated objective function values makes it more difficult to solve the simulation optimization problem (1), it is reasonable to expect that using the aggregated estimates $f_n(\theta)$ to guide the search will lead to better empirical behavior than using only the most recently obtained estimates. However, this is not always the case, because there are situations where the random search algorithm being used can benefit from the added simulation noise (in addition to any randomness inherent in the algorithm itself). For example, if the random search method being used is a descent method (so that it does not have a mechanism for escaping from solutions that are locally, but not globally optimal) and if the underlying optimization problem has locally optimal solutions that are not globally optimal, then there may be instances where the algorithm will converge to a locally optimal solution when the aggregated estimates $f_n(\theta)$ are used, but it will converge to a globally optimal solution when the most recently available estimates of $f(\theta)$ are used (because the noise in the most recently available estimates allows the algorithm to escape from locally optimal solutions). For more discussion on the use of aggregated objective function estimates in simulation optimization, the reader is referred to Prudius and Andradóttir (2005, 2006b).

The discussion in this section is concerned with features a random search method should have to exhibit attractive empirical performance. More details about the design of random search methods are given by Prudius and Andradóttir (2006a). Finally, note that it would of course be desirable to be able to prove results about how fast a random search method is likely to converge in practice. This would include both asymptotic rate-of-convergence and attraction results, see Andradóttir (1999). Similarly, results about the finite-horizon behavior of random search methods, such as determining how many iterations are required to reach a certain level of performance with a specified probability, would also be valuable. However, such rate-of-convergence results are only of practical value if the conditions under which they hold do not unduly limit the flexibility in the algorithm design (so that an algorithm that satisfies these conditions is not likely to perform worse in practice than the algorithms with the best empirical performance on the class of optimization problems under consideration).

5 Summary

In this chapter, we have outlined the features a random search method should have to both be provably convergent and also exhibit attractive empirical performance. We showed that almost sure and global convergence is not difficult to achieve, and as a result it seems reasonable to focus on random search methods that have this property. We have also discussed other desirable characteristics of random search methods, including the ability to move rapidly within the feasible region to identify desirable areas worthy of further investigation, to focus the search in desirable subsets of the feasible region that have not been extensively explored before, and to adapt to information collected by the method about the optimization problem at hand as it becomes available. The application of these principles towards the design of effective random search methods for simulation optimization is presently an active and worthy research area.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. DMI-0000135, Grant No. DMI-0217860 and Grant No. DMI-0400260. The author thanks the editors of this volume, Shane G. Henderson and Barry L. Nelson, for guidance about the scope of this chapter and other helpful comments. The author also appreciates the comments provided by an anonymous referee.

References

Ahuja, R.K., Ergun, Ö., Orlin, J.B., Punnen, A.P. (2002). A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics* 23, 75–102.

Alrefaei, M.H., Andradóttir, S. (1999). A simulated annealing algorithm with constant temperature for discrete stochastic optimization. *Management Science* 45, 748–764.

Alrefaei, M.H., Andradóttir, S. (2001). A modification of the stochastic ruler method for discrete stochastic optimization. *European Journal of Operational Research* 133, 160–182.

Alrefaei, M.H., Andradóttir, S. (2005). Discrete stochastic optimization using variants of the stochastic ruler method. *Naval Research Logistics* 52, 344–360.

Andradóttir, S. (1995). A method for discrete stochastic optimization. *Management Science* 41, 1946–1961.

Andradóttir, S. (1996). A global search method for discrete stochastic optimization. *SIAM Journal on Optimization* 6, 513–530.

Andradóttir, S. (1999). Accelerating the convergence of random search methods for discrete stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 9, 349–380.

Andradóttir, S. (2000). Rate of convergence of random search methods for discrete optimization using steady-state simulation. Preprint.

Andradóttir, S. (2006). Simulation optimization with countably infinite feasible regions: Efficiency and convergence. *ACM Transactions on Modelling and Computer Simulation*, in press.

Benveniste, A., Métivier, M., Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer-Verlag, Berlin.

Boesel, J., Nelson, B.L., Kim, S.-H. (2003). Using ranking and selection to "clean up" after simulation optimization. *Operations Research* 51, 814–825.

Chen, H., Schmeiser, B.W. (2001). Stochastic root finding via retrospective approximation. *IIE Transactions* 33, 259–275.

Dembo, A., Zeitouni, O. (1993). *Large Deviations Techniques and Applications*. Jones and Bartlett, Boston, MA.

Fox, B.L., Heine, G.W. (1996). Probabilistic search with overrides. *The Annals of Applied Probability* 6, 1087–1094.

Gelfand, S.B., Mitter, S.K. (1989). Simulated annealing with noisy or imprecise energy measurements. *Journal of Optimization Theory and Applications* 62, 49–62.

Gong, W.-B., Ho, Y.-C., Zhai, W. (1999). Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization* 10, 384–404.

Gutjahr, W.J., Pflug, G.Ch. (1996). Simulated annealing for noisy cost functions. *Journal of Global Optimization* 8, 1–13.

Healy, K.J., Schruben, L.W. (1991). Retrospective simulation response optimization. In: Nelson, B.L., Kelton, W.D., Clark, G.M. (Eds.), *Proceedings of the 1991 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 901–906.

Homem-de-Mello, T. (2003). Variable-sample methods for stochastic optimization. *ACM Transactions on Modeling and Computer Simulation* 13, 108–133.

Hong, L.J., Nelson, B.L. (2005). Discrete optimization via simulation using COMPASS. *Operations Research* 54, 115–129.

Kiefer, J., Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* 23, 462–466.

Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T. (2001). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12, 479–502.

Kushner, H.J., Clark, D.S. (1978). *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, NY.

Kushner, H.J., Yin, G.G. (1997). *Stochastic Approximation Algorithms and Applications*. Springer-Verlag, New York, NY.

Norkin, V.I., Ermoliev, Y.M., Ruszczyński, A. (1998a). On optimal allocation of indivisibles under uncertainty. *Operations Research* 46, 381–395.

Norkin, V.I., Pflug, G.Ch., Ruszczyński, A. (1998b). A branch and bound method for stochastic global optimization. *Mathematical Programming A* 83, 425–450.

Pflug, G.Ch. (1996). *Optimization of Stochastic Models: The Interface between Simulation and Optimization*. Kluwer, Boston, MA.

Pichitlamken, J., Nelson, B.L. (2003). A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 13, 155–179.

Prudius, A.A., Andradóttir, S. (2004). Simulation optimization using balanced explorative and exploitative search. In: Ingalls, R.G., Rossetti, M.D., Smith, J.S., Peters, B.A. (Eds.), *Proceedings of the 2004 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 545–549.

Prudius, A.A., Andradóttir, S. (2005). Two simulated annealing algorithms for noisy objective functions. In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (Eds.), *Proceedings of the 2005 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 797–802.

Prudius, A.A., Andradóttir, S. (2006a). Balanced explorative and exploitative search with estimation. Working paper.

Prudius, A.A., Andradóttir, S. (2006b). An averaging framework for simulation optimization with applications to simulated annealing. Working paper.

Reidys, C.M., Stadler, P.F. (2002). Combinatorial landscapes. *SIAM Review* 44, 3–54.

Robbins, H., Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics* 22, 400–407.

Robinson, S.M. (1996). Analysis of sample-path optimization. *Mathematics of Operations Research* 21, 513–528.

Rubinstein, R.Y., Shapiro, A. (1993). *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*. Wiley, Chichester, England.

Shapiro, A., Wardi, Y. (1996). Convergence analysis of gradient descent stochastic algorithms. *Journal of Optimization Theory and Applications* 91, 439–454.

Spall, J.C. (2003). *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. Wiley, Hoboken, NJ.

Vassilev, V.K., Fogarty, T.C., Miller, J.F. (2000). Information characteristics and the structure of landscapes. *Evolutionary Computation* 8, 31–60.

Wolpert, D.H., Macready, W.G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 67–82.

Yakowitz, S., L'Ecuyer, P., Vázquez-Abad, F. (2000). Global stochastic optimization with low-dispersion point sets. *Operations Research* 48, 939–950.

Yan, D., Mukai, H. (1992). Stochastic discrete optimization. *SIAM Journal on Control and Optimization* 30, 594–612.

This page intentionally left blank

S.G. Henderson and B.L. Nelson (Eds.), *Handbook in OR & MS, Vol. 13*
Copyright © 2006 Elsevier B.V. All rights reserved
DOI: 10.1016/S0927-0507(06)13021-2

Chapter 21

Metaheuristics

Sigurdur Ólafsson

Department of Industrial and Manufacturing Systems Engineering,
Iowa State University, USA
E-mail: olafsson@iastate.edu

Abstract

Metaheuristics have been established as one of the most practical approaches to simulation optimization. However, these methods are generally designed for combinatorial optimization, and their implementations do not always adequately account for the presence of simulation noise. Research in simulation optimization, on the other hand, has focused on convergent algorithms, giving rise to the impression of a gap between research and practice. This chapter surveys the use of metaheuristics for simulation optimization, focusing on work bridging the current gap between the practical use of such methods and research, and points out some promising directions for research in this area. The main emphasis is on two issues: accounting for simulation noise in the implementation of metaheuristics, and convergence analysis of metaheuristics that is both rigorous and of practical value. To illustrate the key points, three metaheuristics are discussed in some detail and used for examples throughout, namely genetic algorithms, tabu search, and the nested partitions method.

Keywords: Simulation optimization, metaheuristics, tabu search, genetic algorithms, nested partitions method, convergence analysis

1 Introduction

Optimization in simulation has been a topic of intense investigation for decades, but for most of this period, the research effort was rarely transferred to simulation practice. This has changed dramatically over the past decade, however, and optimization routines are now part of most commercial simulation software. On the other hand, these commercial simulation optimizers draw little on the decades of academic research in this field. In fact, they almost exclusively use metaheuristics that have been most extensively studied to solve deterministic combinatorial optimization problems (April et al., 2003; Fu, 2002; Ólafsson and Kim, 2002).

Several classes of simulation optimization problems and solution methodologies have been analyzed in the literature. When the underlying space is continuous, gradient search methods are applicable and derivative estimation is a key issue (see Chapter 19), when the number of alternatives is small, statistical selection methods can be applied (see Chapter 17), and in many situations, metamodeling and response surface methods have been found useful (see Chapter 18). Another relevant line of research has focused on provably convergent random search methods (see Chapter 20). In this chapter we address the use of metaheuristics in simulation optimization. Our focus is on basic methodology and research issues, but we make a deliberate effort to highlight work that bridges the gap between simulation practice and research, and indicate promising research directions in this area that are motivated by the prevalence of metaheuristics.

In simulation practice, the choice and design of an algorithm always boils down to computational efficiency (Kelly, 2002). Rapid progress of the search and demonstrating improvement over the initial solution therefore takes precedence over possible convergence statements, and provably obtaining a global optimum is rarely a significant concern. While this is certainly a reasonable stance in practice, a consequence of this view may be that the simulation takes on a subservient role to the optimization (Fu, 2002). This, in our opinion, is an undesirable property for simulation optimization software. Thus, the challenge to the research community is to shift the focus back to the simulation itself, by rigorously accounting for the simulation noise in the optimization routines, and establishing convergence results that are both well founded and useful in practice.

Indeed, although more advanced implementations do consider statistical significance, there are some concerns about how metaheuristics developed for combinatorial optimization perform for simulation problems, and there are significant lessons that can be learned from the existing body of research. In particular, we believe that by understanding how to account for simulation noise in metaheuristics, their performance in practice may be improved. We also believe that convergence statements do have value, and that it is reassuring to know that given enough computational effort, the search will eventually be successful. Furthermore, it would be ideal to be able to terminate the search and know at that moment how good a solution has been obtained. With regards to metaheuristics, relatively little work has been done in the deterministic combinatorial optimization context to analyze convergence in this manner, and even less in the simulation optimization context. Thus, with the assumption that metaheuristics will continue to dominate simulation practice, we advocate a research agenda that focuses on how to account for simulation noise in these methods, and making convergence statements that are applicable to how the algorithms are actually employed.

As stated above, several simulation optimization packages have recently been developed for commercial simulation software. These include SimRunner® that is used by Promodel® and OptQuest®, which can be used by numerous simulation software packages such as Arena®, Crystal Ball®, Promodel®, SIMUL8®. Such packages use one or more common metaheuristics as their primary search method. Specifically, genetic algorithms are used in SimRunner®, and tabu search, genetic algorithms, and scatter search are all used in OptQuest®. A recent survey that discussed the use of simulation optimization in such commercial software can be found in April et al. (2003). In this chapter we focus on three metaheuristics methods that have either been used widely in practice or shed some light on the convergence issues to be studied. In particular, we study genetic algorithms, tabu search, and the nested partitions method. We review the state of the art with regard to the use of these methods for simulation optimization; consider how to account for simulation noise in their implementation; and explore what convergence statements can be made.

The remainder of the chapter is organized as follows. In Section 2 we give a brief review of metaheuristics methods and their uses in combinatorial optimization. A comprehensive review is beyond the scope of the chapter, and the focus is on common elements and what defines each method. In Section 3 we discuss how those methods can be applied to simulation optimization and focus in particular on what makes simulation optimization fundamentally different and how the challenges of noisy performance can be addressed. The next three sections discuss specific metaheuristics, namely genetic algorithms, tabu search, and the nested partitions method. The focus of these sections is on the issues that arise when these methods are applied in simulation optimization. Section 7 is a discussion of what type of convergence statements can be made for metaheuristics, with a focus on of both asymptotic and finite-time convergence results. Examples from genetic algorithms and nested partitions methods are used for illustration. Finally, in Section 8, we summarize the research agenda that we advocate for metaheuristics in simulation optimization.

2 Background to metaheuristics

Metaheuristics are designed to tackle complex optimization problems where other optimization methods have failed to be either effective or efficient. These methods have come to be recognized as one of the most practical approaches for solving many complex problems, and this is particularly true for the many real-world problems that are combinatorial in nature. The practical advantage of metaheuristics lies in both their effectiveness and general applicability. In the early research literature, specialized heuristics were typically developed to solve complex combinatorial optimization problems. This required a new approach to every problem and lessons learned from one problem did not always generalize well to a different class of problems. On the other hand, with the emergence of more general solution strategies, including such metaheuristics as tabu search, genetic algorithms and simulated annealing, the main challenge has become adapting the metaheuristics to a particular problem or problem class. This usually requires much less work than developing a specialized heuristic for a specific application, which makes metaheuristics an appealing choice for implementation in general purpose software. Furthermore, a good metaheuristic implementation is likely to provide near optimal solutions in reasonable computation times. For further reading, Glover and Kochenberger (2003) provide a good introduction and general reference to many of the most popular metaheuristics.

The applicability of metaheuristics as a preferred method over other optimization methods is primarily to find good heuristic solutions to complex optimization problems with many local optima and little inherent structure to guide the search. The metaheuristic approach to solving such problem is to start by obtaining an initial solution or an initial set of solutions, and then initiating an improving search guided by certain principles. The structure of the search has many common elements across various methods. In each step of the search algorithm, there is always a solution (or a set of solutions) θ_k, which represents the current state of the algorithm. Many metaheuristics, including simulated annealing, tabu search, variable neighborhood search, and GRASP, are solution-to-solution search methods, that is, θ_k is a single solution or point $\theta_k \in \Theta$ in some solution space Θ. Others, including genetic algorithms, scatter search and the nested partitions method, are set-based, that is, in each step θ_k represents a set of solutions $\theta_k \subseteq \Theta$. However, the basic structure of the search remains the same regardless of whether the metaheuristics is solution-to-solution or set-based.

Given a neighborhood $N(\theta_k)$ of the solution (set), a candidate solution (set) $\{\theta^c\} \subset N(\theta_k)$ is selected and evaluated. This evaluation involves calculating or estimating the performance of the candidate solution(s) and comparing them with the performance of θ_k and sometimes with each other. Based on this evaluation, the candidate may be either accepted, in which case $\theta_{k+1} = \theta^c$, or rejected, in which case $\theta_{k+1} = \theta_k$. We now have the following metaheuristic framework:

```
**Obtain** an initial solution (set) $\theta_0$ and set $k = 0$.
**Repeat**:
    Identify the neighborhood $N(\theta_k)$ of the current solution(s).
    Select candidate solution(s) $\{\theta^c\} \subset N(\theta_k)$ from the neighborhood.
    Accept the candidate(s) and set $\theta_{k+1} = \theta^c$ or reject it and
    set $\theta_{k+1} = \theta_k$.
    Increment $k = k + 1$.
**Until** stopping criterion is satisfied.
```

As we will see in this section, this framework can be applied to numerous metaheuristics.

The reason for the “meta-” prefix is that metaheuristics do not specify all the details of the search, which can thus be adapted by a local heuristic to a specific application. Instead, they specify general strategies to guide specific aspects of the search. For example, tabu search uses a list of solutions or moves called the tabu list, which ensures the search does not revisit recent solutions or becomes trapped in local optima. The tabu list can thus be thought of as a restriction of the neighborhood. On the other hand, methods such as genetic algorithm specify the neighborhood as all solutions that can be obtained by combining the current solutions through certain operators. Other methods, such as simulated annealing, do not specify the neighborhood in any way, but rather specify an approach to accepting or rejecting solutions that allows the method to escape local optima. Finally, the nested partitions method is an example of a set-based method that selects candidate solutions from the neighborhood with a probability distribution that adapts as the search progresses to make better solutions be selected with higher probability.

Within the framework presented here, all metaheuristics share the elements of selecting candidate solution(s) from a neighborhood of the current solution(s) and then either accepting or rejecting the candidate(s). With this perspective, each metaheuristic is thus defined by specifying one or more of these elements, but allowing others to be adapted to the particular application. This may be viewed as both a strength and a liability. It implies that we can take advantage of special structure for each application, but it also means that the user must specify those aspects, which can be complicated. For the remainder of this section, we briefly introduce a few of the most common metaheuristics and discuss how they fit within this framework. Three of those methods will then be analyzed in more detail as we discuss how to apply them for simulation optimization in subsequent sections.

One of the earliest metaheuristics is *simulated annealing* (Kirkpatrick et al., 1983; Eglese, 1990; Fleischer, 1995), which is motivated by the physical annealing process, but within the framework here simply specifies a method for determining if a solution should be accepted. As a solution-to-solution search method, in each step it selects a candidate $\theta^{\mathrm{c}} \in N(\theta_k)$ from the neighborhood of the current solution $\theta_k \in \Theta$. The definition of the neighborhood is determined by the user. If the candidate is better than the current solution it is accepted, but if it is worse it is not automatically rejected, but rather accepted with probability

$$\mathrm{P}\bigl[\text{Accept } \theta^{\mathrm{c}}\bigr] = \mathrm{e}^{(f(\theta_k)-f(\theta^{\mathrm{c}}))/T_k}, \tag{1}$$

where $f:\Theta \to \mathbb{R}$ is an objective function to be minimized, and T_k is a parameter called the temperature. Clearly, the probability of acceptance is high if the performance difference is small and T_k is large. The key to simulated annealing is to specify a cooling schedule $\{T_k\}_{k=1}^{\infty}$, by which the temperature is reduced so that initially inferior solutions are selected with a high enough probability so local optimal are escaped, but eventually it becomes small enough so that the algorithm converges. We do not discuss simulated annealing any further here and refer the interested reader to Chapter 20 for its use in simulation optimization.

