Efficient simulation of network performance by importance sampling

Poul E. Heegaard

Submitted to the
Norwegian University of Science and Technology
in partial fulfilment of the requirements
for the degree of Doktor Ingeniør,
May 1998.

To Kristine,
Helene and Solveig

Abstract

Simulation is a flexible means for assessment of the quality of service offered by a telecommunication system. However, when very strict requirements are put on the quality of service, the simulation becomes inefficient because the performance depends on *rare events* to occur. A rare event is, for instance, a cell loss or a system breakdown. A simulation technique that speeds up the experiments must be added. Various techniques are known from the literature and they should be combined to achieve additional speedups. The most efficient speedup techniques for systems dependent on rare events, are *importance sampling* and *RESTART.*

The importance sampling technique is very sensitive to the change of the underlying simulation process. This is denoted the *biasing* of the simulation parameters. In this thesis, explicit expressions of the variance of importance sampling estimates and the likelihood ratio are developed for an M/M/1/N queue. The study of how the variance expressions vary as the biasing changes, demonstrates that the importance sampling is very efficient in a narrow region, and that the variance is unbounded outside. It is also observed that, seemingly, the likelihood ratio and its variance may be used as an indication of the accuracy of simulation results, in combination with the variance of the estimate itself.

Neither importance sampling nor RESTART are easily applied to multidimensional models, e.g. a model of a telecommunication network with a variety of different users. In this thesis, the focus is on how to do *importance sampling* simulations of telecommunication networks with balanced utilisation of the resources. A network system are described by a *multidimensional* model. The balanced resource utilisation implies that the system performance is not given by a single bottleneck. Hence, previous approaches for importance sampling biasing are no longer efficient. The reason is that they assume that the performance of a single resource significantly constrains the system performance, and under this assumption, the parameters can be biased with respect to the bottleneck resource only.

A new *adaptive biasing technique* is defined for dynamically setting up the simulation parameters in the importance sampling experiment. This is the major contribution of this thesis, and it has been successfully applied to several networks. The basic idea is to change the simulation parameters to make the simulation process move toward the parts of the state space where the most important rare events occur. Because this importance depends on the current state $\underset{\sim}{\omega}$, the change of parameters is adapted to the state changes in the simulation process.

The networks used for feasibility demonstration are offered traffic from (i) users with different resource capacities and traffic parameters, (ii) users with and without alternative routing strategies, and (iii) users with different preemptive priority levels and a network with a link failure. The simulation results are validated by comparison with exact results, rough dimensioning rules, and correctness indicators given by the observed likelihood ratio.

Preface

May 5th 1898, Poul Heegaard[1] defended his doctoral thesis. This was a product of his research during the period ’94-’98. The history is now repeated 100 years later, except for the exact date and, of course, the contents of the thesis.

My doctoral study has focused on rare event simulation, and in particular on the application of importance sampling. In my diploma thesis, nearly 10 years ago, I first investigated this technique. Then everything looked simple. However, after several projects during my time as research scientist at SINTEF, numerous problems with the application of importance sampling were experienced. This motivated me to start an in-depth study on this topic, and when Telenor R&D generously made a 3 year scholarship available, I seized the opportunity.

I am indebted to number of peoples for their assistance and encouragements. First of all I want to thank Prof. Bjarne E. Helvik at NTNU for his invaluable contributions to my research. He first introduced me to the exciting topic of rare event simulation when I was writing my masters thesis. Ever since, he has constantly assisted and encouraged me, first as a colleague at SINTEF, and for the last 4 years acting as my supervisor. I thank him for patiently listening to my questions, discussing ideas, raising critical questions, and commenting on my research papers and reports.

During the winter season ’95-’96, I was fortunate to work with Dr. Ragnar Andreassen, at that time a fellow doctoral student. In a joint effort with my supervisor, we succeeded in implementing importance sampling in trace driven simulations of MPEG coded video sources. The result of this research is found in appendix A.

1. Poul Heegaard (1871-1948). Thesis: *Forstudier til en Teori for de algebraiske Fladers Sammenheng* (in danish). Professor in mathematics at University of Copenhagen 1910-1917, and at Oslo University 1918-1941. He was my great grandfather.