Tabu search is another popular metaheuristics (Glover, 1989, 1990; Glover and Laguna, 1997). As is the case for simulated annealing, it is a solution-to-solution search method where the neighborhood is specified by the user. However, the defining characteristic of tabu search is in how solutions are selected from the neighborhood. In each step of the algorithm, there is a list L_k of solutions that were recently visited and are therefore tabu. The algorithm looks through all of the solutions of the neighborhood that are not tabu and selects the best one, that is,

$$\theta^{\mathrm{c}} = \mathop{\arg\min}_{\theta \in N(\theta_k) \cap \overline{L}_k} f(\theta), \tag{2}$$

where as before $f : \Theta \to \mathbb{R}$ is to be minimized. The candidate solution θ^{c} is accepted even if it is worse than the current solution, that is, $\mathrm{P}[\text{Accept } \theta^{\mathrm{c}}] = 1$. Accepting inferior solutions allows the search to escape local optima, and the tabu list prevents the search from immediately reverting to the previous solution. Tabu search for simulation optimization is discussed further in Section 5. However, tabu search has many intricacies and variations that are not discussed in this chapter, and we refer the interested reader to Glover and Laguna (1997) for a more complete discussion of this method.

Other popular solution-to-solution metaheuristics include the *greedy randomized adaptive search procedure* (GRASP) and the *variable neighborhood search* (VNS). The defining property of GRASP is its multi-start approach that initializes several local search procedures from different starting points. The advantage of this is that the search becomes more global, but on the other hand, each search cannot use what the other searches have learned, which introduces some inefficiency. The VNS is interesting in that it uses an adaptive neighborhood structure, which changes based on the performance of the solutions that are evaluated. More information on GRASP can be found in Resende and Ribeiro (2003), and for an introduction to the VNS approach we refer the reader to Hansen and Mladenovic (1999).

Several metaheuristics are set-based or population based, rather than solution-to-solution. This includes genetic algorithms and other evolutionary approaches, as well as scatter search and the nested partitions method. All of these methods are readily adapted to simulation optimization, and both genetic algorithms and the nested partitions method are discussed in detail in later sections. All *evolutionary algorithms* are based on the idea of natural selection, where a population of solutions evolves, or improves, through a series of genetic operators (Goldberg, 1989; Liepins and Hilliard, 1989; Muhlenbein, 1997). This includes survival of the fittest or best solutions, crossover, which is simply a combination of two fit solutions, and mutation, which is a slight modification to fit solutions. From the perspective of the general framework, these operators define the neighborhood of a solution set. Thus, given a current set $\theta_k \subseteq \Theta$ of solutions, the neighborhood is defined as

$$N(\theta_k) = N^{crossover}(\theta_k) \cup N^{mutation}(\theta_k) \cup \theta_k, \tag{3}$$

where

$$\begin{aligned}
&N^{crossover}(\theta_k)\\
&\quad= \{\psi \in \Theta \mid \psi \text{ is the crossover of two solutions } \zeta_1, \zeta_2 \in \theta_k\},\\
&N^{mutation}(\theta_k) = \{\psi \in \Theta \mid \psi \text{ is a mutation of some } \zeta \in \theta_k\}.
\end{aligned} \tag{4}$$

For evolutionary methods, the key feature is therefore this innovative definition of a neighborhood, which allows the search to quickly and intelligently traverse large parts of the solution space. The selection of candidates from the neighborhood is either deterministic or random, which is discussed further in Section 4. The selected candidates are then always accepted.

Scatter search is another metaheuristic related to the concept of evolutionary search. In each step a scatter search algorithm considers a set, $\theta_k \subseteq \Theta$, of solutions called the reference set. Similar to the genetic algorithm approach, these solutions are then combined into a new set $\theta_{k+1} \subseteq \Theta$. However, as opposed to the genetic operators, in scatter search the solutions are combined using linear combinations, which thus define the neighborhood $N(\theta_k)$. For references on scatter search, we refer the reader to Glover et al. (2003).

The final method that we mention in this section is the *nested partitions* method (Shi and Ólafsson, 2000a), which like genetic algorithms and scatter search is a set-based metaheuristic. Unlike any of the previous methods, however, the nested partitions method is global in that the neighborhood is always the entire set of feasible solutions. Thus, given a current set $\theta_k \subseteq \Theta$ of solutions, the neighborhood is always $N(\theta_k) = \Theta$. In addition to the global perspective, the defining element of this method is the adaptive distribution that is used to obtain sample solutions from this neighborhood. By going through a sequence of set partitions, with each partition nested within the last, the sampling is concentrated in sets that are considered promising, that is, where the optimal solution is believed to be contained. Once the random candidate solutions have been generated according to this adaptive distribution, they are always accepted, and the search continues with a new sampling distribution. The nested partitions method for simulation optimization is discussed in more detail in Section 6.

Although the metaheuristics discussed in this section are usually considered separately in the literature, we have made the case that they do in fact have many common elements that make it possible to analyze them within a common framework. An interesting step in this direction is the generalized hill-climbing (GHC) algorithm framework of Jacobson et al. (1998). The GHC framework is general enough to cover various stochastic local search algorithms, including both simulated annealing and tabu search. Analysis of the GHC can be found in Johnson and Jacobson (2002) and Sullivan and Jacobson (2001).

3 Accounting for simulation noise

Metaheuristics have been found to be particularly effective for combinatorial optimization, and it is therefore natural to examine if they perform similarly when applied to such problem in the stochastic or simulation optimization context. In this section, we define the simulation optimization problem, examine what makes this problem uniquely different from the corresponding deterministic problem, and discuss how metaheuristics need to account for the simulation noise when solving such problems.

We define the general simulation optimization problem to be studied as follows. Given a finite feasible region Θ, and performance measure $J : \Theta \to \mathbb{R}$, the objective is to find the solution $\theta^* \in \Theta$ that minimizes the objective, that is,

$$\theta^* = \arg\min_{\theta \in \Theta} J(\theta). \tag{5}$$

In addition to the general difficulties association with combinatorial optimization, we have the complexity that for any solution $\theta \in \Theta$, the performance $J(\theta)$ cannot be evaluated analytically and must be estimated using simulation. The performance is often an expectation of some random estimate of the performance of a complex stochastic system given the parameter θ, that is,

$$J(\theta) = \mathrm{E}\bigl[L(\theta)\bigr]. \tag{6}$$

Here, $L(\theta)$ is random variable, which depends on the parameter $\theta \in \Theta$, which we assume to be the sample performance from a simulation run.

Numerous metaheuristics, such as genetic algorithms, simulated annealing, the nested partitions method, and tabu search, have been adapted for solving the problem above. As mentioned above, metaheuristics, and in particular evolutionary algorithms, certainly dominate simulation practice, and numerous articles that discuss metaheuristics for simulation and stochastic optimization, mostly application oriented, have appeared in the research literature. For same examples of this line of research, see Azadivar and Wang (2000), Faccenda and Tenga (1992), Haddock and Mittenhall (1992), Hall and Bowden (1997), Tompkins and Azadivar (1995) and Watson et al. (2003).

Despite their definite success in practice, directly applying metaheuristics that are designed for combinatorial optimization problems to simulation optimization may ignore some of the differences that exist from deterministic environments. Metaheuristics are generally not designed to account for the simulation noise, which as we discuss below may be critical. As discussed in Section 2, these methods attempt to move from one solution to another better solution, or one set of solutions to another set of solutions, and thus iteratively improving the solution quality until the method terminates. However, when simulation is used to estimate the performance, determining what constitutes a better solution becomes an issue. Given two solutions θ_1 and θ_2, and simulation performance estimates $\widehat{J}(\theta_1)$ and $\widehat{J}(\theta_2)$, the fact that $\widehat{J}(\theta_1) < \widehat{J}(\theta_2)$ does not guarantee that $J(\theta_1) < J(\theta_2)$. In particular, we now have to look at the statistical significance. In other words, the question is whether $\widehat{J}(\theta_1)$ is statistically significantly smaller than $\widehat{J}(\theta_2)$? How this should be implemented in a search algorithm is an issue that has received very little attention. It could be argued that a solution should only be considered better, and hence the algorithm should move to this solution, only if the improvement is statistically significant at some level. However, it is also possible that a sequence of improvements that were not significant individually will become significant when taken together (Kelly, 2002), and many metaheuristics are quite robust to certain level of randomness.

Accounting for the simulation noise is a theoretically challenging issue that has real implications to simulation practice. However, this issue has received somewhat limited attention in the research literature to date. Many of the relevant research issues are application specific, but at least two general research directions seem promising. In particular, ideas from both ordinal optimization and statistical selection have been successfully applied to guide the search in stochastic environments, and we believe more research is warranted in both areas.

One approach to simulation optimization is to shift the focus from cardinal to ordinal comparison (Ho et al., 2000). Making a decision to move from one solution to another only requires comparison between the solutions, that is, if we know that $J(\theta_1) < J(\theta_2)$, then we know if the move should be made. Such comparisons are the focus of ordinal optimization (Ho et al., 1992). It can be shown that ordinal comparisons converge much faster than cardinal estimation (Dai, 1996), which could be utilized to guide the search. These considerations are relevant to any metaheuristic as they all move based on comparisons between solutions.

An alternative approach is to incorporate statistical selection methods into the search. For the nested partitions method, Ólafsson (2004) proposes using statistical selection to guide the movement of the search. In particular, for every step of the algorithm, a statistical selection procedure is used to determine the required amount of samples from each region so that the correct region is selected within an indifference zone with a given minimum probability. In related work, Shi and Chen (2000) use statistical selection to determine the simulation effort for each sample solution used by the nested partitions method. Various statistical selection methods have been studied extensively in simulation (see Chapter 17), and it should be possible to take advantage of these methods in a similar fashion for other metaheuristics.

Another issue in accounting for the simulation noise is how to recover from the incorrect moves that will inevitably be made due to this noise. Tabu search, for example, uses the tabu list to disallow moves to previously visited solutions. This feature enables it to escape local optima and is intuitively appealing in the deterministic context, when we know with certainty the quality of those previously visited solutions, as well as those solutions that are in the neighborhood of the previously visited solutions. However, in the simulation optimization context it is possible, and even likely, that an incorrect move was selected due to the simulation noise, which may make it worthwhile to revisit a solution for a reason that does not exist in the deterministic context. How this can be solved, while retaining the appealing property of the tabu list, is an issue that needs to be addressed.

In genetic algorithms, a solution is selected for survival or crossover due to its fitness. Fit solutions will reoccur either unchanged or as part of their offspring, whereas unfit solutions will not survive. However, in the simulation context, the fitness of these solutions is subject to the same simulation noise and thus good solutions may be judged unfit prematurely. The question now arises of how to account for the simulation noise. One approach might be to account for the simulation noise in every generation by making sure that the selection for survival or crossover is done at a statistically significant level, or by employing statistical screening procedures to select a subset of solutions for survival or crossover. However, genetic algorithms seem quite robust to inaccuracies in the performance estimates as it is actually preferable to have some randomness in the selection of solutions, and further research is certainly warranted to resolve these issues.

The nested partitions method includes a built-in mechanism for recovering from incorrect moves. In every step, the neighborhood of the current solution is the whole solution space, so any solution has a positive probability of being selected to be evaluation. This includes the surrounding region, that is, those solutions that are not in the set currently considered most promising. If the best solution is found in the surrounding region, the algorithm backtracks to an earlier (larger) set of solutions. This can be viewed as an automated mechanism to recover from an incorrect move made either due to the use of random samples to guide the search, or due to the simulation noise. These global search and backtracking mechanisms of the nested partitions method can therefore be used as a global guidance system to local improvement methods in simulation optimization (Pichitlamken and Nelson, 2003). In the context of combinatorial optimization, other efficient metaheuristics such as genetic algorithms (Shi et al., 1999) and tabu search (Shi and Men, 2003) have been incorporated into the nested partitions method. These methods are used to speed the evaluation of each region, but the nested partitions guide the overall search. In terms of computational efficiency, this approach has been found to perform favorably when compared to the direct use of either genetic algorithms or tabu search for many applications, and in the simulation optimization context, it has the added benefit of allowing the search to recover from mistakes made due to the simulation noise. Such promising results from combinatorial optimization applications also motivate further investigation of such global guidance systems for metaheuristics in simulation optimization.

Finally, an overarching issue in accounting for simulation noise for any metaheuristic is the question of how the inevitably very limited computational effort should be spent. That is, given a fixed computing-budget, there is a tradeoff between obtaining high quality estimates of each solution, and allowing the search to traverse the solution space quickly and exploring more solutions. At least some partial resolution to this issue is the goal of much ongoing research in this area, and has implications in the design, implementation, and analysis of the algorithms.

4 Genetic algorithm

This section looks closer at one of the popular metaheuristics for simulation optimization, namely genetic algorithms. As an approach to global optimization, genetic algorithms (GA) have been found to be applicable to optimization problems that are intractable for exact solutions by conventional methods (Holland, 1975; Goldberg, 1989). It is a set-based search algorithm, where at each iteration it simultaneously generates a number of solutions. In each iteration, a subset of the current set of solutions is selected based on their performance and these solutions are combined into new solutions. The operators used to create the new solutions are survival, where a solution is carried to the next iteration without change, crossover, where the properties of two solutions are combined into one, and mutation, where a solution is modified slightly. The same process is then repeated with the new set of solutions. The crossover and mutation operators depend on the representation of the solution, but not on the evaluation of its performance. They are thus the same even though the performance is estimated using simulation. The selection of solutions, however, does depend on the performance. The general principle is that high performing solutions (which in genetic algorithms are referred to as fit individuals) should have a better chance of both surviving and being allowed to create new solutions through crossover. The simplest approach is to order the solutions $J(\theta_{[1]}) \leqslant J(\theta_{[2]}) \leqslant \cdots \leqslant J(\theta_{[n]})$, and only operate on the best solutions. If a strict selection of say, the top k solutions, were required, this would complicate the issue significantly in the simulation optimization context, and considerable simulation effort would have to be spent to obtain an accurate ordering of the solutions.

Fortunately, genetic algorithms appear to be quite robust with respect to which solutions are selected to create the next set. Indeed, a purely deterministic selection of the top k solution is typically not the best approach for deterministic problems, and some randomness is usually introduced into the process. A popular example of this is the roulette strategy, which several authors have used for the application of genetic algorithms to stochastic problems (see, e.g., Dasgupta and Mcgregor, 1992; Grefenstette, 1992; Ishibuchi and Murata, 1996; Vavak and Fogarty, 1996; Yoshitomi and Yamaguchi, 2003). In the roulette strategy the probability of selecting a solution θ is calculated as follows,

$$P(\theta) = \frac{\hat{f}(\theta)}{\sum_{\mathrm{All}\,\theta} \hat{f}(\theta)} \tag{7}$$

Here $\hat{f}(\theta)$ is an estimate of the fitness function $f:\Theta \to \mathbb{R}$ that measures the quality of the solution and is to be maximized (higher value implies more fit). Thus, every solution has a positive probability of being selected, but the fitter solutions are selected with higher probability. Assuming an unbiased simulation estimate, this statement will continue to be true when the roulette strategy is applied to simulation optimization. However, existing studies in this area are based almost exclusively on numerical evaluations and the claim is simply made that genetic algorithms are robust and hence applicable to simulation optimization. While the robustness of genetic algorithms with respect to noise in the selection method is unquestionable, it would be desirable to have research that provides a better understanding into how much noise is acceptable, and thus how much simulation effort should be devoted to the evaluation of solutions during the search.

In another study, Boesel et al. (2003a) use genetic algorithm together with statistical selection method to develop a system for simulation optimization. Their approach consists of three phases. First, there is an initialization phase, where the parameters of the algorithms are specified. Second, there is the actual search phase, that is, the usual iteration of the genetic algorithm selection, crossover, and mutation operators. Finally, then the search terminates there is a solutions phase where the alternatives generated by the GA search are evaluated using ranking and selection. In particular, they use a screening and selection procedure to first quickly filter out inferior solutions and then determine the best solution by carrying out additional simulation runs for the remaining solutions. They also consider how best to implement the solution selection procedure in a noisy environment. Their design choice is to use what is called anti-ranks and a q-tournament selection (Back, 1996), which also appears to be quite robust with regards to the simulation noise.

Genetic algorithms and other evolutionary approaches appear to be readily adaptable to simulation optimization. As discussed in this section, the primary reason is that only the selection of solutions for survival, crossover, and mutation depends on the performance estimates, and a certain amount of noise is desirable in this process even for deterministic optimization. Thus, it is possible to envision that certain amount of simulation noise might even assist the search. However, most results that point in that direction are based on numerical evidence and we encourage further research in this area that focuses on how to best select solutions in the presence of noise.

5 Tabu search

Tabu search was introduced by Glover (1989, 1990) to solve combinatorial optimization problems and it has been used effectively for simulation optimization, most notably by the OptQuest simulation optimization software (April et al., 2003). It is a solution-to-solution method and the main idea is to make certain moves or solutions tabu, that is they cannot be visited as long as they are on what is called the tabu list. The tabu list L_k is dynamic and after each move, the latest solution θ_k, or the move that resulted in this solution, is added to the list and the oldest solution or move is removed from the list. Another defining characteristic of tabu search is that the search always selects the best nontabu solution from the neighborhood, even if it is worse than the current solution. This allows the search to escape local optima, and the tabu list ensures that the search does not revert back. Tabu search has numerous other elements, such as long-term memory that restarts the search with a new tabu list at previously found high quality solutions, and a comprehensive treatment of this methodology can be found in Glover and Laguna (1997).

Several aspects of tabu search are affected by the noisy estimates of simulation optimization. First, the simulation noise is relevant to one of the two main components of this approach, namely that in every step the best nontabu solution is selected from the neighborhood $N(\theta_k)$ of the current solution θ_k, that is, $\theta_{k+1} = \arg\min_{\theta \in N(\theta_k) \cap \overline{L}_k} \widehat{J}(\theta)$ where $\overline{L}_k$ denotes the compliment of L_k, or the nontabu solutions. The issue thus arises of how accurately this selection should be made, that is, is it worthwhile to spend considerable simulation effort on each solution in $N(\theta_k) \cap \overline{L}_k$ to make the correct selection with a high probability, or is this simulation effort better spent by making a quick selection and exploring more of the search space. Note that unlike genetic algorithms that select a set of solutions and randomness is beneficial even in the deterministic context, here it is desirable to select the single best solution. The answer to how accurate this selection needs to be is not clear and deserves further investigation.

The simulation noise is also relevant to the other main component of tabu search, namely the tabu list itself. In the deterministic context, the best solution is selected from a neighborhood $N(\theta_k)$ and thus if the search visits a solution θ_k again, the same solution will be selected from the neighborhood (assuming it is not tabu in either visit). This would imply a cycle, which is avoided by the use of the tabu list. However, in the presence of simulation noise, each time a solution is visited, a different neighborhood solution may be selected, even if the tabu list is identical. Also, if the incorrect solution is selected from the neighborhood $N(\theta_k)$, that is, another nontabu solution $\xi \in N(\theta_k) \cap \overline{L}_k$ satisfies $J(\xi) < J(\theta_{k+1})$, then it may be desirable to revisit θ_k to correct the mistake. Thus, for simulation optimization, the tabu list loses some of the appeal it has for deterministic optimization. In particular, cycles can be broken without it due to the simulation noise, and allowing the algorithm to revert back to an old solution enables the search to correct mistakes made due to the simulation noise. On the other hand, as noted earlier, tabu search has be effectively applied for simulation optimization, so it is certainly worthwhile to investigate how its performance is affected by the simulation noise and what implications this has in terms of its ideal implementation.

6 The nested partition method

Introduced by Shi and Ólafsson (2000a), the nested partition (NP) method is another metaheuristic for combinatorial optimization that is readily adapted to simulation optimization problems (Shi and Ólafsson, 2000b). The key idea behind this method lies in systematically partitioning the feasible region into subregions, evaluating the potential of each region, and then focusing the computational effort to the most promising region. This process is carried out iteratively with each partition nested within the last. The computational effectiveness of the NP method relies heavily on the partitioning, which, if carried out in a manner such that good solutions are clustered together, can reach a near optimal solution very quickly.

In the kth iteration of the NP algorithm, a region $\sigma(k) \subseteq \Theta$ is considered most promising. What this means is that the algorithm believes that this is the most likely part of the solution space to contain the optimal solution, and thus the computation effort should be concentrated in this region. As in the beginning nothing is known about the location of the optimal solution, the algorithm is initialized with $\sigma(0) = \Theta$. The most promising region is then partitioned into M subsets or subregions and the entire surrounding region $\Theta \backslash \sigma(k)$ is aggregated into one. Thus, in each iteration $M + 1$ disjoint subsets that cover the entire feasible region are considered. Each of these $M + 1$ regions is sampled using some random sampling scheme to obtain sets of solutions, and the simulated performance function values at randomly selected points are used to estimate the promising index for each region. This index determines the most promising region in the next iteration by taking the subregion scoring highest on the promising index. If the surrounding region rather than the subregion is found to have the highest promising index, the method backtracks to a previous solution corresponding to a larger region. The new most promising region is partitioned and sampled in a similar fashion. This generates a sequence of set partitions, with each partition nested within the last.

The effect of the simulation noise comes into play in the NP method in the selection of a region as the most promising region. If the performance estimates for each sample solution are noisy, the next most promising region is selected with less confidence. However, this is already a noisy selection even for deterministic problem since there are two sources of randomness:

- There is a sampling error due to a small sample of solutions being used to estimate the overall promise of what is often a large set of solutions (region).
- For each of the sample solutions, the performance is estimated using simulation, and is hence noisy.

The nested partitions method is therefore a set-based method that has an inherent noise in the selection of move, which as for genetic algorithms, appears to make it relatively insensitive to the simulation noise. Furthermore, as mentioned in Section 3, the nested partitions method includes a built-in mechanism for recovering from incorrect moves. Sometimes the simulation noise may cause the algorithm to move to the wrong subregion. However, as the search progresses the surrounding region continues to be sampled, which allows the algorithm to recover from incorrect moves through backtracking.

Regardless of the robustness of the method with regard to simulation noise, it is desirable to be able to control this noise. Ólafsson (2004) addresses this in a variant of the nested partitions method that uses statistical selection to guide the search. In this variant, the sampling of each region has two phases. First, a small sample is obtained from each region to get an initial idea of the performance variance within each region, and possibly screen out very poor regions. Then, based on these initial samples, a second set of samples is obtained from all viable regions to ensure that the correct selection of a region is made within an indifference zone with a pre-specified minimum probability. This can thus be thought of as a mechanism to control the noise in the performance estimates of each region, which guides the search. However, this only controls the overall noise and simply prescribes additional samples from each region. Some of the previously sampled solutions could be sampled again, which would improve the estimate of that solution's performance, or it could be completely different solutions. Thus, this method does not prescribe how much effort should be devoted to estimating the performance of each solution and how much should be devoted to obtaining more samples solutions. Investigating this balance is an important future research topic.

7 Making convergence statements

Research in simulation optimization has largely focused on algorithms where rigorous convergence statements can be made. However, such analysis has frequently been under either severely restrictive assumptions, or the design of the algorithm sacrifices efficiency for theoretically appealing properties. This has contributed to an unfortunate perception of a trade-off between rigor and practicality.

Much of this work in this area has focused on some type of asymptotic convergence. For example, many stochastic optimization procedures can be shown to converge to the optimal solution θ^* almost surely or with probability one (w.p.1), that is,

$$\hat{\theta}_k^* \overset{k\to\infty}{\longrightarrow} \theta^*, \quad \text{w.p.1.} \tag{8}$$

Here, $\hat{\theta}_k^*$ denotes the best solution found by the kth iteration, that is, the current estimate of the optimal solution. Other algorithms may converge in probability or in distribution. The drawback to any asymptotic analysis is that having such convergence results may be of limited practical value. It is of course somewhat reassuring to have asymptotic convergence, but since little can be inferred about what happens when the optimizer inevitably terminates in finite time, its value to a practitioner is often hard to argue.

Although most research on metaheuristics in the deterministic domain has focused on implementation and computational issues, the research has also resulted in numerous convergence proofs of these algorithms. For example, genetic algorithms have been investigated to determine how many iterations are needed before the optimal solution θ^* has been seen with a given probability, that is, the best solution found so far, $\hat{\theta}_k^*$, is equal to the optimum. In particular, Liepins (1992) shows that a genetic algorithm generates a homogeneous Markov chain, and uses this to analyze the number of iterations $k = t(1-\alpha)$ needed so that

$$\mathrm{P}\big[\hat{\theta}^*_{t(1-\alpha)} = \theta^*\big] \geqslant 1-\alpha. \tag{9}$$

Aytug et al. (1996) develop a bound on the number $t(1-\alpha)$ of iterations required by the genetic algorithm for deterministic optimization. Greenhalgh and Marshall (2000) further refine this bound and show that

$$t(1-\alpha) \leqslant \left\lceil \frac{\ln(\alpha)}{n\ln(1-\min\{(1-\mu)^{\gamma-1}(\mu/(K-1)), (\mu/(K-1))^{\gamma}\})} \right\rceil, \tag{10}$$

where $\mu > 0$ is the mutation probability, and each solution is represented as a point in the space $\{0, 1, 2, \ldots, K-1\}^{\gamma}$, which means K is the string length, for example, $K = 2$ corresponds to the traditional binary string representation. Reynolds and Gomatam (1996) also analyze genetic algorithms for deterministic optimization using a Markov chain analysis, and prove convergence of certain variations.

Note that (9) does not guarantee the quality of the estimate $\hat{\theta}^*_{t(1-\alpha)}$ of the best solution, just that it is equal to the best solution with a minimum probability. In other words, it does not make a direct statement about the performance difference $|J(\hat{\theta}^*_{t(1-\alpha)}) - J(\theta^*)|$, even in expectation or probability. This shift away from cardinal comparison is related to the concept of ordinal optimization where the analysis is solely based on ordinal comparisons (Ho et al., 1992, 2000). In this approach, the objective is relaxed from finding the optimal solution to finding a subset $\Theta_{\mathrm{G}} \subset \Theta$ of 'good enough' solutions (called goal softening). In the kth step, there is a set of selected solutions $\hat{\theta}_k^* \subset \Theta$, and the key convergence statements are made about the alignment probability $\mathrm{P}[|\Theta_{\mathrm{G}} \cap \hat{\theta}_k^*| \geqslant y]$ between those two sets. For example, if $\hat{\theta}_k^*$ is a set of $g = |\Theta_{\mathrm{G}}|$ uniformly sampled solutions, then the alignment probability is (Ho et al., 1992)

$$\mathrm{P}\big[|\Theta_{\mathrm{G}} \cap \hat{\theta}_k^*| \geqslant y\big] = \sum_{i=y}^{g} \binom{g}{i}\binom{N-g}{g-i} \Big/ \binom{N}{g}, \tag{11}$$

where $N = |\Theta|$. We note that (9) is a special case of the alignment probability with $\Theta_G = \{\theta^*\}$, $y = 1$, and $k = t(1 - \alpha)$.

Significant research has also been devoted to the convergence of tabu search in the deterministic context. In an early work, by introducing the Metropolis criterion and annealing process of simulated annealing into the general framework of tabu search, Tian et al. (1997) proposed a stochastic tabu search strategy for discrete optimization. This algorithm is provably asymptotically convergent to a global optimal. Later, Hanafi (2000) proved convergence using the observation that if the neighborhood employed is strongly connected there will be a reversible path from each solution to the every other one, which is a conjecture originally stated by Glover (1990). Glover and Hanafi (2002) provided more efficient forms of convergent tabu search, offering specific bounds, and establishing the finite convergence of tabu search for deterministic combinatorial optimization.

Shi and Ólafsson (2000a) showed that in the case of combinatorial optimization problems, the nested partition method converges with probability one to a global optimum in finite time. In their analysis, they showed that the NP algorithm generates an absorbing Markov chain and that any absorbing state corresponds to a global optimum. Thus, the algorithm may spend a certain random number of iterations in the transient states, but since the number of states is finite, it will visit an absorbing state, and hence converge, in time T, which is finite with probability one, that is, $\mathrm{P}[\hat{\theta}^*_T = \theta^*] = 1$ for a random stopping time T, with $\mathrm{P}[T < \infty] = 1$. They also use the Markov chain analysis to provide bounds on $\mathrm{E}[T]$, the expected number of iterations required for convergence.

With only a few exceptions, both asymptotic and finite-time convergences results of metaheuristics have been done for deterministic problems only. However, notable exceptions are simulated annealing and the use of global guidance systems, and in particular the nested partitions method. The simulated annealing algorithm is discussed in Chapter 20 and we do not explore it further here. With regards to the nested partitions method, Shi and Ólafsson (2000b) prove the asymptotic convergence of NP algorithm for stochastic problems. They show that the NP algorithm generates an ergodic Markov chain, and used this to develop conditions under which the NP method converges to an optimal solution with probability one, that is, under which the algorithm satisfies (8). The key to the proof is to show that the estimate of the best solution, which is the singleton region visited the most often, converges with probability one to a maximizer of the stationary distribution of the chain. Under some mild condition, this maximum corresponds to the global optimum, and it follows that for simulation optimization the NP method converges asymptotically to an optimal solution. Pichitlamken and Nelson (2003) also employ the NP method as a global guidance system. In their approach, they combine NP with statistical selection of the best, which is used to control the selection error, and local improvement heuristics, which is used to speed the search. The result is an algorithm that is both provably convergent and exhibits good performance in practice.

Despite the reassurance of asymptotic convergence, for the practitioner it would be more useful to be able to make a convergence statement once the optimizer terminates. Ideally, one would like to be able to make a statement about how close in performance our estimate $\hat{\theta}_k^*$ of the optimal solution is to the true optimal solution θ^*, that is,

$$\left|J\left(\hat{\theta}_k^*\right) - J\left(\theta^*\right)\right| \leqslant \varepsilon, \tag{12}$$

where ε is an error that is either pre-specified or calculated after the algorithm terminates after the kth iteration. This is an unreasonably strong statement to hope for in simulation optimization. On the other hand, relaxed versions of such statements are common in the simulation literature when ranking and selection algorithms are used (see Chapter 17). In particular, relaxing the statement (12) to being with an indifference zone $\varepsilon > 0$ of the optimal performance with a given minimum probability $1 - \alpha$, that is,

$$\mathrm{P}\left[\left|J\left(\hat{\theta}_k^*\right) - J\left(\theta^*\right)\right| \leqslant \varepsilon\right] \geqslant 1 - \alpha, \tag{13}$$

makes it a more reasonable goal for simulation optimization. Such convergence statements are commonly made for statistical selection, but those are only applicable when the number of alternatives is small.

Utilizing the idea of relaxing the convergence to obtaining a solution that has performance within an indifference zone of the optimum with a minimum probability, Ólafsson (2004) analyzed the finite time behavior of a variant of the nested partitions method called the two-stage nested partitions (TSNP). This variant uses a statistical selection method in each step to guide the amount of computational effort of the algorithm. They show that (13) is satisfied when the algorithm terminates at a random time $k = T$, given that statistical selection methods are used to ensure that the correct selection is made in each step within the same indifference zone ε and with a given probability $\mathrm{P}(CS)$. This probability can be calculated based on the final probability $1 - \alpha$, and the depth d of the partitioning tree, that is,

$$\mathrm{P}(CS) = \frac{(1-\alpha)^d}{\alpha^d + (1-\alpha)^d}. \tag{14}$$

The depth d is the number of partitions needed until the regions become singletons. This number is proportional to $\log(|\Theta|)$, and is therefore monotone increasing, albeit slowly, in the problem size. Ólafsson (2004) also provides simple bounds on the expected number of steps until termination, namely

$$\mathrm{E}[T] \leqslant \frac{d}{2P(CS) - 1}. \tag{15}$$

We finally note that by selecting the indifference zone small enough, that is, if θ^* is unique and

$$\varepsilon < \left| \min_{\theta \in \Theta \setminus \{\theta^*\}} J(\theta) - J\big(\theta^*\big) \right| \tag{16}$$

then (13) reduces to (9). This illustrates that these approaches are related, although for the NP for simulation optimization, the time is necessarily random and thus the bounds (15) are in terms of expectation whereas (10) is deterministic, and (13) is more general than (9) in that it provides a performance statement for the final solution when it is not the optimal solution.

Even though they are the most popular metaheuristics for simulation optimization in practice (April et al., 2003), there has been very little work done regarding the convergence of genetic algorithms when it comes to its application to stochastic problems. Thus, more work is needed to analyze both asymptotic convergence and finite time convergence properties. An interesting analysis of the output of genetic algorithm is the work done by Boesel et al. (2003b), which builds on the work by Boesel et al. (2003a) described in Section 4. As mentioned before, in order to apply deterministic genetic algorithms in stochastic setting, the authors combined the search with both multiple comparison procedure and ranking and selection procedures. Thus, in this approach the search is separated from the solution selection. Genetic algorithm is first used to generate a set of solutions, and these solutions are then further analyzed to select the one that is truly the best. This does not prove convergence, or even make a statement regarding the quality of the final solution relative to the optimal solution, but it does rigorously analyze the output generated by the genetic algorithm and accounts for the simulation noise. This analysis does, however, seem to naturally complement the analysis of Liepins (1992), Aytug et al. (1996) and Greenhalgh and Marshall (2000) and Equation (10) in particular, which could to be extended to the simulation optimization setting by accounting for the simulation noise.

Most convergence analysis for simulation optimization algorithms appears to be based on the algorithm generating a homogeneous Markov chain. This is in particular true for the various variants of the nested partitions method (Shi and Ólafsson, 2000b; Ólafsson, 2004), and allows for eloquent analysis of the method. However, the disadvantage is that to guarantee time homogeneity, independence must be enforced between iterations, and independent simulation runs must be made for solutions, even if they have been previously estimated. Due to the computational expense of simulation runs, this is inefficient (Kim and Ólafsson, 2002). On the other hand, this property may sometimes be helpful when the variance of the estimates is high and previous biased estimates have caused the algorithm to make an incorrect move. In such cases, obtaining new estimates in every step may help the algorithm to recover from wrong moves as discussed in Section 3. However, we believe further research is needed to extend existing convergence results to allow reuse of simulation runs, and to determine the proper balance that also enables the algorithm to recover from incorrect moves.

8 Future directions

Due to their prevalence and effectiveness in simulation practice, we believe that metaheuristics will continue to be an important part of simulation methodology in the future, which also makes this an important area of research. Such research should focus on the simulation aspects of the problem, that is, making the simulation a driver rather than being secondary to the optimization routine. However, as this research is carried out, it is important to take into account the concerns of practitioners, where computational efficiency is the key issue. Thus, effectiveness should never be sacrificed at the expense of theoretical insights.

In this chapter, we have outlined two areas that we believe to be of particular importance, accounting for simulation noise in guiding the search and deriving convergence results. We start with the first issue of accounting for the simulation noise in the design of the metaheuristics. Too often, metaheuristics that are designed for deterministic, most often combinatorial, optimization problems are simply applied to a simulation problem with only minimal consideration into the effects of the simulation noise. As outlined in Section 3, two of the concerns include determining if a new (candidate) solution is really better and should be selected, and recovering from incorrect moves made due to the noisy performance estimates. Fortunately, several areas of research appear to be promising. Ideas from ordinal optimization and statistical selection can be incorporated to help determine if a move should be made, which addresses the first issue. Second, the local search can be embedded into a global guidance system, such as the nested partitions method, that allows the search to recover from incorrect moves. Such combination of metaheuristics has been found effective in combinatorial optimization, and we believe that such careful consideration of the simulation noise will improve the computational performance of the metaheuristics, and hence have great practical value.

The second main area of research discussed in this chapter is providing convergence results, which we explored in Section 7. We do believe that the ability to make such convergence statements is of value, but also think that the focus should be shifted away from the traditional asymptotic convergence results to finite time behavior and goal softening. For example, rather than proving asymptotic convergence to a global optimum, it would be of more practical value to show that a solution has been obtained that is within an indifference zone of the optimal performance with a pre-specified minimum probability. This terminology stems from ranking and selection, which can be used to guide search methods in simulation optimization. As noted in Section 7, this idea is also related to the goal softening of ordinal optimization. Further research along those directions would both make existing use of metaheuristics more rigorous and may lead to insights that would improve the algorithm performance.

References

April, J., Glover, F., Kelly, J.P., Laguna, M. (2003). Practical introduction to simulation optimization. In: *Proceedings of the 2003 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 71–78.

Aytug, H., Bhattacharrya, S., Koehler, G.J. (1996). A Markov chain analysis of genetic algorithms with power of 2 cardinality alphabets. *European Journal of Operational Research* 96, 195–201.

Azadivar, F., Wang, J. (2000). Facility layout optimization using simulations and genetic algorithms. *International Journal of Production Research* 38 (17), 4369–4383.

Back, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York.

Boesel, J., Nelson, B.L., Ishii, N. (2003a). A framework for simulation optimization software. *IIE Transactions* 35 (3), 221–229.

Boesel, J., Nelson, B.L., Kim, S. (2003b). Using ranking and selection to "clean-up" after simulation optimization. *Operation Research* 51 (5), 814–825.

Dai, L. (1996). Convergence properties of ordinal comparison in the simulation of discrete event dynamic systems. *Journal of Optimization Theory & Applications* 91 (2), 363–388.

Dasgupta, D., Mcgregor, D.R. (1992). Nonstationary function optimization using the structured genetic algorithm. *Parallel Problem Solving from Nature* 2, 145–154.

Eglese, R.W. (1990). Simulated annealing: A tool for operation research. *European Journal of Operational Research* 46, 271–281.

Faccenda, J.F., Tenga, R.F. (1992). A combined simulation/optimization approach to process plant design. In: *Proceedings of the 1992 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 1256–1261.

Fleischer, M.A. (1995). Simulated annealing: Past present and future. In: *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 155–161.

Fu, M.C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing* 14 (3), 192–215.

Glover, F. (1989). Tabu Search – Part I. *ORSA Journal on Computing* 1, 190–206.

Glover, F. (1990). Tabu Search – Part II. *ORSA Journal on Computing* 2, 4–32.

Glover, F., Hanafi, S. (2002). Tabu search and finite convergence. *Discrete Applied Mathematics* 119, 3–36.

Glover, F., Kochenberger, G.A. (2003). *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston, MA.

Glover, F., Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Boston, MA.

Glover, F., Laguna, M., Marti, R. (2003). Scatter search. In: Ghosh, A., Tsutsui, S. (Eds.), *Theory and Applications of Evolutionary Computation: Recent Trends*. Springer-Verlag.

Goldberg, D.E. (1989). *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Greenhalgh, D., Marshall, S. (2000). Convergence criteria for genetic algorithm. *SIAM Journal on Computing* 30, 269–282.

Grefenstette, J.J. (1992). *Genetic Algorithm for Changing Environments*. *Parallel Problem Solving from Nature 2*. Elsevier, Amsterdam, The Netherlands.

Haddock, J., Mittenhall, J. (1992). Simulation optimization using simulated annealing. *Computer and Industrial Engineering* 22, 387–395.

Hall, J.D., Bowden, R.O. (1997). Simulation optimization by direct search: A comparative study. In: *Proceedings of the 6th International Industrial Engineering Research Conference*. IEE, Norcross, GA, pp. 298–303.

Hanafi, S. (2000). On the convergence of tabu search. *Journal of Heuristics* 7, 47–58.

Hansen, P., Mladenovic, N. (1999). An introduction to variable neighborhood search. In: Voss, S., et al. (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, Boston, MA, pp. 433–458.