The doctoral study has been conducted at NTNU-Department of Telematics. I would like to thank all my fellow students and faculty staff at this department, and the research scientists at SINTEF - Department of Systems Engineering and Telematics, for stimulating discussions. In particular, I would like to thank Arne Folkestad. We have shared office most of the time during the study, and have had an inspiring atmosphere which has helped me through the ups and downs for the past four years.

A special thanks to the editors and the anonymous reviewers for their comments and constructive criticisms of the papers I submitted to *AEÜ, Special issue on Rare Event Simulation*. Parts of these papers are included my thesis.

Last, but not least, I will thank all my family and friends for giving me a social life outside the research community. In particular, my wife and two daughters, to whom I have dedicated this thesis, for their love, support and patience. They have constantly reminded me that there are much more important things in life than completion of a doctoral thesis!

Poul E. Heegaard

Trondheim, 30th April 1998.

List of publications

The research results from this doctoral study have been partly published in journals and presented at conferences. The following is the complete list of publications in the period 1994-98.

Journals

Poul E. Heegaard. *A Scheme for Adaptive Biasing in Importance sampling*. To appear in AEÜ International Journal of Electronics and Communication. Special issue on Rare Event Simulation.

Poul E. Heegaard: *Speed-up techniques for high-performance evaluation*. Telektronikk, vol. 91(2/3), pp 195-207, 1995. Special issue on teletraffic.

In conference proceedings

Poul E. Heegaard. *A survey of Speedup simulation techniques*. Workshop on Rare Event Simulation. Aachen, Germany, 28-29 Aug., 1997. Session 1: Opening session & System Reliability.

Poul E. Heegaard. *A Scheme for Adaptive Biasing in Importance sampling*. Workshop on Rare Event Simulation. Aachen, Germany, 28-29 Aug., 1997. Session 4: Tandem Queues and Networks.

Poul E. Heegaard. *Efficient simulation of network performance by importance sampling*. 15th International Teletraffic congress - ITC 15. Washington D.C., USA, 23-27 June, 1997. Session: Simulation, Measurement and Data Analysis

Poul E. Heegaard: *Adaptive Optimisation of Importance Sampling for Multi-Dimensional State Space Models with Irregular Resource Boundaries*. In P. Emstad,B. Helvik, and

A. Myskja, editors, The 13'th Nordic Teletraffic Seminar (NTS-13), pages 176-189, Trondheim, Norway, 20-22 August 1996.

Ragnar Ø. Andreassen, Poul E. Heegaard, Bjarne E. Helvik: *Importance Sampling for Speed-up Simulation of Heterogeneous MPEG Sources*. In P. Emstad,B. Helvik, and A. Myskja, editors, The 13'th Nordic Teletraffic Seminar (NTS-13), pages 190-203, Trondheim, Norway, 20-22 August 1996.

Poul E. Heegaard: *Rare event provoking simulation techniques*. In proceeding of the International Teletraffic Seminar (ITS), 28 Nov.-1 Dec. 1995 in Bangkok, Thailand. Session III: Performance Analysis I.

Poul E. Heegaard: *Comparison of speed-up techniques for simulation*. In proceeding of the 12. Nordiske Teletrafikk Seminar (NTS-12), 22-24 August 1995 in Helsinki, Finland. Session 8: Intelligent Network Performance.

Bjarne E. Helvik and Poul E. Heegaard. *A Technique for Measuring Rare Cell Losses in ATM Systems*. STF40 A94083. Reprint from 14th International Teletraffic congress - ITC 14. Antibes Juan-les-Pins, France 6-10 June, 1994. Session D31: B-ISDN Network Management and Design. pp 917-930.

Poster and PhD colloquium

Poul E. Heegaard: *Efficient simulation of network performance*. Poster. 1997 Winter Simulation Conference, Atlanta, Dec. 7-10, 1997.

Poul E. Heegaard: *Adaptive Biasing of Importance Sampling Parameters in Simulation of Telecommunication Network*. PhD colloquium. 1997 Winter Simulation Conference, Atlanta, Dec. 7-10,1997.