Ho, Y.C., Sreenivas, R., Vakili, P. (1992). Ordinal optimization of discrete event dynamic systems. *Journal of DEDS* 2 (2), 61–88.

Ho, Y.C., Cassandras, C.G., Chen, C.-H., Dai, L. (2000). Ordinal optimization and simulation. *Journal of the Operational Research Society* 51, 490–500.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

Ishibuchi, H., Murata, T. (1996). Multi-objective genetic local search algorithm. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, pp. 119–124.

Jacobson, S.H., Sullivan, K.A., Johnson, A.W. (1998). Discrete manufacturing process design optimization using computer simulation and generalized hill-climbing algorithm. *Engineering Optimization* 31, 247–260.

Johnson, A.W., Jacobson, S.H. (2002). On the convergence of generalized hill climbing algorithms. *Discrete Applied Mathematics* 119, 37–57.

Kelly, J.P. (2002). Simulation optimization is evolving. *INFORMS Journal on Computing* 14 (3), 223–225.

Kim, J., Ólafsson, S. (2002). Two-stage NP method with inheritance. In: *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 279–284.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P. (1983). Optimization by simulated annealing. *Science* 220, 671–680.

Liepins, G.E. (1992). On global convergence of genetic algorithms. *SPIE Neutral and Stochastic Methods in Image and Signal Processing* 1766, 61–65.

Liepins, G.E., Hilliard, M.R. (1989). Genetic algorithm: Foundations and applications. *Annals of Operations Research* 21, 31–58.

Muhlenbein, H. (1997). Genetic algorithms. In: Aarts, E., Lenstra, J.K. (Eds.), *Local Search in Combinatorial Optimization*. Wiley, New York, pp. 137–172.

Ólafsson, S. (2004). Two-stage nested partitions method for stochastic optimization. *Methodology and Computing in Applied Probability* 6, 5–27.

Ólafsson, S., Kim, J. (2002). Simulation optimization. In: *Proceedings of the 2002 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 79–84.

Pichitlamken, J., Nelson, B.L. (2003). A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation* 13 (2), 108–133.

Resende, M.G.C., Ribeiro, C.C. (2003). Greedy randomized adaptive search procedures. In: Glover, A., Kochenberger, A. (Eds.), *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston, MA.

Reynolds, D., Gomatam, J. (1996). Stochastic modeling of genetic algorithms. *Artificial Intelligence* 82, 303–330.

Shi, L., Chen, C.H. (2000). A new algorithm for stochastic discrete resource allocation optimization. *Journal of Discrete Event Dynamic Systems: Theory and Applications* 10, 271–294.

Shi, L., Men, S. (2003). Optimal buffer allocation in production lines. *IIE Transactions* 35, 1–10.

Shi, L., Ólafsson, S. (2000a). Nested partitions method for global optimization. *Operations Research* 48, 390–407.

Shi, L., Ólafsson, S. (2000b). Nested partitioned method for stochastic optimization. *Methodology and Computing in Applied Probability* 2, 271–291.

Shi, L., Ólafsson, S., Chen, Q. (1999). A new hybrid optimization method. *Computers and Industrial Engineering* 36, 409–426.

Sullivan, K.A., Jacobson, S.H. (2001). A convergence analysis of generalized hill climbing algorithms. *IEEE Transaction on Automatic Control* 46 (8), 1288–1293.

Tian, P., Ma, J., Fan, Z. (1997). A stochastic tabu search strategy and its global convergence. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE Press, Piscataway, NJ, pp. 410–414.

Tompkins, G., Azadivar, F. (1995). Genetic algorithms in optimizing simulated systems. In: *Proceedings of the 1995 Winter Simulation Conference*. IEEE Press, Piscataway, NJ, pp. 757–762.

Vavak, F., Fogarty, T.C. (1996). Comparison of steady state and generational genetic algorithms for use in nonstationary environments. In: *Proceedings of IEEE International Conference on Evolutionary Computation*. IEEE Press, Piscataway, NJ, pp. 192–195.

Watson, J., Beck, J.C., Howe, A.E., Whitley, L.D. (2003). Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence* 143 (2), 189–217.

Yoshitomi, Y., Yamaguchi, R. (2003). A genetic algorithm and the Monte Carlo method for stochastic job-shop scheduling. *International Transaction of Operational Research* 10, 577–596.

Author Index

Roman numbers refer to pages on which the author (or his/her work) is mentioned. Italic numbers refer to reference pages.

Abate, J., 398, 404, 411, *411*, *412*
Acworth, P., 351, 364, *376*
Ahamed, T.P.I., 316, 317, 322, 323, *346*
Ahrens, J.H., 92, 95, *116*, *117*
Ahuja, R.K., 627, *630*
Åkesson, F., 364, *376*
Akhiezer, N.I., 102, *117*
Ala-Nissila, T., 77, *81*
Aldous, D., 116, *117*
Alexandrov, N., 546, 564, *570*
Alexopoulos, C., 199, 206, 208, 210, *222*, 459, 463, 467, 469–474, *474*, *475*
Allen, M.D., 165, *180*
Allison, P.D., 167, *178*
Alrefaei, M.H., 620, 621, *630*
Altuntas, B., 554, *573*
Anantharam, V., 328, *346*
Anderson, B., 328, 329, *348*
Anderson, T., 337, *346*, 444, *451*, 480, *499*
Andradóttir, S., 227, 236, *253*, 299, *346*, 492, *499*, 536, *570*, 609–611, *612*, 620–624, 628, 629, *630*, *631*
Andres, T.H., 63, *79*
April, J., 545, *570*, 633, 635, 644, 651, *653*
Argon, N.T., 459, 463, 469–473, *474*
Arkin, B.L., 162, *178*
Armero, C., 244, *253*
Arnold, B.C., 128, *150*
Arnold, S., 537, *574*
Arnold, S.F., 554, *573*
Asau, Y., 89, *117*
Asmussen, S., 51, 52, *52*, 114, *117*, 292, 300, 303, 313, 331–335, 337, *346*, 386, *412*
Asostsky, D.I., 351, *379*
Avellaneda, M., 275, 276, *288*
Avram, F., 328, *346*
Avramidis, A.N., 21, 26, *52*, 273, 274, 281, *288*
Aytug, H., 648, 651, *653*
Azadivar, F., 640, *653*, *654*

Babu, A.J.G., 95, *120*
Babu, G.J., 436, *452*
Back, T., 644, *653*
Baggerly, K., 316, 318, 319, *349*
Bahadur, R., 308, *346*
Bailey, R.W., 86, *117*
Balakrishnan, N., 136, *152*
Banerjee, S., 508, 509, *532*
Banks, D., 77, *81*
Banks, J., 2, 6, *18*, 160, *179*, 184, 186, *191*, 513, *532*, 538, 545, *570*, 609, *612*
Barker, E., 77, *81*
Barlow, R.E., 233, *253*
Bartlett, M.S., 213, *222*
Barton, R.R., 226, 239, *253*, 417, *451*, 537, 539, 541–543, 545, 546, 567, 570, *570–574*
Basawa, I.V., 449, *451*
Bashyam, S., 608, 610, *612*
Bassamboo, A., 334, 335, 345, 346, *346*
Basu, A.P., 165, *180*
Bauer, K.W., 529, *534*
Bayarri, M., 244, *253*
Bazaraa, M.S., 546, *571*
Bechhofer, R.E., 246, 247, *253*, 502–504, 517, 522, 524, 526, 528, 529, *532*
Beck, B., 317, 323, 325, 330, *346*
Beck, J.C., 640, *654*
Beckman, R.J., 281, *289*
Bedford, T., 146, 147, *150*, *152*, 245, *255*
Behnken, D.W., 553, *571*
Bell, S., 12, *18*
Benhamou, E., 609, *612*
Bennett, L.E., 165, *180*
Benveniste, A., 619, *630*
Beran, J., 392, 411, *412*

Berger, J.O., 227, 231, 239, 240, *253*
Berger, R.L., 160, *179*
Berk, R., 240, *254*
Bernardo, J.M., 227, 229–232, 235, 240, 241, *254*
Bertsekas, D.P., 316, *346*, 538, *571*
Besag, J., 111, 113, *117*
Best, D.J., 95, *117*
Best, N.G., 112, *118*, 238, 239, *255*, *256*
Bettonvil, B., 444, *452*
Bhatia, M., 338, *348*
Bhatnagar, S., 50, *53*, 610, *612*
Bhattacharrya, S., 648, 651, *653*
Bickel, P.J., 429, 431, *452*
Bier, V.M., 227, *253*
Biles, W.E., 548, *571*
Biller, B., 124, 134, 138, 142–144, 150, *150*
Billingsley, P., 22–24, *53*, 204, *222*, 231, *254*, 263, 264, 267, *288*, 457, 463, 464, *474*, 497, *499*
Binswanger, K., 331–333, 337, *346*
Blanchet, J., 335, *347*
Block, H.W., 141, *150*
Blomqvist, N., 200, *222*
Blouin, F., 63, *80*
Blum, L., 75, *78*
Blum, M., 75, *78*
Boender, C.G.E., 546, 568, *571*
Boesel, J., 502, 518, *532*, 627, *630*, 644, 651, *653*
Bolia, N., 327, *347*
Bollobas, B., 263, 264, *288*
Bondesson, L., 107, *117*
Booker, A.J., 541, 542, 545, 564, 567, *571*, *573*
Booth, T., 316, 318, *347*
Boots, N., 301, 334, *347*
Borkar, V., 326, *347*
Borkar, V.S., 308, 316, 317, 322–326, *346*, *347*
Bowden, R.O., 640, *653*
Bowman, R.A., 609, *612*
Box, G.E.P., 85, *117*, 538, 539, 548, 551, 553, *571*
Boyle, P., 355, *379*
Braaten, E., 367, *376*
Branke, J., 250, 252, *254*
Bratley, P., 189, 191, *191*, 368, *376*, 393, *412*, 457, *474*
Bray, T.A., 92, *120*, 127, *152*
Breiman, L., 493, *499*, 541, *571*
Brémaud, P., 610, *612*
Broadie, M., 351, 364, *376*, 608, *612*
Browne, S., 401, *412*
Bucklew, J., 301, 309, 311, 312, *350*
Bucklew, J.S., 324, *347*
Buff, R., 275, 276, *288*
Burdick, J.F., 165, *180*
Burman, D., 135, 137, *151*
Buss, A., 609, *614*

Caflisch, R.E., 107, *117*, 351, 353, 360, 361, 363, 364, *376*, *378*
Cahya, S., 546, 556, *571*, *573*
Calvin, J.M., 474, *474*, 484, 492, 496, *499*
Cao, X.R., 610, *612–614*
Cario, M.C., 135, 142–144, *151*, 167, *180*
Carmeli, M., 174, *179*
Carnahan, J.V., 546, *574*
Carson, J.S., 513, *532*, 609, *612*
Carson II, J.S., 2, 6, *18*, 160, *179*, 184, 186, *191*, 538, 545, *570*
Casella, G., 160, *179*, 386, *412*
Cassandras, C.G., 610, 612, *613*, *616*, 641, 648, *654*
Castellana, J.V., 216, *222*
Castillo, E., 128, *150*
Chambers, J.M., 87, *117*
Champoux, Y., 330, *349*
Chandra, A., 333, *349*
Chang, C.S., 292, 298, 303, 317, 323, 325, 330, *347*
Charlier, C.V.L., 134, *151*
Charnes, A., 546, *571*
Chatfield, C., 201, *222*
Chen, C.-H., 227, 235, 246, 249, 250, 252, *254*, *255*, 531, *532*, 641, 648, *654*
Chen, E., 183, 184, *191*
Chen, E.J., 57, 77, *80*
Chen, H., 227, 250, *254*, 384, *412*, 618, *630*
Chen, H.-C., 89, *117*, 227, *254*, 531, *532*
Chen, H.-F., 609, *616*
Chen, J., 608, *613*
Chen, Q., 642, *654*
Chen, W., 542, *572*
Cheng, R.C.H., 34, *53*, 95, *117*, 227, 240–242, *254*, 417, 444, 445, *452*, 570, *571*
Chernick, M.R., 417, *452*
Chick, S.E., 226, 227, 233, 235, 236, 239–242, 244, 245, 247–250, 252, 253, *254–256*, 531, *532*, 570, *571*
Chien, C., 460–462, 467, *474*
Chistyakov, V.P., 331, *347*
Chong, E.K.P., 609, *613*
Chow, Y.S., 205, *222*
Cieslak, M., 368, *378*
Çinlar, E., 159, 166, *179*, 186, *191*
Clark, D.S., 610, *615*, 619, *631*
Clemen, R.T., 130, *151*
Clewlow, L.J., 608, *613*
Cochran, W.G., 381, *412*
Cogliano, V.J., 609, *615*

Cohen, J.W., 383, 398, *412*
Collamore, J.F., 313–315, *347*
Compagner, A., 74, *81*
Conover, W.J., 281, *289*
Conway, J.H., 62, *78*
Cooke, R.M., 146, 147, *150–152*
Cooper, R.B., 383, 400, *412*
Cooper, W.W., 546, *571*
Cornell, J.A., 538, *572*
Côté, S., 64, 65, *79*
Couture, R., 62, 63, 67, 68, 71, 72, *78–81*
Cox, D., 158, 170, 171, *179*, 316, 318, 319, *349*, 478, *499*, 551, *571*
Cox, G.M., 381, *412*
Craig, A.T., 207, 208, *222*
Craig, P., 227, 244, *254*
Cramér, H., 486, *499*
Crane, M., 478, *499*
Cranley, R., 371, *376*
Craven, P., 541, *571*
Crawford, M.M., 165, *179*
Cressie, N.A., 241, 243, *254*
Currie, C.S.M., 227, 240, *254*, 570, *571*

Dabrowski, A., 317, 323, 325, 330, *346*
Dai, J., 328, *346*
Dai, L., 250, *254*, 610, *613*, 641, 648, *653*, *654*
Daily, O.P., 165, *180*
Dalal, S.R., 246, *254*, 523, *532*
Damerdji, H., 469, 474, *475*, 519, 523, 524, *532*
Dasgupta, D., 643, *653*
David, H.A., 169, *179*
Davis, J.R., 134–136, *153*
Davis, P., 352, *376*
Davison, A.C., 417, 422, 426, 428, 443, 451, *452*
De Boer, P.T., 316, 319, 321, *347*
de Boor, C., 541, *571*
de Finetti, B., 227, 229, 231, 232, *254*
de Freitas, N., 370, *376*
de Groot, M.H., 227, 233, 234, 247, *254*
DeBrota, D.J., 135, *151*
Deheuvels, P., 132, *151*
Deift, P., 174, *179*
Dekker, R., 548, *573*
del Castillo, E., 546, 555, 556, *571*, *573*
Delampady, M., 239, *253*
Dembeck, D.C., 370, *376*
Dembo, A., 293, 308, 309, *347*, 624, *630*
Deng, L.-Y., 63–65, *78*
Dennis, J.E., 541, 564, *571*
Dennis Jr., J.E., 546, 564, *570*
Derflinger, G., 83, 98, *119*, 133, *151*
Desai, P., 316, 318, *347*

Deutsch, S., 136, *153*
Devroye, L., 83, 86–88, 92, 94, 95, 97–101, 103–106, 109, *117*, *118*, 132, *151*, 173, *179*
Dickman, K., 105, *118*
Dierckx, P., 541, 565, *571*
Dieter, U., 92, 95, *116*, *117*
Dieudonné, J.A., 31, *53*
Dittus, R.S., 135, *151*
Dmochowski, J., 240, *254*
Donohue, J.M., 554, *571*
Doucet, A., 370, *376*
Draper, D., 226, 236, 239, *254*
Draper, N.R., 538, 553, 555, *571*, *572*
Dray, J., 77, *81*
Dudewicz, E.J., 246, *254*, 509, 526, *532*, *533*
Dufresne, D., 327, *350*
Dugué, D., 101, *118*
Dupuis, P., 326, *347*
Durbin, J., 445, *452*
Durrett, R., 199, 202, 203, *222*, 269, *288*
Dussault, J.P., 610, *613*

Eagle, D.J., 71, *81*
Edelman, A., 176, 177, *179*
Edgeworth, F.Y., 134, *151*
Edwards, A.W.F., 230, *254*
Edwards, E.B., 165, *180*
Efron, B., 417, 421, 427, 430, 451, *452*, 486, *499*
Eglese, R.W., 637, *653*
Eichenauer-Herrmann, J., 75, *78*
Eick, S.G., 411, *412*
Elmaghraby, S., 528, *532*
Embrechts, P., 332, *347*
Epanechnikov, V.A., 213, *222*
Ergun, Ö., 627, *630*
Ermoliev, Y.M., 621, *631*
Escobar, L.A., 157, 167, 168, *179*
Ethier, S.N., 494, *499*
Eubank, R.L., 541, *572*
Evans, M., 98, *118*, 237, *255*

Fabian, V., 515, 516, *533*
Faccenda, J.F., 640, *653*
Fan, Z., 649, *654*
Fang, K.-T., 134, *151*, 337, *347*, 351, 360, 361, 375, *379*
Farin, G., 130, *151*
Faure, H., 353, 368, 369, 373, *376*
Feast, G.M., 95, *117*
Feller, W., 104, *118*
Fendick, K.W., 408, *412*
Fill, J., 106, 116, *117*, *118*

Fincke, U., 62, *79*
Finger, C., 338, *348*
Fisher, M.J., 208, *222*
Fishman, G.S., 2, *18*, 62, 64, 75, *79*, 172, *179*, 194, 209, 210, *222*, 278, *288*, 316, 319, *347*, 393, *412*, 478, *499*
Fishwick, P.A., 2, *18*
Flajolet, P., 84, *118*
Fleischer, M.A., 637, *653*
Floudas, C.A., 546, *572*
Fogarty, T.C., 625, *631*, 643, *654*
Foley, R.D., 463, 464, 474, *475*
Fournié, E., 609, *613*
Fox, B.L., 107, *118*, 189, 191, *191*, 363, 368, *376*, *377*, 393, *412*, 457, *474*, 621, *630*
Frank, M., 554, *573*
Frank, P.D., 541, 564, *571*
Franke, R., 543, *572*
Frater, M., 328–330, *347*, *348*
Frecker, M., 546, *574*
Freedman, D.A., 429, 431, 449, *452*
Friedman, C., 275, 276, *288*
Friedman, J.H., 541, *572*
Fu, B.-R., 610, *615*
Fu, M.C., 25, 50, *53*, 252, *255*, 327, *350*, 537, 538, 545, *572*, 599, 608–610, 612, *612–616*, 633, 634, *653*
Fujimoto, R.M., 3, *18*
Fushimi, M., 70, *79*, 371, 372, *378*