Table of contents

1
Introduction

1.1 Background and motivation

Today's society has become very dependent on telecommunication systems in everyday life. A network breakdown may cause severe consequences, even in a short breakdown period. For instance, AT&T experienced a breakdown in 1990 that reduced the capacity to the half of long distance, international, and toll free calls all over the U.S.A. for more than 9 hours [Fit90]. A direct consequence of this dependency is that the users will expect and should require that the quality of the services offered by a telecommunication system must be high. Hence, the network operators and service providers must make sure that the network are adjusted to the changes in the number of subscribers, in the offered services, and in the service usage patterns. A sufficient number of network resources must be available, and "intelligent" access and routing procedures must be applied. Furthermore, the communication must be reliable and secure. However, for the user this is a trade-off between the quality of service and the price he is paying, and for the providers between the revenue and cost. This implies that it is not in the interest of the network operators and service providers to add more resources or dependability mechanisms than absolutely necessary in order to reduce the cost to a minimum. This has become increasingly important now after the opening of the telecom market where a strong competition between different providers is introduced. It is very important, both for the users and the providers, to be able to evaluate a network with respect to performance measures like the blocking probability, loss, resource utilisation and availability, mean time to failure, grade of service, end-to-end delays, etc. Such measures are important input to obtain optimal dimensioning, to provide fair and robust access mechanisms, sufficient redundancy, optimal routing strategies, etc.

To evaluate the performance of the network systems, a model is required. This model must include the mechanisms described above, in addition to the users that offer traffic to the network. The users are characterised by attributes like arrival rate, call duration, priority

level, origin and destination nodes, connection routing, bandwidth requirements, etc. Users with the same attributes constitute a group denoted a *user type*. Since the number of attributes are large, the number of different user types will become large. Each user type will typically be modelled separately, and be represented by a dimension in the state space of the model. Hence, the model of a network is multidimensional with a large number of dimensions. The property of interest in a network is related to the users utilisation of the network resources such as links and nodes. The capacity of these resources imposes restrictions to the model as boundaries in the state space. Examples of resources are a communication channel on a link, the bandwidth of a link, etc. To be able to distinguish between the users with different quality of service requirements, a preemptive priority mechanism is needed. This adds new complexity to the model, and it become even more complex when the users are given alternative routes through the network.

In a well engineered network, the resource utilisation is fairly balanced. This implies that in the multidimensional model of a network, all boundaries must be considered when the system performance is evaluated. This makes the model large and complex, and hence numerical solutions become intractable. Another challenge, with respect to performance evaluation, is due to the users restrictive quality of service requirements. For instance, the cell loss ratio in ATM should be very small, less than $<10^{-9}$. When conducting simulations and measurements on such systems, this means that a very large number of events, e.g. cell arrivals, will be simulated or observed, between every occurrence of the events of importance to the network performance, e.g. the loss of a cell. The events of interest is denoted *rare events* because they are very unlikely to occur. When evaluating performance measures dependent on rare events, the simulations and measurements will be inefficient because of the enormous overhead between every event of interest.

The *rare events* in a *multidimensional* model with several boundaries, impose a number of new challenges to the performance evaluation which makes the traditional means insufficient:

- *Analytical (numerical) analysis*: The computations may be very effective if the size of the model is moderate. The modelling requires a high level of abstraction, which involves considerable efforts, skills, and system knowledge to make a tractable and realistic model. For computer and communication networks, queuing models are typically applied for performance evaluation [Lav83]. However, when the size and complexity of the performance models of telecom systems is large, this is a formidable, and in many cases, unattainable task. Another pitfall is the risk of making oversimpli-