Gaivoronski, A., 610, *614*
Gamba, R., 275, 276, *288*
Gelatt Jr., C.D., 637, *654*
Gelfand, A.E., 113, *118*
Gelfand, S.B., 621, *630*
Gelman, A., 127, 129, *151*
Geman, D., 111, 113, *118*, 128, *151*
Geman, S., 111, 113, *118*, 128, *151*
Gentle, J.E., 62, 75, *79*
George, E.I., 241, *255*
Ghosh, M., 436, *452*
Ghosh, S., 135, 138, 144–146, 149, *151*, 176, *179*
Gilks, W.R., 97, 112, *118*, 127, *151*, 172, *179*, 226, 231, 237–239, *255*, *256*, 446, *452*
Girault, M., 101, *118*
Giroux, N., 609, *615*
Glasscock, N.F., 134–136, *152*, *153*
Glasserman, P., 2, *18*, 29, 31, 40, *53*, 260, 275–278, *288*, 292, 299, 327, 329, 335–338, 340, 341, 344, 345, *347*, *348*, 351, 364, *376*, 606, 608–610, *612*, *614*
Glover, F., 545, *570*, *572*, 633, 635, 636, 638, 639, 644, 645, 649, 651, *653*
Glynn, P., 114, *117*, 226, 227, 237, *255*, 292, 298, 304, 316, 318, 327, 335, *347*, *348*, *350*
Glynn, P.W., 7, *18*, 22, 25, 27, 34, 44, 45, 50, *53*, 204, 209, 210, 220, 221, *222*, 236, *253*, 262, 271, 274–276, 278, 287, *288*, *289*, 383, 386, 393, 408, 409, *412*, 457, 460, 462, 463, 474, *475*, 483–486, 488, 490, 492, 493, 496, *499*, *500*, 513, 524, *532*, *533*, 609, 611, *614*, *615*
Goel, A.L., 543, *574*
Goldberg, D.E., 638, 643, *653*
Goldberg, M.S., 145, *152*
Goldie, C.M., 105, *118*
Goldsman, D., 123, *152*, 199, 210, *222*, 246, 247, 250, *255*, *256*, 458–474, *474*, *475*, 502, 508, 517, 518, 520, 526–529, *532–534*
Goldsman, D.M., 246, 247, *253*
Goldstein, M., 227, 244, *254*
Gomatam, J., 648, *654*
Gong, W.-B., 610, *614*, 620, *630*
Gordon, N., 370, *376*
Goresky, M., 67, 68, *79*
Goyal, A., 292, 304, *348*
Grandchamp, N., 275, 276, *288*
Granger-Piché, J., 75–77, *79*
Grassberger, P., 292, *348*
Grassman, W.K., 393, *412*
Green, P., 240, *256*
Green, P.J., 113, 116, *117*, *119*
Greenhalgh, D., 648, 651, *653*
Grefenstette, J.J., 643, *653*
Gross, D., 157, *179*, 200, 208, *222*, 244, *255*
Grossman, I.E., 546, *572*
Grübel, R., 105, *118*
Gupta, G., 338, *348*
Gupta, S.S., 248, *255*, 503, *533*
Gürkan, G., 610, *614*
Gutjahr, W.J., 621, *631*
Guttman, I., 553, 555, *572*
Györfi, L., 132, *151*, 216, *222*, *223*

Haas, P.J., 7, *18*, 50, *53*, 490, *500*
Haber, S., 366, *377*
Haddock, J., 417, *452*, 640, *653*
Häggström, O., 109, *119*
Halfin, S., 408, 410, *412*
Hall, J.D., 640, *653*
Hall, P., 417, 434, 451, *452*
Hall, W.J., 523, *532*
Halton, J.H., 353, *377*
Hamada, M.S., 381, *413*
Hammond, J.L., 144, *152*
Hanafi, S., 649, *653*
Hannan, E.J., 140, *151*

Hansen, E., 112, *119*
Hansen, P., 638, *653*
Härdle, W., 211, *222*
Hardy, R.L., 543, *572*
Harper, A.M., 165, *180*
Harris, C.A., 167, *180*
Harris, C.M., 157, *179*, 200, *222*
Harrison, J.M., 4, 12, *18*, 403, *412*
Hartmann, M., 515, *533*
Hasenbein, J., 328, *346*
Has'minskii, R.Z., 480, *500*
Hastings, W.K., 109, *119*, 172, *179*
Healy, K.J., 608, *613*, 618, *631*
Heckert, A., 77, *81*
Heidelberger, P., 34, *53*, 209, 210, *222*, 278, *288*, 292, 293, 298, 303–305, 317, 321, 323, 325, 327, 328, 330, 335–337, 340, 341, 344, 345, *346–348*, 486, *500*, 610, *614*
Heidergott, B., 608, 609, 611, *614*
Heine, G.W., 621, *630*
Hellekalek, P., 58, 71, 75, 76, *79*
Henderson, S.G., 13, *18*, 20, 25, 27, 50, *53*, 135, 138, 144, 145, 149, *151*, 162, 176, *179*, 220, 221, *222*, 227, 237, *256*, 287, *288*, 474, *475*, 488, 492, 493, *500*
Herrmann, E., 75, *78*
Hesterberg, T.C., 276, 277, 288, *288*
Heyde, C., 344, *348*
Heyman, D.P., 299, *346*
Hickernell, F.J., 351, 353, 360, 366, 373–375, *377*
Hill, J.R., 123, *152*
Hill, R.R., 148, 149, *151*
Hill, S.D., 610, *613*
Hilliard, M.R., 638, *654*
Hinkley, D.V., 417, 422, 426, 428, 443, 451, *452*
Hjorth, J.S.U., 416, 422, 426, 428, 430, 440, *452*
Ho, Y.-C., 610, *613*, *614*, 620, *630*, 641, 648, *653*, *654*
Hoaglin, D., 382, *413*
Hochberg, Y., 502, 507, 510, *533*
Hoeffding, W., 359, *377*
Hogg, R.V., 207, 208, *222*
Hojgaard, B., 332, 333, 337, *346*
Holden, L., 112, *119*
Holland, J.H., 643, *654*
Holland, W., 34, *53*, 241, 242, *254*
Hollick, L.J., 570, *573*
Homem-de-Mello, T., 610, 612, *614*, *616*, 618, 621, *631*
Hong, H., 373, 374, *377*
Hong, H.S., 366, *377*
Hong, L.J., 515, *533*, 620, 621, *631*
Hörmann, W., 83, 95, 98, *119*, 127, 133, *151*
Horn, R.A., 177, *179*
Hotelling, H., 139, *152*
Houck, E.C., 554, *571*
Howe, A.E., 640, *654*
Howell, W.C., 609, 612, *614*, *615*
Hsu, J.C., 502, 507, *533*
Hu, J.-Q., 25, *53*, 252, *255*, 608–610, *613*, *614*
Huang, Z., 334, 340, 341, 344, *348*
Huffman, D., 89, *119*
Hugus, D.K., 141, *152*
Hull, J.C., 608, 609, *614*
Hussain, M.F., 543, *572*

Ibragimov, I.A., 480, *500*
Iglehart, D.L., 44, *53*, 204, *222*, 292, *348*, 457, 463, *475*, 478, 483, 484, *499*, *500*, 513, *533*
Ignatiouk-Robert, I., 328, *348*
Ignatyuk, I., 328, *348*
Inoue, K., 226, 227, 235, 245, 247–250, 252, *254*, *255*, 531, *532*
Inskip, H., 239, *256*
Irizarry, M.A., 34, *53*
Isham, V., 158, *179*
Ishibuchi, H., 643, *654*
Ishii, N., 644, 651, *653*

Jackson, D., 103, *119*
Jacobson, S.H., 609, *614*, 639, *654*
Jain, R.K., 2, *18*
Jamison, S.A., 167, *180*
Jarner, S.F., 112, *119*
Jaynes, E.T., 231, *255*
Jeffreys, H., 230, *255*
Jennison, C., 515, 517, 526, *533*
Jiang, T., 284, *288*, 375, *377*
Jin, R., 542, *572*
Joe, H., 130, 134, *152*
Joe, S., 364–366, 372, *379*
Johns, M.V., 451, *452*
Johnson, A.W., 639, *654*
Johnson, C.R., 177, *179*
Johnson, M.A., 165, *179*
Johnson, M.E., 133–136, *152*
Johnson, N.L., 98, *119*, 128, 134–136, *152*
Johnstone, I.M., 515, 526, *533*
Joines, J.A., 467, 474, *475*
Jones, M.C., 27, *53*, 133, *153*, 213, *222*, *223*
Jones, O.D., 445, *452*
Jong-Dev, K., 139, *152*
Joshi, S., 556, *572*
Joshi, S.B., 543, *572*

Juneja, S., 292, 294, 298–301, 303, 308, 316, 317, 322–327, 329–331, 333–335, 337, 345, 346, *346–349*

Kabak, I., 478, *500*
Kachitvichyanukul, V., 95, *119*
Kahneman, D., 229, 232, *255*
Kang, K., 458, 465, 466, 468, 472, 473, *475*
Kang, W., 345, *348*, *349*
Kankaala, K., 77, *81*
Kanter, M., 87, *119*
Kapuscinski, R., 608, *614*
Karandikar, R., 331, 334, *349*
Karlin, S., 401, *412*, 493, *500*
Karr, A.F., 195, 196, 219, *222*
Kass, R.E., 230, 231, *255*
Keane, M.S., 94, *119*
Keller, A., 355, 367, 371, *377*
Kelly, F.P., 384, *412*
Kelly, J.P., 545, *570*, *572*, 633–635, 641, 644, 651, *653*, *654*
Kelton, W.D., 2, 6, *18*, 28, *53*, 57, 62, 75, 77, *79*, *80*, 125, 127, 145, *152*, 160, 161, *179*, 182–184, 186, 189, *191*, 194, 207, 208, *222*, *223*, 232, 235, 239, *255*, 260, 266, *288*, 362, *377*, 456, 457, *475*, 502, 513, *533*, 538, 545, 554, 557, *572*, 598, 609, *615*
Kennedy, M., 227, 244, *255*, *256*
Kherani, A.A., 308, 317, 324–326, *347*
Khuri, A.I., 538, 555, *572*, *573*
Kiefer, J., 610, *615*, 619, *631*
Kim, J., 633, 651, *654*
Kim, S., 651, *653*
Kim, S.-H., 246, 247, 250, 252, *255*, 468, 473, *475*, 502, 515–518, 520, 524, 527, *532*, *533*, 627, *630*
Kim, Y.B., 417, *452*
Kimball, A.W., 510, *533*
Kirkpatrick, S., 637, *654*
Klapper, A., 67, 68, *79*
Kleijnen, J.P.C., 241, 244, *256*, 444, *452*, 538, 542, 545, 548, 551, 554, 570, *572*, *574*
Kleywegt, A.J., 618, *631*
Kloeden, P.E., 402, *412*
Kluppelberg, C., 332, *347*
Knott, M., 445, *452*
Knuth, D.E., 56, 60–64, 67, 70, 75, 76, *79*, 84, *119*, 172, *179*
Kochenberger, G.A., 636, *653*
Koehler, G.J., 648, 651, *653*
Koenig, L.W., 207, *223*, 521, *533*
Kohrt, K.D., 95, *117*
Kollig, T., 355, 367, 371, *377*
Kollman, C., 316, 318, 319, *349*
Kolmogorov, A.N., 544, *572*
Kontoyiannis, I., 324, *349*
Koopman, J.S., 227, 244, *254*
Korobov, N.M., 353, 357, 364, *377*
Korte, J.J., 541, *574*
Kostlan, E., 176, 177, *179*
Kotz, S., 87, 98, *119*, 128, 134, 136, *151*, *152*, 337, 344, *347*, *349*
Kou, S., 299, 329, 344, *348*
Kozubowski, T., 344, *349*
Kraan, B., 245, *255*
Krein, M.G., 102, *119*
Kreyszig, E., 263, 265, 269, *288*
Kroese, D., 327, 330, 332, 334, *346*, *349*
Kroese, D.P., 319, 320, *347*, *350*
Kruk, L., 275, 276, *288*
Kruskal, W., 144, *152*
Kuhl, M.E., 34, *53*, 165, *179*, *180*, 190, *191*
Kuo, F.Y., 366, *379*
Kurita, Y., 72–74, *80*
Kurowicka, D., 146, 147, *152*
Kurtz, T.G., 494, *499*
Kushner, H.J., 323, *349*, 610, *615*, 619, *631*

Labrecque, D., 610, *613*
Lada, E.K., 34, *53*, 467, 474, *475*
Laguna, M., 545, *572*, 633, 635, 638, 639, 644, 645, 651, *653*
Laha, R.G., 101, *119*
Lal, R., 95, *120*, 138, *153*
Lancaster, H.O., 143, *152*
Langberg, N.A., 141, *150*
Laplace, P.S., 230, 232, *255*
Larcher, G., 58, 71, *79*
Larson, R.C., 244, *255*
Lasry, J.M., 609, *613*
Lavenberg, S.S., 260, *288*, 304, *348*
Law, A.M., 2, 6, *18*, 28, *53*, 57, 62, 75, *79*, 125, 127, 145, *152*, 160, 161, *179*, 184, 186, 189, *191*, 194, 207, 208, *222*, *223*, 232, 235, 239, *255*, 260, 266, *288*, 362, *377*, 456, 457, *475*, 502, 513, 521, *533*, 538, 545, 554, *572*, *573*, 598, 609, *615*
Lawless, J.F., 167, *179*
Le Minh, D., 95, *119*
Leadbetter, M.R., 216, *222*
Lebuchoux, J., 609, *613*
L'Ecuyer, P., 50, *53*, 56–58, 62–72, 74–78, *78–81*, 183, 184, *191*, 330, *349*, 354, 355, 358, 361, 362, 364–366, 371–373, 375, *377*, *378*, 609–611, *613*, *615*, *616*, 628, *631*
Lee, S., 165, *179*
Lee, S.-H., 227, 237, *255*
Leeb, H., 60, *80*
Leemis, L.M., 161, 162, 171, *178*, *179*, 190, *191*

Lehmann, E.L., 386, *412*, 523, *533*
Lehoczy, J.P., 364, *376*
Lehtonen, T., 313, *349*
Leigh, S., 77, *81*
Lemieux, C., 58, 63, 71, 72, *80*, 354, 355, 358, 360–362, 364–366, 368, 371–373, 375, *377*, *378*
Lennon, T., 328, 329, *348*
Letac, G., 111, *119*
Levenson, M., 77, *81*
Levitan, Y.L., 368, *379*
Lewis, P.A.W., 104, 105, *119*, 141, *152*, 167, 170, *179*, 486, *500*
Lewis, R.M., 546, 564, *570*
Leydold, J., 83, 98, *119*, *120*, 127, 133, *151*, *152*
Li, J., 292, 338, 344, 345, *348*
Li, S., 610, *614*
Li, S.T., 144, *152*
Liepins, G.E., 638, 648, 651, *654*
Ligetti, C., 546, *574*
Lin, D.K.J., 63, *78*, 553, *572*
Lin, G.D., 102, *120*
Lin, J., 227, *254*
Lindley, D.V., 200, *223*, 227, 239, *255*
Linnik, Yu.V., 87, *120*
Lions, P.L., 609, *613*
Liu, J.S., 110, *120*
Liu, R., 375, *378*
Liu, T., 546, *574*
Livny, M., 124, *152*
Loh, W.L., 281, *288*
Loh, W.W., 260, 273, 274, 286, *288*
Luenberger, D., 315, *349*
Lurie, P.M., 145, *152*
Lüscher, M., 64, *80*
Lütkepohl, H.L., 140, *152*
Luttmer, K., 368, *378*

Ma, J., 649, *654*
Mackenzie, G.R., 145, *152*
Maclaren, M.D., 92, *120*
Macready, W.G., 625, *631*
Mallik, A.K., 449, *451*
Mallows, C.L., 87, *117*
Malyshev, V., 328, *348*
Mannor, S., 319, *347*
Marcus, R., 260, 287, *289*
Marcus, S.I., 50, *53*, 610, *612*
Margolin, B.H., 554, *574*
Marron, J.S., 213, *222*
Marsaglia, G., 65, 67, 68, 75, 77, *80*, 92, 95, *120*, 127, *152*, 175, *179*
Marshall, S., 648, 651, *653*
Marshall, W., 520, *533*
Marshall, W.S., 246, 247, 250, *255*
Marti, R., 639, *653*
Martin, B., 609, *615*
Martin, D.L., 165, *180*
Masi, D.M.B., 208, *222*
Masry, E., 216–219, *222*, *223*
Massey, W.A., 411, *412*
Másson, E., 543, *573*
Mata, F., 529, *533*
Matejcik, F.J., 508, 514, *534*
Mathé, P., 281, *289*
Matoušek, J., 353, 373, *378*
Matsumoto, M., 64, 70, 72–74, 77, *80*, *81*
Mauery, T.M., 541, *574*
McAllister, C.D., 554, *573*
McClure, D., 111, *118*
McComas, M.G., 545, *573*
McCormick, W.P., 449, *451*
McCulloch, R.E., 241, *255*
McDonald, D., 317, 323, 325, 330, *346*
McGrath, M.F., 244, *255*
Mcgregor, D.R., 643, *653*
McKay, M.D., 281, *289*
McKenzie, E., 141, *152*
McLeish, D.L., 611, *615*
Meckesheimer, M., 541, 542, 545, 567, *573*
Meeker, W.Q., 157, 167, 168, *179*
Meeuwissen, A.M.H., 146, *152*
Mehta, M.L., 174, *179*
Meketon, M.S., 460, 462–464, 468, 469, *475*, 486, *500*
Melamed, B., 123, 124, 147, *152*, 460–462, 467, *474*, 612, *613*, *616*
Meliani, L., 371, 375, *377*
Men, S., 642, *654*
Mendel, M.B., 233, *253*, *254*
Merino, S., 344, 346, *349*
Merton, R., 338, *349*
Métivier, M., 619, *630*
Metropolis, N., 109, *120*, 172, *179*
Meyn, S., 324, *349*, 393, *412*, 485, *500*
Meyn, S.P., 12, 13, *18*, 37, 39, 42, 44, 46, 47, 49, 50, *53*, 112, *120*, 325, 326, *347*
Miescke, K.J., 248, *255*
Mikosch, T., 332, *347*
Miller, J.F., 625, *631*
Miller, J.O., 167, *180*, 528, 529, *534*
Miller, R.G., 486, *500*
Mira, A., 116, *120*
Mirka, G.A., 134–136, *152*, *153*
Miró-Quesada, G., 555, *573*
Mistree, F., 541, *574*
Mitchell, T.J., 244, *256*, 541, 542, *573*
Mittenhall, J., 640, *653*

Mitter, S.K., 621, *630*
Mladenovic, N., 638, *653*
Moeller, T.L., 260, *288*
Moeschberger, M.L., 169, *179*
Moller, J., 116, *120*
Monro, S., 610, *615*, 619, *631*
Montgomery, D.C., 381, *412*, 538, 546, 551, 553, 554, 566, *573*
Morohosi, H., 371, 372, *378*
Morokoff, W.J., 107, *117*, 345, *349*, 351, 353, 360, 361, 363, 364, *376*, *378*
Morris, M.D., 541, 542, 553, *573*
Morris, S.B., 172, *179*
Morse, N., 528, *532*
Moskowitz, B., 363, *376*
Muhlenbein, H., 638, *654*
Mukai, H., 620, *631*
Mukhopadhyay, N., 523–525, *534*
Müller, M.E., 85, *117*
Murata, T., 643, *654*
Murdoch, D.J., 116, *119*, *120*
Myers, R.H., 538, 546, 553–555, 570, *571*, *573*

Nadas, A., 205, *223*
Nadler, W., 292, *348*
Nahmias, S., 234, *256*
Nakayama, M.K., 305, *349*, 474, *474*, 496, *499*, 519, 523, 524, *532*
Naylor, J., 237, *256*
Nechvatal, J., 77, *81*
Neddermeijer, H.G., 548, *573*
Neininger, R., 106, *118*
Nelsen, R.B., 88, *120*, 130, 137, 139, *152*
Nelson, B.L., 2, 4, 6, 7, *18*, 123, 124, 134, 135, 138, 142–144, 150, *150*, *151*, *153*, 157, 160, 167, *179*, *180*, 184, 186, *191*, 227, 246, 247, 250, 252, *255*, *256*, 260, 276, 277, *288*, *289*, 502, 508, 513–518, 520, 524, 526–529, *532–534*, 538, 545, *570*, 609, *612*, 620, 621, 627, *630*, *631*, 642, 644, 649, 651, *653*, *654*
Nelson, W.B., 156, *180*
Newman, J., 275, 276, *288*
Ney, P., 311, *349*
Ng, K., 337, *347*
Ng, K.W., 134, *151*
Ng, S.-H., 227, 236, 241, 242, *256*
Nicol, D., 513, *532*
Nicol, D.M., 2, 6, *18*, 160, *179*, 184, 186, *191*, 538, 545, *570*, 609, *612*
Nicola, V., 292, 304, 305, 316, 321, 327, 330, *347–349*
Nicolai, R.P., 548, *573*
Niederreiter, H., 56, 58, 69–71, 73, 75, *80*, 352–354, 364–368, 376, *377*, *378*
Nijenhuis, A., 173, *180*
Ninomiya, S., 351, *378*
Nishimura, T., 70, 73, 74, 77, *80*, *81*
Norgaard, A., 450, *452*
Norkin, V.I., 621, *631*
Notz, W.I., 226, 227, 241, 243, 244, *256*, 538, 542, 570, *574*
Nozari, A., 554, *573*
Nyefeler, M., 344, 346, *349*
Nyrhinen, H., 313, *349*