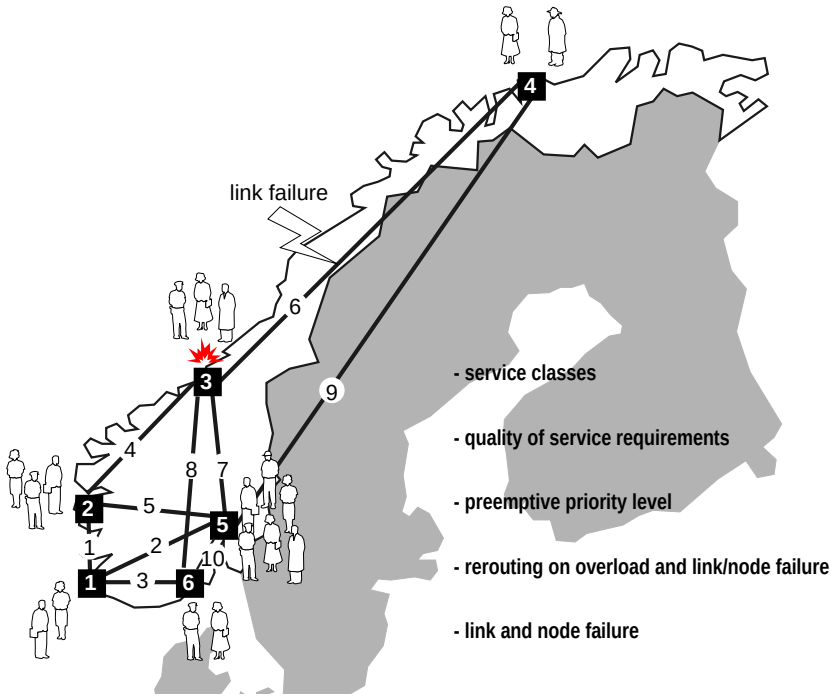


Figure 1.1 : System example of a typical complexity.

fied assumptions with the result that the model no longer reflects the true nature of the system. Hence, the analytic solution is limited to models with specific structures, which can be solved (numerically) when the size of the model is moderate.

- *Simulations*: To build performance models for simulations, extensive knowledge of the system is required. But, compared to the analytic approach, the models are more flexible in the sense that arbitrary levels of detail are included. This means that all details that affect the performance of interest are included, but nothing more. Computations of these models, i.e. the simulations, are normally more demanding than analytic (numerical) computations. Systems with strict quality of service requirements, and high network performance, are even more demanding. The reason is simply that the system has a high event activity (e.g. many packet arrivals, or call setups) relative to the occurrence of service degradation (e.g. loss of packets, or call blocking). Hence, an enormous number of events must be simulated for each rare event that influences the performance measure. Several rare events are required to achieve a certain confidence in the estimates.

- *Measurements*: No modelling is required, but a real system, or at least a prototype, must exist. This adds a considerable cost to the experiment. Measurements are limited to evaluation of the properties that can be recorded by the equipment. Furthermore, it is difficult to control the measurement, both with respect to the load offered to the system, and the internal state of the system. As a consequence of this, it is difficult to reproduce the results from a measurement experiment. In a controlled laboratory, for instance the B-lab at the Norwegian Telecom Research [Orm91], it is possible to reduce these limitations to some extent. The B-lab has defined a test environment where ATM (Asynchronous Transfer Mode) equipment is offered artificial, but realistic, traffic by the Synthesized Traffic Generator (STG) [HMM93]. The evaluation of properties dependent on rare events is still a problem, even if the number of events generated per time unit is normally much higher in a measurement experiment than it is for simulations.

In this thesis, the *simulation approach* is taken because of its flexibility and expressive modelling power. The main focus is speedup techniques for simulation of the network performance dependent on *rare events* in the *multidimensional* model with several boundaries.

1.2 Speedup simulation techniques

Several techniques are known in the literature that will reduce the required simulation time more or less significantly. A *speedup simulation technique* refers to any technique that reduces the computational effort, compared to direct simulations, that is required to produce an estimate with a specific level of accuracy. Figure 1.2 identifies some techniques that are applied to speedup discrete event simulations:

- *Parallel and distributed simulation,*
- *Hybrid techniques,*
- *Variance reduction by use of correlation,*
- *Rare event provoking techniques.*

These techniques will be presented in chapter 2. The techniques are not mutually exclusive, and hence they can, and should, be combined to achieve additional speedup.