Oakes, D., 170, 171, *179*
Oakley, J., 227, 244, *256*
O'Brien, G.L., 94, *119*
O'Hagan, A., 227, 240, 244, *255*, *256*
Ólafsson, S., 633, 639, 641, 642, 646, 647, 649–651, *654*
Olkin, I., 175, *179*
O'Neill, P.D., 244, *256*
Orav, E.J., 486, *500*
Ord, J.K., 139, *153*
Orlin, J.B., 627, *630*
Ostrovskii, I.V., 87, *119*
Ott, T., 299, *346*
Owen, A.B., 107, *117*, 281–283, 285, 288, *289*, 351, 354, 355, 359, 360, 373–375, *376–378*
Özge, A.Y., 610, *614*

Pabst, M.R., 139, *152*
Page, T.W., 123, *153*
Pakes, A., 335, *349*
Panayiotou, C.G., 612, *613*, *615*, *616*
Panneton, F., 70–72, 74, 77, *80*, *81*
Papadimitriou, C.H., 383, *412*
Papageorgiou, A., 351, 364, *378*
Pardalos, P.M., 546, *572*
Parekh, S., 321, 329, *349*
Parr, W.C., 436, *452*
Parrish, R.S., 136, *153*
Parzen, E., 132, *153*, 218, *223*
Paskov, S., 351, 354, 360, *378*
Patera, A.T., 538, *574*
Patterson, T.N.L., 371, *376*
Paulson, E., 505, 526, 527, *534*
Pearson, K., 134, *153*
Pegden, C.D., 554, *573*
Pericchi, L.R., 239, 240, *253*
Perng, S.K., 525, *534*
Perron, G., 610, *615*
Peterson, J.J., 546, 556, *573*

Pflug, G.Ch., 608, 609, 611, *614*, *615*, 619, 621, *631*
Picard, R., 316, 318, 319, *349*
Pichitlamken, J., 502, *534*, 627, *631*, 642, 649, *654*
Piersma, N., 548, *573*
Plambeck, E.L., 610, *615*
Platen, E., 402, *412*
Podgorski, K., 344, *349*
Pohst, M., 62, *79*
Potoradi, J., 554, *573*
Prakasa Rao, B.L.S., 27, *53*
Priouret, P., 619, *630*
Pritsker, A.A.B., 165, *180*
Propp, J.G., 109, 114, *120*, 370, *379*
Prudius, A.A., 620, 621, 623, 628, 629, *631*
Psihogios, J.P., 134–136, *153*
Puhalskii, A.A., 410, *412*
Punnen, A.P., 627, *630*

Rabinowitz, P., 352, *376*
Rachev, S.T., 85, *120*
Raftery, A.E., 239, *256*
Ramadge, P.J., 609, *613*
Ramberg, J.S., 136, *153*, 570, *573*
Ramsay, T., 189, *191*
Randhawa, R.S., 299, 329, *349*
Rao, R.R., 308, *346*
Rea, C.B., 417, *452*
Reidys, C.M., 625, *631*
Reilly, C.H., 148, 149, *151*, 528, 529, *534*
Reilly, T., 130, *151*
Reiman, M.I., 410, *412*, 611, *615*
Rényi, A., 104, *120*
Resende, M.G.C., 638, *654*
Resnick, S.I., 157, 159, *180*
Reust, J.S., 165, *180*
Reynolds, D., 648, *654*
Rezstov, A., 366, *379*
Ribeiro, C.C., 638, *654*
Rice, J.A., 43, *53*
Richardson, S., 172, *179*, 226, 231, 237, 238, 240, *255*, *256*, 446, *452*
Richtmyer, R.D., 352, *379*
Rigdon, S.E., 165, *180*
Riley, G.F., 612, *616*
Rinooy Kan, A.H.G., 546, 568, *571*
Rinott, Y., 250, *256*, 511, 514, *534*
Robbins, H., 205, *222*, 524, *534*, 610, *615*, 619, *631*
Roberts, G.O., 116, *120*
Roberts, J.P., 165, *180*
Roberts, S.D., 135, *151*
Robinson, P.M., 216, *223*
Robinson, S.M., 610, *614*, *615*, 618, *631*
Robinson, W.D., 71, *81*
Rollins, S., 611, *615*
Rosenblatt, M., 128, 132, *153*, 204, 216, *223*
Rosenbluth, A.W., 109, *120*, 172, *179*
Rosenbluth, M.N., 109, *120*, 172, *179*
Rösler, U., 105, *120*
Ross, S.M., 156, 159, 172, *180*, 530, *534*
Rothrock, L., 546, *574*
Rougier, J., 227, 244, *254*
Royden, H.L., 297, *349*
Rubinstein, R.Y., 260, 287, *289*, 292, 300, 316, 319–321, 334, *346*, *347*, *349*, *350*, 608–611, *613*, *615*, 618, *631*
Rukhin, A., 77, *81*
Rüschendorf, L., 105, *120*
Ruszczyński, A., 621, *631*

Sacks, J., 244, *256*, 541, *573*
Sadowski, R.P., 557, *572*
Sadowsky, J.S., 298, 300, 301, 309, 311, 312, *350*
Saheb, N., 84, *118*
Saksena, V.R., 408, *412*
Samorodnitsky, G., 392, *412*
Sanchez, P.J., 570, *573*
Sanchez, S., 570, *573*
Santner, T.J., 226, 227, 241, 243, 244, 246, 247, *253*, *256*, 502, 517, 528, 529, *532*, 538, 542, 570, *574*
Sarabia, J.M., 128, *150*
Sargent, R.G., 458, 473, *475*, 543, 545, *572*, *574*
Sato, K., 106, *120*
Savage, L.J., 227, 229, 232, *256*
Schechtman, E., 451, *452*
Scheffé, H., 112, *120*
Scherbakov, V., 328, *348*
Schmeiser, B.W., 95, *119*, *120*, 136, 138, *153*, 227, *254*, *256*, 263, *289*, 457, 460, 462, 467–469, *475*, 618, *630*
Schmidt, C., 250, 252, *254*
Schoenberg, F.P., 156, *180*
Schrage, L.E., 189, 191, *191*, 393, *412*, 457, *474*
Schruben, L.W., 44, *53*, 226, 239, *253*, 417, *451*, 462–465, *475*, 554, *574*, 609, *614*, *615*, 618, *631*
Schub, M., 75, *78*
Schurz, H., 402, *412*
Scott, D.W., 211, *223*
Scott, E.M., 227, *256*
Seheult, A.H., 227, 244, *254*
Seila, A.F., 206, 208, *222*, 465, 466, 468, 472, 473, *475*
Serafini, D.B., 541, 564, *571*
Serfling, R.J., 33, *53*, 262, *289*, 430, *452*, 498, *500*
Serfozo, R.F., 197–199, *222*, *223*

Shahabuddin, P., 278, *288*, 292, 298–301, 303–305, 317, 323, 325, 330, 331, 333–337, 340–345, *347–350*

Shao, J., 417, 430, 433, 434, *452*, 567, *574*

Shapiro, A., 609–612, *614–616*, 618, *631*

Sheather, S.J., 213, *222*, *223*

Shedler, G.S., 7, *18*, 34, *53*, 104, 105, *119*, 167, 170, *179*

Sherali, H.D., 546, 556, *571*, *572*

Shetty, C.M., 546, *571*

Shi, L., 609, 610, *614*, 639, 641, 642, 646, 649, 651, *654*

Shi, L.Y., 610, *616*

Shin, M., 543, *574*

Shiue, W.-K., 417, *452*

Shortle, J.F., 208, *222*

Shparlinski, I.E., 75, *80*

Shub, M., 176, *179*

Shwartz, A., 293, *350*

Siegmund, D., 300, 304, 313, *350*, 515, *534*

Sigman, K., 331, *350*, 386, *413*, 491, *500*

Silverman, B.W., 132, 133, *153*, 211–213, 216, *223*

Simard, R., 57, 64, 65, 76, 77, *80*, 183, 184, *191*

Simpson, T.W., 541, 542, 545, 546, 567, *572–574*

Singh, K., 431, 432, 436, *452*

Singpurwalla, N.D., 244, *255*

Sklar, A., 129, *153*

Slepian, D., 510, *534*

Sloan, I.H., 351, 360, 364–366, 372, 375, *379*

Sloane, N.J.A., 62, *78*

Slovic, P., 229, 232, *255*

Smid, M., 77, *81*

Smith, A., 237, *256*

Smith, A.F.M., 113, *118*, 227, 229–232, 235, 240, 241, *254*

Smith, B.J., 239, *256*

Smith, M.J., 530, *534*

Smith, W., 478, *499*

Sobel, M., 524, *534*

Sobol', I.M., 351, 353, 360, 367, 368, *379*

Solanky, T.K.S., 523–525, *534*

Song, W., 247, 250, *256*

Song, W.-M.T., 457, 460, 462, 467, 469, *475*, 517, 518, *534*

Soorapanth, S., 227, 244, *254*

Soto, J., 77, *81*

Spall, J.C., 609, 610, *616*, 619, *631*

Spanier, J., 351, 367, *379*

Spearman, M.L., 610, *614*

Speed, T.P., 127, 129, *151*

Spiegelhalter, D.J., 172, *179*, 226, 231, 237–239, *255*, *256*, 446, *452*

Sreenivas, R.S., 610, *614*, 641, 648, *653*

Srikant, R., 385, 410, *413*

Stadler, P.F., 625, *631*

Stadlober, E., 95, *120*, *121*

Stanfield, P.M., 134–136, *152*, *153*

Starr, N., 205, *223*, 524, *534*

Steckley, S.G., 227, 237, *256*

Steiger, N.M., 459, 460, 467, 470, 472–474, *474*, *475*

Stein, C., 509, *534*

Stein, M., 281, 282, *289*

Steinbugl, J., 167, *180*

Sterman, J.D., 3, *18*

Stidham, S., 198, *223*

Stoffer, D.S., 141, *150*

Stone, M., 545, *574*

Stoyanov, J., 102, *120*

Strickland, C.R., 608, *613*

Strickland, S.G., 610, *613*

Struckmeier, J., 367, *379*

Stuart, A., 139, *153*

Stuck, B.W., 87, *117*

Stump, G., 546, *574*

Sturrock, D.T., 557, *572*

Su, Y., 327, *350*

Subramanian, S., 34, *53*

Sullivan, D.W., 513, *534*

Sullivan, K.A., 639, *654*

Sumant, S., 190, *191*

Sun, G., 612, *613*, *616*

Suri, R., 610, *614–616*

Swain, J.J., 135, *151*

Swann, J., 247, 250, *256*, 517, 518, *534*

Swartz, T., 98, *118*, 237, *255*

Swendsen, R., 113, *120*

Szechtman, R., 262, 271, 274–276, 278, *288*, *289*, 327, *350*

Szekeres, G., 104, *120*

Szpankowski, W., 300, *350*

Taaffe, M.R., 227, *256*, 263, *289*

Tadić, V., 13, *18*

Takács, L., 401, *413*

Tamhane, A.C., 502, 507, 510, 531, *533*, *534*

Tan, K.K.C., 112, *118*, 238, *255*

Tan, K.S., 355, *379*

Tang, Q.-Y., 609, *616*

Tanner, M.A., 237, 238, *256*

Taqqu, M.S., 392, *412*

Tausworthe, R.C., 72, *81*

Taylor, H.M., 401, *412*, 493, *500*

Taylor, R.L., 449, *451*

Tayur, S.R., 608, *614*

Teller, A.H., 109, *120*, 172, *179*

Teller, E., 109, *120*, 172, *179*

Tenga, R.F., 640, *653*
Tew, J.D., 554, 556, *572*, *574*
Tezuka, S., 56, 62, 66, 67, 70–74, 76, *80*, *81*, 351, 364, 369, 373, *376*, *378*, *379*
Thorisson, H., 114, *117*
Thurston, D.L., 546, *574*
Tian, P., 649, *654*
Tibshirani, R.J., 417, 451, *452*, 486, *499*, 567, *574*
Tierney, L., 110, *120*
Tokol, G., 459, 463, 467–473, *474*, *475*
Tokuyama, T., 369, *379*
Tomick, J., 537, *574*
Tompkins, G., 640, *654*
Tong, Y., 337, *350*
Tong, Y.L., 128, *153*
Tootill, J.P.R., 71, *81*
Torczon, V., 541, 546, 564, *570*, *571*
Touzi, N., 609, *613*
Touzin, R., 63, 65, 66, *80*
Traub, J., 351, 354, *378*
Trosset, M.W., 541, 564, *571*
Tsang, W.W., 92, 95, *120*
Tsiolis, A.K., 124, *152*
Tsitsiklis, J., 316, *346*
Tsoucas, P., 328, *346*
Tu, C.-H., 543, *574*
Tu, D., 417, 430, 433, 434, *452*
Tuffin, B., 367, *379*
Tukey, J., 382, *413*
Turnbull, B.W., 515, 517, 526, *532*, *533*
Tversky, A., 229, 232, *255*
Tweedie, R.L., 37, 39, 42, 44, 46, 47, 49, *53*, 112, *120*, 485, *500*

Ulrich, G., 86, *120*

Vakili, P., 610, *616*, 641, 648, *653*
van Beeck, P., 431, *452*
van Beers, W.C.M., 241, 244, *256*, 542, 570, *572*, *574*
van der Corput, J.G., 366, *379*
van Groenendaal, W., 554, *572*
van Oortmarssen, G.J., 548, *573*
Vangel, M., 77, *81*
Vassilev, V.K., 625, *631*
Vattulainen, I., 77, *81*
Vaucher, J., 371, 375, *377*
Vavak, F., 643, *654*
Vázquez-Abad, F., 327, *350*, 609–611, *612*, *614*, *615*, 628, *631*
Vecchi, M.P., 637, *654*
Velleman, P., 382, *413*
Venkatraman, S., 135, *151*, 260, *289*
Vervaat, W., 105, *121*
Vo, S., 77, *81*
von Mises, R., 425, *453*
von Neumann, J., 90, *121*

Wagner, M.A.F., 130, 131, *153*, 165, *180*
Wahba, G., 541, *571*
Wald, A., 505, 506, *534*
Walker, A.J., 89, *121*, 145, *153*
Walrand, J., 321, 329, *349*, 384, *413*
Walsh, L., 365, *379*
Wand, M.P., 27, *53*, 133, *153*
Wang, D., 74, *81*
Wang, H., 326, *347*
Wang, I.J., 610, *612*, *616*
Wang, J., 113, *120*, 227, *256*, 263, *289*, 640, *653*
Wang, X., 351, 360, 361, 375, *377*, *379*
Wang, Y., 299, *348*
Wang, Y.-J., 543, *573*
Wardi, Y., 612, *613*, *616*, 618, *631*
Ware, P., 123, *153*, 167, *180*
Wasserman, L., 231, *255*
Watlington, J.A., 543, *574*
Watson, J., 640, *654*
Wegenkittl, S., 64, 75–77, *78–81*
Weiss, A., 293, *350*, 611, *615*
Welch, B.L., 249, *256*, 508, *534*
Welch, P.D., 208, *223*, 260, *288*
Welch, W., 244, *256*
Welch, W.J., 541, *573*
Weller, G., 367, *376*
Wheeden, R.L., 109, *121*
White, K.P., 156, *180*
Whitley, L.D., 640, *654*
Whitt, W., 22, *53*, 148, *153*, 157, *180*, 262, *288*, 298, *348*, 382–385, 390–393, 395, 398, 401, 403, 404, 406, 408, 410, 411, *411–413*, 460, 462, *475*, 483, *500*
Wilcox, R.R., 517, *534*
Wild, P., 97, *118*, 127, *151*
Wilf, H.S., 173, *180*
Willemain, T.R., 417, *452*
Williams, B.J., 226, 227, 241, 243, 244, *256*, 538, 542, 570, *574*
Williams, D., 263, 266, *289*
Williams, R., 12, *18*
Wilson, D.B., 109, 114, 116, *120*, *121*, 370, *379*
Wilson, J., 190, *191*
Wilson, J.R., 21, 26, 34, *52*, *53*, 130, 131, 134–136, *151–153*, 165, *179*, *180*, 227, 236, 240, 242, *257*, 260, 273, 274, 281, *288*, *289*, 459, 460, 463, 467,

469–474, *474*, *475*, 513, 517, 524, *532*, *534*, 554, *574*
Wilson, K.B., 539, 548, 553, *571*
Winston, W., 239, *257*
Wolff, R.W., 11, *18*
Wolfowitz, J., 610, *615*, 619, *631*
Wolpert, D.H., 625, *631*
Woo, B., 342–344, *350*
Woźniakowski, H., 351, *379*
Wright, R., 189, *191*
Wu, C.F.J., 381, *413*, 440, *453*
Wu, P.-C., 65, *81*
Wu, Y., 252, *254*
Wynn, H.P., 244, *256*, 541, *573*

Xing, C., 353, *378*
Xiong, X., 252, *255*, 610, *616*
Xu, C.-W., 417, *452*
Xu, H., 64, 65, *78*

Yakowitz, S., 628, *631*
Yamaguchi, R., 643, *654*
Yan, D., 620, *631*
Yang, J., 167, *180*
Yao, A.C., 84, *119*
Yao, D.D., 384, *412*
Yesilyurt, S., 538, *574*
Yin, G., 323, *349*, 610, *615*, 619, *631*
Yoshitomi, Y., 643, *654*
Yu, B., 275–277, *288*
Yücesan, E., 227, 250, *254*, 531, *532*

Zaman, A., 67, 68, *80*
Zazanis, M.A., 610, *614*, *616*
Zechner, H., 95, *121*
Zeevi, A., 334, 335, 345, 346, *346*
Zeitouni, O., 293, 308, 309, *347*, 624, *630*
Zhai, W., 620, *630*
Zhang, H., 608, *616*
Zhou, Y., 288, *289*
Zimmer, R.J., 263, *289*
Zionts, S., 546, *574*
Zolotarev, V.M., 87, *121*
Zouaoui, F., 227, 236, 240, 242, *257*
Zygmund, A., 109, *121*