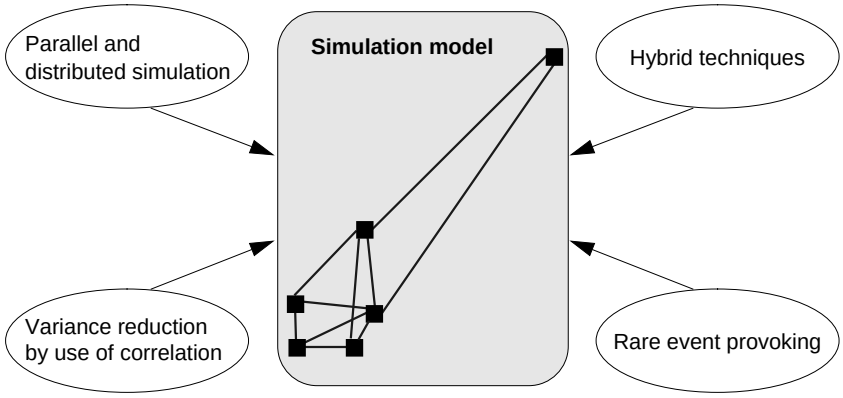


Figure 1.2 : An overview of speed-up simulation techniques.

1.3 Rare event simulation of network models

For the purpose of speeding up the simulations of networks with rare events, a rare event provoking approach must be applied. In figure 1.3, an example of a simulation speedup is given which compares direct simulation with *importance sampling*. A significant speedup in simulation efficiency is observed.

Two techniques have been described in the literature with a speedup similar to this:

- *RESTART*[1]*/splitting*: the simulation process is split when visiting a state related to a certain threshold. At this state, several replicas of the process are generated.

- *Importance sampling*: the parameters of the simulation process are changed to make the rare events less rare.

The techniques apply two different approaches to manipulate the underlying simulation process. The differences between RESTART and importance sampling will briefly be described in chapter 2. In addition, a few simulation comparisons are made between separate experiments of RESTART and importance sampling, and the combination of the two.

1. REpetitive Simulaiton Trials After Reaching Thresholds (RESTART).

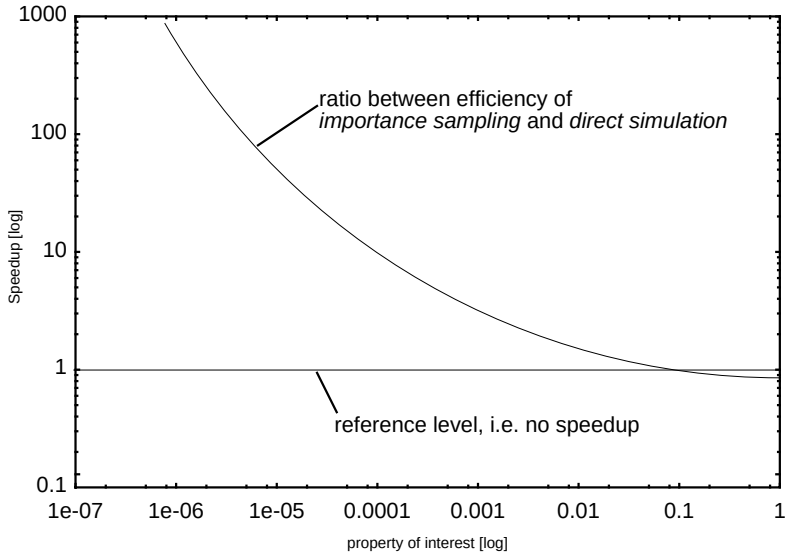


Figure 1.3 : Speedup of importance sampling over direct simulation.

The impressive speedups reported in the literature are mainly observations made from simulations on simple, one dimensional models. To the author's best knowledge, no results have yet been reported on efficient simulation of rare events in network systems with balanced utilisation of the resources. With such models, both the rare event provoking techniques face severe problems:

- *RESTART/splitting*: at what states should the simulation process be split, and how many replicas should be made?

- *Importance sampling*: how can the parameters of the simulation process be changed when the rare events occur in very different parts of the state space?