Subject Index

α-mixing 218
α-spending 517

A

absolute width 388
accelerated bias corrected (BCa) 427, 428
accelerated life model 427, 428
acceptance/rejection method 126
adaptive distribution 639
adaptive importance sampling techniques 291, 293, 316, 318, 327
adaptive Monte Carlo 316, 318
adaptive rejection method 99
adaptive rejection Metropolis sampler 112, 238
adaptive rejection sampling 97, 238
adaptive stochastic approximation 316, 322
additive lagged-Fibonacci generator 68
add-with-carry 68
alias method 89, 145
alignment probability 648, 649
almost-exact inversion 95
alternating renewal process 158, 160, 166
alternating series method 93, 94, 101, 103, 104, 106
ANOVA 285, 359, 360, 375, 378
antithetic variates 189
area estimator 464, 466, 468, 470, 473
– batched 463, 464, 467, 470
– overlapping 470, 471
– weighted 465, 470
Arena® 68, 557, 572, 635
arrival processes 153, 155, 157, 181, 191, 408
artificial neural networks (ANNs) 543, 573
asymptotic analysis 458, 502, 522, 523, 647
asymptotic bias 32, 381, 385, 390, 391, 393, 401, 403, 404, 406, 410, 411, 473, 486
asymptotic efficiency 262, 348, 412, 524, 609, 616
asymptotic normality 44, 219, 231, 237, 421
asymptotic probability of correct selection 523
asymptotic sample size 525
asymptotic variance 218, 235, 262, 375, 381, 384, 385, 390–393, 396–401, 403, 404, 406, 407, 409–412, 436, 472, 475, 484, 486, 520
asymptotically consistent 132, 524
asymptotically efficient 300, 349, 350; *see also* asymptotically optimal
asymptotically first-order efficient 525
asymptotically optimal 326, 327, 330, 332–335, 339, 340, 342, 348, 462, 533; *see also* asymptotically efficient
asymptotically random 70, 81
asymptotically robust 523
asymptotically second-order efficient 525
asymptotically unbiased 467, 472, 485, 578
autocorrelated interarrival times 124
autoregressive time-series processes 140
autoregressive moving-average models 140
autoregressive-to-anything (ARTA) process 142, 143

B

b-spline basis 541
bandwidth matrix 132
bandwidth parameter 132, 133
Banerjee's inequality 509, 517
basis functions 539, 540, 542, 543
batch arrivals 160
batch means 43, 44, 46, 210, 449, 457, 460, 461, 475, 512, 513, 519
batched CvM estimator 465–467, 471
Bayes' rule 225, 229, 230, 237, 240, 252
Bayesian approach 225, 226, 239, 245, 252, 255, 447, 448; *see also* Bayesian method
Bayesian LLN 231
Bayesian metamodeling 241
Bayesian method 243, 247, 252, 254; *see also* Bayesian approach
– subjective formulation 226
Bayesian procedures 247, 531; *see also* Bayesian method
BCa – *see* accelerated bias corrected
BC – *see* bias corrected

Behrens–Fisher problem 249, 508
Bernoulli distribution 228, 595
Bernoulli selection 501, 529, 530
Bernstein polynomial 130
Berry–Esséen theorem 430–432, 452
best linear unbiased prediction (BLUP) 243
beta distribution 117
Bézier curve 126, 130
Bézier distribution 130, 153
bias corrected (BC) 427, 428
bias-variance tradeoff 458
binary entropy 84, 89
binomial distribution 95, 596
bivariate gamma processes 138
block sampling 449, 450
– whitening 449
BLUP – *see* best linear unbiased prediction
Bonferroni inequality 513, 514
Boole's inequality 25
bootstrap 132, 415–431, 434–436, 438–453, 486, 499
– confidence interval 423, 424, 441, 452
– double 424, 425
– principle 420
– quantile interval 426
– quantile method 422
bootstrapping – *see* bootstrap
bounded relative error property 299
Box–Behnken design 553
Box–Cox transformation 551
Box–Müller method 85
branch-and-bound method 621
Brownian bridge 107, 117, 363, 364, 378, 429, 463
– process 463
– technique 363, 364
Brownian motion 44, 107, 203, 328, 363, 364, 384, 411, 412, 457, 506, 515, 516, 518, 524
– standard 44, 203, 363, 457, 515, 516
– ϕ-mixing 203, 204
Burr distribution 128

C

canonical/ridge analysis 549
Carleman's condition 102
case sampling 439, 441, 449
Cauchy distribution 85, 128, 589
Cauchy sequence 263
central composite design (CCD) 553, 555, 559
central limit theorem (CLT) 11, 20, 21, 32, 39, 42, 44, 53, 157, 163, 196, 202, 203, 209, 210, 223, 261, 289, 297, 315, 388, 407, 411, 457, 479, 500, 520, 578
characteristic functions 83, 100, 117
characteristic polynomial 72–75
Chebyshev's inequality 195
chessboard copula 141, 145
chessboard distribution 145, 151
CLT – *see* central limit theorem
coefficient of variation 559–562
collision-free value 75
combinatorial objects 109, 155, 172–174
combinatorial optimization 627, 633–635, 640, 642, 646, 649, 652
combined generator 68, 74–76, 79
combined MRGs and LCGs 66
common random numbers (CRN)
comparisons with a standard 526, 534
COMPASS method 620
competing risks model 168, 169
competing risks technique 169
composition finding problem 149
compound Poisson process 159, 160
compound symmetry 514
conditional importance sampling 341, 342, 350
conditional Monte Carlo (CMC) 25, 30, 34, 41, 53, 259, 260, 266, 273, 274, 286, 364, 607
conditional rank correlation 146, 147
confidence intervals (CI) 13, 23, 27, 29, 34, 42–44, 162, 163, 187–189, 193, 195, 219, 222, 223, 239, 279, 297, 385, 391, 398, 416, 422, 423, 428, 434, 452, 474, 475, 477, 507, 508, 513, 534, 539
conjugate distribution, noninformative 231
conjugate-gradient direction 556
consistent estimator 22, 29, 195, 196, 201, 204, 386, 387, 443
continuous-decision-variable optimization 17
control variate (CV) 16, 17, 189, 259–262, 266–268, 270, 274, 280, 285, 286, 288, 289, 330, 375, 377, 451, 554
convergence analysis 631, 633, 651, 654
convergence in distribution 23, 195, 261, 388, 407, 460
convergence in mean 195; *see also* quadratic mean
convergence in total variation 109
converging together lemma 20, 24, 27
copula 88, 120, 129, 130, 141, 145–147, 151, 152, 338, 345, 349
Cornish–Fisher expansion 434
correlation induction 119, 184, 571, 573, 574
correlation matrix 138, 139, 142, 145–148, 151, 175, 176, 510
counting function 156
counting process 156, 159, 165

coupling 85, 109, 114–116, 121
coupling from the past (CFTP) 109, 114–116
covariance function 222, 243, 411, 457, 467
covariance matrix 33, 34, 135, 136, 138, 140, 143, 151, 267, 315, 364, 551
covariates 170, 171, 443, 559, 560
coverage probability 196
Cox model 171
Cox proportional hazards model 170
Cramér–von Mises (CvM) 444, 445, 451, 452, 459, 465, 475
Cranley–Patterson method 371, 372
credit risk 293, 337, 344–346, 348, 349
CRN – *see* common random numbers
cross-entropy 316, 319, 334, 346, 347, 350
cross-validation 416, 442, 539, 541, 545, 564, 567, 574
Crystal Ball® 635
cumulative hazard function 167–172
cumulative intensity function 161–163, 178
cycle boundaries 491
cycle length 183; *see also* period length

D

d-variate copula 146
De la Vallée–Poussin density 100
delta method 32, 34, 262, 279, 426, 498
density estimation 20, 25, 27, 118, 133, 153, 194, 210, 211, 216, 221, 222, 480
dependence measures 137, 150
derivatives of measures 591
descent algorithm 620
design matrix 439, 550
design of experiments 539, 583
design region 554–558, 564, 565
determinant of an M-matrix 175, 177
deterministic recurrence 56
Dickman's distribution 105, 106
DIEHARD 77, 80
difference gradient estimator
– forward 579, 581
– one-sided 579, 581
– random directions 582, 610
– symmetric 581
diffusion process 381, 385, 394, 397, 401–405, 412
digital net 381, 388, 394, 397, 401–405, 412
digital sequence 366–368, 377
digital shift 372
dimension reduction 363, 364
dimension-stationary 361, 366
Dirac δ-function 594
direct gradient estimation 580, 583, 584, 607, 608
direction numbers 368
discontinuous perturbation analysis 610, 616
discrepancy measures 58
discrete Metropolis chain 110
discrete optimization 13, 576, 618, 630, 631, 649
distributional identities 83, 105, 106
divide-and-conquer algorithm 67
dominated convergence 32, 50, 605, 606
dominating point 311, 312, 349
double exponential distribution 97
doubly stochastic Poisson process 159
dual lattice 62, 71, 72, 372, 373

E

EDFIT statistics 445
EDF – *see* empirical distribution function
Edgeworth expansion 417, 433, 434, 452
efficiency improvement techniques 16, 260; *see also* variance reduction techniques
eigenvalues 140, 174–176, 179, 232, 324, 325, 364, 556, 563
elliptical cupolas 147
elliptically symmetric distribution 134
empirical distribution function (EDF) 415, 418
empirical statistical testing of RNG 56
EM – *see* expectation–maximization
equidistribution 70–72, 78, 79, 289
equilibrium residual-life 401; *see also* stationary-excess
ergodic 36–38, 40, 42, 44, 45, 114, 200, 202, 203, 349, 485, 649; *see also* ergodicity
ergodicity 36, 37, 39, 119, 200, 202, 588; *see also* ergodic
error estimation 11, 14, 378; *see also* estimation error
estimated approximate probability of correct selection 249
estimation error 11, 13, 14, 16, 17; *see also* error estimation
event-epoch process 6–8
event-scheduling world view 7
evolutionary algorithm 638, 653
evolutionary computing 625
exact Markov chain method 109
exchangeable 229, 231, 232, 241, 245
expectation–maximization (EM) 237
expected information 235
expected loss 234, 235, 248, 249
expected opportunity cost 249, 252, 254

expected value of information 225, 227, 233, 244, 248, 252
exponential distribution 97, 389, 436, 557, 598
exponential family 230, 235, 242, 331, 595
exponential power distribution 97
exponentially twisted distribution 293, 308
exponential-polynomial-trigonometric function with multiple periodicities (EPTMP) 654, 665
extended circle 450
extremal distribution 148, 149

F

Fabian's bound 516
face centered design 561, 562
factorial experiment designs 542
failure rate 169, 299; *see also* hazard function
failure times 156, 158, 159
Faure sequence 353, 367–369
FCLT – *see* functional central limit theorem
Fejer kernel 102
Finetti-type representation 229, 233, 253
finite automata 84
finite difference perturbation 580
finite differences 537, 575, 577, 579, 580, 582, 583, 610, 616
finite perturbations 610
finite-difference approximation 537
finite-horizon performance measures 9, 10
finite-horizon simulation 193, 221
first Lyapunov condition (FLC) 36
first-order unbiased 464, 466, 467, 471, 472
– estimator 464, 467, 472
force of mortality 168; *see also* hazard function
Foster–Lyapunov condition 36; *see also* first Lyapunov condition
Fourier coefficients 83, 101, 118, 372
Fourier decomposition 445
fractional factorial design 548, 553, 555
frequency curve 134, 152
frequency domain experimentation 609
fully sequential procedure 255, 506, 512, 516, 527, 533
functional ANOVA decomposition 359
functional central limit theorem (FCLT) 44, 203, 457, 500, 520
future event schedule 489

G

gain sequence 578
gamma distribution 95, 220, 239, 241, 597
Gaussian distribution 97, 309, 582
Gaussian quadrate 352
Gaussian random function (GRF) 241, 253
generalized cross-validation (GCV) 541
generalized feedback shift register (GFSR) 73
generalized hill-climbing (GHC) 639, 654
generalized mean-value theorem 31, 32, 605
generalized scale parameter 589, 600
generalized semi-Markov process (GSMP) 18, 53, 489, 500, 611
generating a random combination of k integers 173
generating a random permutation 174
generating a random shuffle 174
generating a random subset 173
generating combinatorial objects 109, 155, 173
generating correlation matrices 175, 179
generating Markov chains 172
generating random colors 172
generating random geometric objects 155, 172
generating random sequences 172
generating random spawning trees 155
genetic algorithm 617, 620, 633–640, 642–646, 648, 651, 653, 654
– anti-ranks 644
– q-tournament selection 644
– roulette strategy 643, 644
geometric distribution 333
Gibbs sampling 118, 128, 151, 238
global metamodel 535, 537, 538, 545, 546, 548, 563–567, 569
goal softening 649, 652
goodness-of-fit 58, 165, 210, 239, 444–446
– Anderson–Darling test 211
– chi-square test 210
– Kolmogorov–Smirnov test 210
gradient-based methods 237, 536
gradient estimation 25, 53, 417, 500, 575–577, 580, 583, 584, 607, 608, 610, 611, 614, 615
gradient-free methods 237
gradient search 17, 327, 634
Gram–Schmidt process 267, 284
greedy randomized search procedure (GRASP) 638, 654
Greeks 608
grid search 564, 568
group-screening procedure 518
GSMP – *see* generalized semi-Markov process
Gumbel distribution 85, 97, 589

H

Hahn–Jordan decomposition 592, 593, 596
Halton sequence 353, 361, 367, 379
Hamiltonian 174
hash tables 89

hazard function 167–172, 334, 339, 341, 348
hazard rate twisting 333, 334, 344, 349
hazard rates 104
hazard-based method 167
heavy-tailed random variable 291–293, 331, 332, 334, 337
heavy-traffic stochastic-process limit 381, 406, 408
hedging 2, 505, 576, 608, 613
Hermitian operator 174
Hessian matrix estimation 610
heteroscedastic 438, 439, 533
highly-uniform point set 55, 58, 351–355, 357, 361, 362, 364, 369, 371, 375
Hilbert space 259, 260, 263–266, 269, 271, 273, 274, 277–281, 287–289
homogeneous Poisson process 27, 119, 157, 158, 161, 166, 168, 170, 178, 179
Huffman tree 89
hyperbolic secant distribution 97
hypergeometric distribution 95, 120
hyper-Poisson distribution 98

I

importance sampling 236, 237, 288, 289, 291–301, 304–306, 308, 313, 314, 316–319, 321, 323, 324, 327–329, 332–336, 338–342, 344–350, 452, 580, 593, 612
incomplete block design 553
independence sampler 110
indifference judgment 230
indifference zone (IZ) 245–247, 252, 255, 503, 504, 506, 520, 522, 528, 529, 533, 534, 641, 647, 650–652
indifference-zone formulation 503, 504
indifference-zone parameter 520, 522, 529
indirect gradient estimation 580, 583
indirect-gradient optimization strategy 537
indirectly specified distributions 83
infinite horizon discounted reward 477, 478, 498, 499
infinite-horizon performance measures 10
infinitely divisible distributions 106, 117, 120
infinitesimal generator matrix 394
infinitesimal perturbation analysis (IPA) 586, 603, 613, 614
influence function 425–427, 429
information matrix 231, 421
initial bias 21, 477, 479, 483–486
initial sample size problem 511
initial transient 53, 117, 208, 210, 222, 385, 398, 479, 480, 483, 484, 486
initialization bias 21, 485, 486
input uncertainty 34, 225, 227, 233, 236, 240, 242, 252, 253, 254
integration lattice 79, 364, 365, 377
intensity function 158, 159, 161–165, 168, 171, 178, 179
in-tree network 330
inverse cumulative hazard function technique 167–170
inverse function metamodel 570
inverse marginal transformations 141
inversion 75, 83, 84, 88–90, 92, 95, 98, 100, 103, 113, 114, 131, 146, 162, 163, 166, 167, 170, 186, 255, 342, 356, 362, 423, 424
inversion method 83, 84, 88, 90, 113, 170, 356

J

Jackson network 317, 322, 328–330, 348, 349, 607
Jeffreys' prior distribution 230, 255
Johnson translation system 134, 135
Johnson variates 136
joint input–output model 240, 244
jump-ahead method 57, 67, 70

K

Kendall's τ 139, 140
kernel 27, 53, 102, 126, 132, 133, 150, 153, 194, 195, 212–218, 220, 221, 223, 236–238, 253, 256, 402, 422
– bandwidth 133, 214–216
– estimator 27, 126, 132, 133, 194, 195, 212–214, 216, 217, 220, 221, 223, 236–238, 253, 256, 422
– functions 212, 216
– roughness 213
– smoothing parameter 212
– variable estimator 214
kernel density estimation 132, 133, 150
Kiefer–Wolfowitz algorithm 579–581, 610
Kimball's inequality 511
Kimball's theorem 510
knots 541
Koksma–Hlawka inequality 354, 359
Kolmogorov measure 83
Kolmogorov's canonical representation 106
Kolmogorov–Smirnov
– distribution 104, 118
– limit distribution 94, 118
Korobov formulation 357
Korobov lattice rules 59, 353, 364, 365
Korobov point set 357, 358, 366, 369

Korobov rules 353, 366
Krein's condition 102
kriging 243, 256, 541, 572, 574; *see also* spatial correlation
Kullback–Leibler distance 319, 320
Kullback–Leibler divergence 240, 245, 252
Kummer's distribution 97

L

lagged–Fibonacci generator 63, 64, 68
landscape structure 625
Laplace density 91
large deviations theory 309, 329, 624
large-sample properties 19
large-sample theory 222, 388, 392, 533
Latin hypercube sampling (LHS) 259, 260, 268, 281, 283, 285, 286, 288, 289
Latin hypercube 286, 542, 543, 548
lattice
– dual 62, 64, 71, 72, 372, 373
– integration 79, 364, 365, 377
– property 62
– random variable 308, 431
– rule 59, 61, 64, 80, 353, 364–366, 371, 373, 377–379
– structure 56, 61–63, 66, 71, 72, 78, 79
law of large numbers (LLN) 20, 22, 39, 196, 202, 294, 296, 310, 387, 431, 479, 500, 546, 588, 622
LCG – *see* linear congruential generator
least-favorable configuration (LFC) 505
leave-one-out method 564, 567
Lebesgue density theorem 109
Legendre polynomial 103
Letac's lower bound 111
Lévy measure 106
Lévy processes 107, 120
lifetime generation 171, 172
likelihood function 160, 164, 225, 230, 244, 256, 542
– ratio (LR) 53, 236, 292, 294, 295, 297, 299–301, 306, 307, 309, 311, 313–315, 319, 321, 326, 329, 333, 338, 339, 343, 349, 496, 497, 500, 537, 575, 584, 586, 611, 614, 615
limit distribution 94, 106, 109, 118
limit law method 109
Lindley's equation 356, 586
linear congruential generator (LCG) 79, 80, 177, 357
linear feedback shift register (LFSR) 59, 72
linear recurrence 55, 56, 60, 67, 69, 74, 77, 80, 81
linear recurrences modulo *m* 60, 61, 64, 65, 67, 75
linear recurrences with carry 67
linear-type generator 55
Linnik distribution 87, 119
Linnik–Laha distribution 101
Lin's condition 102
Lipschitz condition 374
Lipschitz densities 99
Little's law 198, 401
LLN – *see* law of large numbers
local metamodel 546–549, 552, 557, 558, 563, 569
location parameter 134, 135, 524, 532, 589
logistic distribution 97, 589
longest route problem 14–16
look-ahead density estimators 25, 27, 29
loss function 228, 233–235, 247, 248, 252, 338, 342
low-discrepancy point set 353
L^p limit properties 202
ℓ_p-norm 233
Lyapunov condition 20, 36, 37, 39, 47, 49
Lyapunov function 20, 21, 46, 48, 50