The focus in this thesis is on application of importance sampling in network systems with balanced utilisation of the resources. The basics of how to change the parameters of the simulation process in a simple model is presented in chapter 3. The new and adaptive parameter biasing is presented in chapter 5. In chapter 6, this basing is applied to network systems that are modelled by the framework described in chapter 4.

1.4 Main focus of research in this thesis

In this thesis it is assumed that *discrete event simulations* are conducted. The underlying simulation process is a *continuous time Markov chain*, which can be substituted by an embedded *discrete time Markov chain* for simulation of steady state behaviour.

As mentioned in section 1.1, a model of a telecommunication network is a *multidimensional* model because a number of users with different traffic parameters and requirements have to be considered. The network resources introduce the boundaries of the model. The system performance is associate with these boundaries. When the resource utilisation is balanced, the biasing of importance sampling parameters is no longer trivial. The research focus in this thesis in on application of *importance sampling* simulation of *rare events* in *multidimensional* models with balanced resource utilisation.

To describe the telecommunication network, a modelling framework is required. It is assumed that a set of user types are responsible for the traffic offered to the network. They have different traffic parameters, different preemptive priorities, alternative routing strategies, and can be exposed to link or node failures. Furthermore, it is assumed that a call from one of the users, sets up a connection, which is either a circuit switched connection in a connection-oriented network, or an equivalent bandwidth in a connection-less network.

Within this framework, a new, adaptive biasing technique is developed, that enables the use of importance sampling in multidimensional models with balanced utilisation of the resources. Several simulation experiments are conducted to demonstrate the feasibility of this technique.

1.5 Other importance sampling applications

In addition to the main focus summarised in the previous section, the author has been involved in two other activities where importance sampling is applied:

1. *Framework for accelerated measurements and simulation of ATM equipment*. Several speedup techniques were combined, and importance sampling was applied to change the parameters of the traffic sources offering load to test ATM equipment. The source models used in the Synthesized Traffic Generator (STG) [HMM93] are described over several time scales, see [Hel95]. It is the *burst level* parameters that are changed to increase the load offered to the ATM equipment. The equipment in itself is unchanged

because it is not accessible. This means that internal buffer capacities and service rates cannot be altered. The framework is tested by simulations and is not yet implemented on the STG. The result of this work is reported in [HH94].

2. *Multiplexing of MPEG coded video sources*. A new trace driven simulation technique is developed which is prepared for evaluation of cell losses in ATM buffers loaded by a large number of heterogeneous MPEG coded video sources. Statistically firm results are obtained, within a reasonable computation effort and time, by applying a special importance sampling approach. The properties of the technique are examined and compared to a previously suggested stratified sampling technique. The capabilities of the technique are demonstrated by simulation of 76 sources of nineteen different MPEG VBR video source types with cell losses in the 10^{-9} - 10^{-12} domain. The results of this work are reported in [AHH96]. A reprint of this paper is given in appendix A for the sake of completeness. A description can also be found in [And97].

As mentioned above, importance sampling biasing is very model dependent. In general, it is not feasible to use previous results to new systems, unless the same modelling framework is applied.

1.6 Guide to the thesis

Chapter 2 gives a brief overview of some of the speedup simulation techniques that are applicable to discrete event simulations. The overview is not limited to rare event simulation techniques because it is known that several simulation techniques should be combined to increase the speedup.

Importance sampling is the rare event simulation technique that has been given the main focus in this thesis. In chapter 3, the technique is briefly described, and a detailed discussion is given on the main challenge to make importance sampling efficient, namely the change of measure or biasing of the simulation parameters. The known, asymptotically optimal, change of measures that exist for simple models are briefly mentioned.

However, for more complicated models, e.g. models of communication networks, current importance sampling biasing do not suffice. In chapter 4, a description is given of a flexible modelling framework which allows both traffic and dependability aspects to be included. Chapter 5 describes a new, adaptive parameter biasing that enables efficient simulation of such networks.