M

majorization technique 170
majorizing function 120, 167
majorizing hazard function 170
majorizing value 167
Malliavin calculus 609, 612, 613
MAP – *see* maximum a posteriori probability
Markov additive process 313, 317, 323, 325
Markov chain central limit theorem (MCCLT) 42
Markov chain strong law of large numbers (MCSLLN) 20, 22, 39
Markov chain
– continuous time (CTMC) 305, 394, 493
– discrete time (DTMC) 36, 305, 394, 490–492, 494, 530
– general state space 20, 36
– mixing time 115
– Monte Carlo (MCMC) 237, 255, 256, 446
– stationary distribution 41, 114, 115, 238, 244, 399, 404, 494, 649
Markov decision process 316, 347, 610, 613
Markov process 20, 36, 204, 209, 216, 293, 302, 303, 316, 381, 384, 485, 487–491, 493, 494, 497, 614
Markov-additive process 313
martingale 53, 287–289, 478, 493, 494, 496, 497
– semi- 328

matrix linear recurrence 69
maximally equidistributed 71, 74, 75, 79
maximally equidistributed combined LFSR generator 79
maximally equidistributed combined TGFSR generator 74
maximum a posteriori probability (MAP) 232
maximum entropy methods
– diffuse 231
maximum likelihood (ML) 160, 163, 210, 226, 420, 438, 539, 541, 551, 558
– estimator 160, 161, 163, 420, 539, 542, 558
– function 160, 163
Maxwell distribution 597
MCMC – *see* Markov chain Monte Carlo
mean squared error (MSE) 16, 132, 133, 195, 218, 262, 263, 386, 387, 390, 462, 473, 475
measures of equidistribution 70
measure-valued differentiation 584, 624
median deviation concordance 139
Mersenne twister 70, 73, 74, 77, 80, 81
metaheuristics 536, 537, 633–643, 646, 648, 649, 651–654
metamodel 17, 241, 437, 535–574
– Bayesian 241–245
– kriging 243, 256, 541, 572, 574
– linear 438, 440
– nonlinear 440
metamodel-based optimization 537, 538, 545–549, 563, 564, 569, 570
method-of-moments approach 240
Metropolis random walk 112
Metropolis–Hastings chain 109, 111, 112
minimum information copula 146
Mittag–Leffler distribution 87
mixed Poisson process 159
mixing hypotheses 21
mixture method 89, 148
MLE – *see* maximum likelihood
modeling error 11, 13, 14
moment condition 34, 50, 434, 435
moment function 199
– autocovariance 199, 200, 405, 409
– mean 199
moment generating function 38, 83, 103, 309, 331, 340, 343
monotonic-transformation-invariant measures of dependence 139
Monte Carlo 15
– adaptive 316, 318, 347
– conditional 25, 29, 34, 41, 53, 259, 260, 266, 273, 274, 286, 364, 607
– Markov chain 179, 237, 255, 256, 446, 452
– quasi 58, 59, 78, 80, 81, 118, 288, 351, 352, 354, 376–379, 582
– randomized quasi 80, 354, 376, 377
– weighted 260, 269, 275, 285, 286, 288
Morris design 553
multinomial selection 501, 527–529, 533, 534
multiple comparison procedures (MCP) 502, 519, 523, 532, 533
multiple comparisons with the best (MCB) 507
multiple recursive matrix generator 60, 61, 80
multiple recursive matrix method 70, 80
multiplicative Poisson equation 317, 323, 324
multiply-with-carry (MWC) 67, 78, 79
multiquadric basis functions 543
multistage procedures 514
multistart gradient-based optimizer 568
multivariate input processes 123, 126, 181
multivariate time series 124, 125, 138, 140, 142, 150

N

naïve resampling 133
naïve simulation 294–298, 330
negative binomial distribution 596
neighborhood structure 620, 626–628, 638
nested partitions method 633, 635–642, 646, 647, 649–652, 654
– backtracking 642
– global guidance system 642, 649
NETLIB 656, 573
network completion time 21
neural network metamodel 543
Newton–Raphson scheme 578
No-free-lunch theorem 625
nonhomogeneous Poisson process (NHPP) 119, 158, 159, 161, 166, 168, 170, 178, 179
nonlinear generator 55
nonoverlapping batch means 457
nonoverlapping batches 459, 463, 468, 469, 474
nonparametric approximation method 149
nonparametric estimations 126, 178, 179, 191, 222, 223
nonparametric functional estimation 27, 53
nonparametric method 211
nonstationary Poisson process 180, 186
nonstationary processes 6
nonuniform random variate generation 83, 94, 95, 119
normal copula model 338
normal distribution 106, 117, 125, 133, 134, 139, 152, 153, 162, 163, 196, 220, 231–234, 240, 241, 249, 308, 337, 338, 345,

350, 363, 377, 405, 408, 427, 431, 437, 446, 504, 505, 510, 524, 582, 589
– standard 162, 163, 196, 427, 431, 524
normal-to-anything (NORTA) distribution 142, 143, 151, 179

O

optimal computing budget allocation 236, 245, 249, 255
optimal stopping problem 609
optimization via simulation 17, 502, 534, 572, 613, 631, 654
OptQuest® 545, 634, 635, 644
Ornstein–Uhlenbeck 404
orthant probability 139, 140
orthogonal array 542, 548
orthogonal array-based Latin hypercube 542, 548
– output function 541
overdispersed processes 158
overlapping area estimator 470, 471
overlapping batch means 457, 469
overlapping batches 459, 463, 467–471, 475
overlapping CvM estimator 468, 471–473

P

paired *t* method 187, 188
parallel and distributed simulation 3, 18
parametric bootstrap 420, 421, 446
Pareto 85, 97, 222, 332–335
Pareto distribution 595
Pareto-type tails 332, 334
Pearson correlation 138
Pearson-type distribution 134
– type IV 97
percentile 213, 244, 256, 422, 428, 455; *see also* quantiles
perfect sampling 106, 115, 118, 121; *see also* coupling from the past
period length 56, 57, 60, 65, 66, 68–70, 73–75, 77
PeRK 542, 573
Perks' distribution 97
Perron–Frobenius eigenvector 324, 325
Person-in-the-loop simulations 3
perturbation analysis (PA) 53, 575, 584, 605, 610–614, 616
perturbation matrix 175
piecewise polynomial basis function 540
plug-in approach 419
Poisson distribution 98, 108, 116, 117, 230, 400, 597
Poisson equation 317, 323, 324, 412
Poisson point process 105, 108
Poisson process 27, 105, 119, 156, 164–166, 168, 170, 179, 180, 186, 190, 220, 363
polar method 86
Pollaczeck–Khinchine formula 331
Pólya–Eggenberger distribution 98
polynomial LCG 70
polynomial trend 165
positive density assumption 490
posterior distribution 120, 226, 230, 231, 233, 235, 237, 238, 246, 248, 253, 254, 256, 447, 448
posterior normality 231
– smoothness 232
– steepness 232
posterior probability 229, 247
power law process 164
powers-of-two decomposition 65
precision 64, 144, 162, 177, 182, 186–189, 194, 232, 234–246, 248, 381–383, 385, 386, 388, 390, 391, 398, 467, 473, 554, 577, 625
predictive distribution 234, 248
primitive polynomial 61, 70, 368
prior distribution 228–235, 240, 245, 247, 248, 253, 255, 447, 448
probability integral transform 139, 169
probability of an incorrect selection (ICS) 506
probability of correct selection (PCS) 226, 246, 247, 249, 501, 518, 523, 528
process-interaction view 156, 165, 182
process-interaction 5–7
product-moment (and rank) correlation matching problem 144
product-moment correlation 130, 138–141, 143, 146
product-moment covariance 137
projection theorem 265
Promodel® 634, 635
proportional hazards model 170–173, 179
pseudorandom number generator 352, 369
pseudorandom numbers 77, 78, 81, 152, 351, 352, 375
pure bit model 84
purely periodic sequence 56, 58, 66, 70
push in method 611
push out method 608, 611, 615

Q

quadratic mean 195, 216; *see also* convergence in mean
quantile estimation 207, 208, 277, 288, 422, 432
quantile estimator 50, 432
quantile method 422, 426–478

quantile 9, 15, 50, 53, 193, 196, 206–208, 253, 277, 419, 422–428, 430, 432, 434–436, 459, 504, 505, 509, 583; *see also* percentiles
quasi Monte Carlo (QMC) 58, 59, 78, 70–81, 118, 288, 351, 352, 354, 376–378, 380, 382
quasi-Monte Carlo integration 58, 377, 379
quasi-Newton direction 556
quasi-random numbers 351–354, 357, 375

R

radial basis function 541, 543, 548, 572, 574
radical-inverse function 366, 367
Radon–Nikodym derivative 297
Radon–Nikodym theorem 24
random correlation matrices 175, 176; *see also* generating correlation matrices
random directions method 610
random lifetimes 155, 167, 168, 170, 181
random matrices 174–177, 179
random number generator 55, 56, 76, 78–81, 177, 182–185, 189, 190, 252, 294
random objects 110, 155, 172, 190
random partition of an integer 173, 174
random polynomials 177–179
random search 68, 537, 617–624, 626–630, 634
random variate generation 17, 56, 57, 83–121, 131, 132, 179
random walk 52, 76, 112, 115–117, 291–293, 295, 303, 304, 309, 314, 315, 317, 322, 331, 334, 338, 347, 349, 350, 378, 530
random walks on graphs 115–117
randomized quasi-Monte Carlo 80, 354, 376–378
random-number generation 17, 182, 186
random-number generator 182–185, 189, 190; *see also* random-number generation
random-structure generation 184, 186
– body art 185, 188, 189
– faucets 185, 188–190
rank correlation 130, 139, 141, 143–147, 152
ranking and selection (R&S) 226, 236, 245, 252–255, 474, 501, 502, 532–534, 536, 537, 576, 627, 630, 644, 651–653
rare perturbation analysis 610, 611
rare-event probabilities 292, 302, 309, 317, 319, 327, 330
rare-event simulation 291, 293–296, 299, 302, 330, 335, 346–350
rate function 105, 168, 186, 310, 333; *see also* hazard function
raw event times 161, 162
RBM – *see* reflected Brownian motion
recurrence-based point set 59, 358, 361, 369
recurrence-based QMC 70
recursive estimators 216, 219
reflected Brownian motion (RBM) 328, 384
regeneration 53, 204, 316, 457, 474, 477, 478, 482–485, 489–493, 496–499
– randomized 490
regeneration state 483, 484
regenerative cycles 363, 478, 484, 486
regenerative estimator 480, 494, 499, 602
regenerative method 34, 43, 210, 477, 478, 487, 488, 492, 494, 495, 499, 500
regenerative process theory 622
regenerative structure 316, 477, 478, 480, 481, 486–488, 490, 491, 497, 498, 601
rejection methods 83, 86, 90, 93–96, 99, 103, 105, 106, 111, 112, 116, 119, 126, 127
relative error 206, 297, 299, 300, 306, 332, 341, 346, 383, 400, 434
relative width 223, 297, 388, 389, 392, 398, 404, 405
reliability and survival analysis 155, 167
reliability theory 157, 159
renewal process 157, 158, 160, 165, 166, 171
renewal theory 485
replication 9, 15, 190, 204–210, 229, 278, 375, 387–390, 497, 512
replication method 9, 15
required sample size 388
residual sampling 132, 133, 415–417, 422, 437, 439, 440, 442, 444, 446, 450–453
– innovations 448
residual workload 35
resolution 71, 72, 74, 181, 191, 373, 555, 643
– gap 71, 74
– V 555
response surface methodology (RSM) 537, 539, 547, 548, 552, 572, 573
response surface modeling 225–227, 252, 539, 545, 546, 548, 553, 554, 556, 573
Rinott's procedure 511, 514, 520, 534
Robbins–Monro algorithm 323, 578–580
robust design 538, 570, 573
RQMC – *see* randomized quasi-Monte Carlo
run length 194, 381, 383, 384, 390–393, 397–400, 406, 408–411, 413, 537, 552, 594

S

safety stock selection problem 13
sample averaging approximation 578
sample path optimization 578, 615, 631
sampling strategy 618–620, 626

– point-based 620, 621, 626
– population-based 620, 626
– set-based 620, 621, 626
sampling with replacement 174, 444, 446, 450
scale density 402
scale parameter 135, 138, 164, 220, 589, 595, 597, 598, 600
scatter search 635, 638, 639, 653
Scheffé's identity 112
score function (SF) 575, 586, 611, 615
scrambling 373, 374, 376
screening 246, 247, 253, 502, 516–520, 534, 576, 642, 644
second Lyapunov condition (SLC) 47–49
selecting the largest probability of success 529
selecting the system most likely to be the best 501, 527
selection of the best 501, 526, 532–534, 649
sensitivity analysis 231, 236, 546, 552, 575, 576, 581, 582, 611–615, 631
sensitivity indices 360, 379
separable metric space 487
sequential selection procedures 505, 533
serial tests of uniformity 76, 80
shuffling 76, 172, 179
Silverman's rule 219
simplex search 537
Simpson's rule 236, 352
SimRunner® 634, 635
SIMUL8® 635
simulated annealing 545, 617, 620, 621, 630, 631, 635–640, 649, 653, 654
simulation-based methods 237
simultaneous perturbation stochastic approximation algorithm (SPSA) 582, 612, 613, 620
sinusoidal periodicities 165
Skorohod space 203
Slepian's inequality 249, 250, 511
slice sampler 113, 116, 120
slippage configuration (SC) 516
SLLN – *see* strong law of large numbers
Slutsky's theorem 24, 163, 219
small composite design 553, 555
small-sample properties 19
smoothed bootstrap 423
smoothed perturbation analysis (SPA) 587, 613, 614
smoothing parameter 212, 541, 567
Sobol' sequence 353, 360, 361, 364, 367, 368, 376, 378, 379
solution-to-solution search method 636–638, 644; *see also* point-based search
spatial correlation 541, 542, 545, 548, 573
spawning trees 155, 172
Spearman rank correlation 139
spectral density 450, 480
spectral resampling 450
spectral test 62, 64, 79, 372
speed density 402
sphericity 514
splines 540, 541, 543, 548, 564, 565, 571–573
– cubic 543
– interpolating 541
– metamodel 540
– smoothing 541, 564, 565, 572
– thin plate 543
S-plus 175, 176, 178, 239
squared coefficient of variation 389
standardized sample mean 465
standardized time series (STS) 44, 53, 222, 457, 459, 462, 465, 469, 470, 475, 533
star discrepancy 353, 354
state-change process 8
state-dependent drift 49
static importance sampling technique 293, 317, 323, 327
stationarity 117, 156, 157, 199, 200, 243, 460, 479, 482; *see also* stationary
– weak 200
stationary arrival process 5
stationary distribution 41, 114, 115, 238, 244, 399, 404, 494, 649
stationary probability vector 115, 394, 395
stationary process 6, 21, 199–204, 216, 217, 222, 223, 391, 392, 398, 404, 464, 480
stationary-excess – *see* equilibrium residual-life
stationary – *see also* stationarity
– arrival process 5, 159, 186
– distribution 36, 39, 45, 111, 114, 238, 394, 402, 494, 649
– process 11, 140–142, 147, 156, 199–204, 216–220, 390–393, 449, 464, 480, 482
Statlib 542
steady-state estimation problem 10, 11, 479, 494
steady-state gradient 478
steady-state parameter mean 193
steady-state performance measures 479, 496, 588
steady-state simulation 10, 19, 21, 34, 35, 193, 194, 200, 221, 287, 454–456, 459, 475, 477–481, 484, 487, 489, 490, 500, 512, 513, 517, 519, 520, 533, 619, 622, 623, 630
Stirling's approximation 96
stitching transformation 148

stochastic activity network (SAN) 4, 14, 20–22, 286, 287, 364, 575, 576, 581, 598, 609, 613
stochastic approximation (SA) 50, 316, 317, 322, 323, 327, 346, 349, 537, 578, 580, 582, 583, 610, 613, 615, 616, 619, 630, 631
stochastic comparison 620, 630
stochastic counterpart 578, 610, 613
stochastic gradient estimators 575, 577, 580, 588, 607, 611
stochastic optimization 573, 575, 609, 615, 618, 630, 631, 640, 647, 654
stochastic root finding algorithm 140, 630
stochastic ruler method 620, 621, 630
stratification 259, 260, 270, 278–282, 286, 288, 340, 348
– post- 279, 281
– proportional (PS) 278–280
– refined proportional (rST) 282
streams 57, 76, 77, 80, 176, 178, 183–186, 188–191, 539, 554
strip method 92
strong law of large numbers (SLLN) 20, 22, 39, 196, 202, 431, 588, 622
strong law of large numbers estimator 22, 196
structural IPA 610
structure function 168
Student *t* 86, 90, 97, 98, 247
Studentization 424, 435; *see also* Studentized
Studentized 423–425, 427, 428, 431, 432, 434
Studentized bootstrap method 424, 428, 434
subexponential family 331
subjective probability 225–257
subset selection formulation 254, 503–505, 508, 515, 517, 518
substreams 57, 80, 183–185, 188, 190, 191
subtract-with-borrow 68, 81
surrogate models 538
surrogate-processes evaluation 458
survival models involving covariates 170, 172
survivor function 168
synchronization 184–186, 188–190, 362, 581
systems dynamics 3

T

tabu list 637, 638, 641, 642, 645
tabu search 545, 617, 620, 628, 633, 635–642, 644–646, 649, 653, 654
target arrival process 160
Tausworthe generator 72, 79
T-convexity 98
t -copula model 345, 349
tempering 73, 81
terminating simulation 9, 15, 20, 43, 45, 259, 260, 287, 456, 512, 517; *see also* transient simulation
terminating-simulation performance measures 9
TES – *see* transform-expand-sample process
tests of linear complexity 77
theta distribution 229, 594
thinning 104, 105, 108, 119, 167, 168, 170, 179, 180
third Lyapunov condition (TLC) 49
threshold function 544
time-average variance constant 478, 479; *see also* asymptotic variance
time-shift invariant 200
transformation-based simulation metamodels 34
transformed-density-rejection method 127
transform-expand-sample process (TES) 141, 147, 148, 152
transient simulation 622; *see also* terminating simulation
transition function 56, 57, 75, 77, 369, 394
tree-based generator 84
trinomial-based LFSR generator 72, 73
truncation 90, 208–210, 334, 360, 361, 363, 532, 601
Tukey's *g* and *h* transformations 136
twisted GFSR (TGFSR) 70, 73, 80
two-stage nested partition (TSNP) 650, 654

U

uncertainty analysis 152, 226, 228, 233, 236, 239, 244, 252, 253, 256
– experimental design 226, 233, 234, 243, 252
– parameter uncertainty 226, 235, 236, 241–243
– response surface uncertainty 226, 227
– stochastic uncertainty 226, 240, 242, 252
underdispersed process 158
uniform convergence 27
uniform random number generation 55–81; *see also* random number generation
uniformity measures 58
uniformly bounded time 94, 95
universal generator 93, 97, 113, 119, 120

V

validation 125, 126, 245, 444, 445, 452, 545, 554, 556, 563, 564, 567–569, 574
value of information procedure, 245, 248
value-at-risk 2, 293, 336–340, 342, 348, 350

variable neighborhood search (VNS) 636, 638, 653
variance estimator 457–459, 469, 472, 474, 475, 500, 509, 511, 512, 520, 524, 525
variance reduction 27, 50, 52, 53, 80, 85, 185–189, 240, 259–289, 291, 292, 295, 298, 301, 313, 315, 330, 348, 355, 372, 374, 377, 382, 413
variance stabilizing transformation 551, 554, 566
variance-correction algorithm 133
variance-reduction technique 185, 186, 189, 382
vector-autoregressive-to-anything (VARTA) process 143
vine copula method 142, 146, 151, 152
virtual generator 57
virtual reality 3
V-uniform ergodicity 45

W

Wald's lemma 506
warm-up interval 194
Wasserstein metric 85
weak convergence 203, 457, 479, 497, 578
weak derivatives (WD) 575, 584, 587, 591–593, 598, 602, 603, 606, 608, 611, 614
Weibull distribution 85, 333, 595, 597, 598
weighted Cramér–von Mises estimator 465
weighted Monte Carlo 259, 260, 269, 275, 286–288
world view 6, 7
worst-case standardized spectral test 69

Z

zero-variance estimator 293, 295, 302, 311, 320
zero-variance measure 293–295, 298, 307, 308, 316–319, 320, 322, 325, 327, 342–344