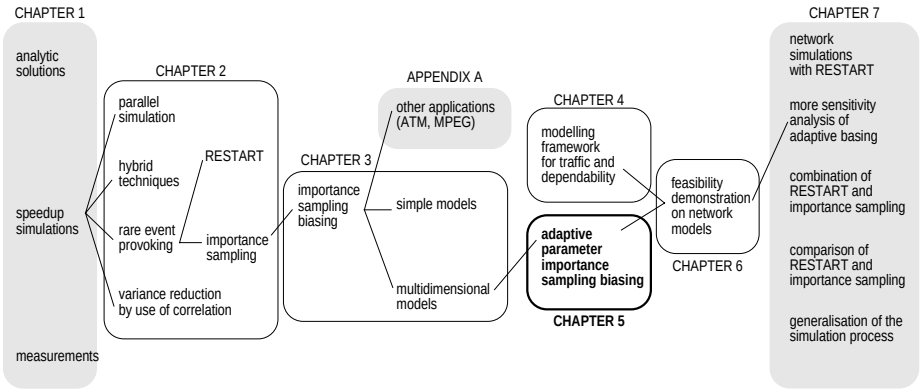


Figure 1.4 : An outline of the thesis.

In chapter 6, the feasibility of the modelling framework and the adaptive biasing is demonstrated on a few network models. The users have different service requirements, they have preemptive priorities to allow quality of service differentiation, and the network provides alternative routing to some users when the primary route is blocked. A route is blocked either when an overload has occurred, or after a link or node failure.

Still, some work remains on the details in the adaptive technique, but also on learning more about the feasibility of this approach. A list of further work and concluding remarks is found in chapter 7.

The importance sampling is applied to speedup the simulation of a multiplexer of MPEG coded video sources. A paper was written on this work which is reprinted in appendix A for the sake of completeness. The results was a joint effort with Prof. Bjarne Helvik, NTNU, and Dr. Ragnar Andreassen, Telenor R&D.

A list of symbols and notations used in this thesis can be found in appendix B along with an overview of the concepts of the modelling framework presented in chapter 4. The other appendices describe details in the derivation of expressions and plots used in the main parts of the thesis.

2

Speedup techniques for discrete event simulations

Several techniques have been proposed in the literature for speeding up simulation experiments. This chapter gives a brief overview of some of these techniques applicable to discrete event simulations.

Of particular interest are techniques for speeding up simulation of systems with properties dependent on rare events. This chapter is not limited to the rare event provoking techniques, but also includes other techniques that can be combined and give increased speedup This chapter is based on [Hee95c] and [Hee97b].

2.1 Settings

For description of the various techniques in this chapter, the general settings and the required notation are introduced.

The performance evaluation is steady state blocking and system unavailability. Let the property of interest be denoted γ. This is the expected value $\gamma = E(g(\underset{\sim}{s}))$ of a response function $g(\underset{\sim}{s})$, taking samples, $\underset{\sim}{s}$ from the density distribution $f(\underset{\sim}{s})$.

Example 2.1: Consider a single server queue. The property of interest, γ, is the probability of blocking in a busy period of this queue. A busy period is the time between two time epochs where the queue is empty. A sample $\underset{\sim}{s}_r$ is the observed sequence of call arrivals and departures during busy period r. The response function $g(\underset{\sim}{s}_r)$ is 1 if the queue is blocked in the rth busy period and 0 otherwise.

The samples are independent and identically distributed. In a regenerative simulation a sample $\underset{\sim}{s}$ is a sequence of n_r events constituting the rth regenerative cycle, i.e.

$$\underset{\sim}{s}_r = \{\underset{\sim}{\omega}_i\}_{i=0}^{n_r}, \tag{2.1}$$

where $\underset{\sim}{\omega}_i$ is the system state after event i. An unbiased estimate of γ is then:

$$\hat{\gamma} = \frac{1}{R}\sum_{r=1}^{R} g(\underset{\sim}{s}_r) \tag{2.2}$$

where R is the number of samples $\underset{\sim}{s}$. An *event* is either an arrival or departure of an *entity* (e.g. a customer) to the queue. The variance of this estimate $\hat{\gamma}$ is:

$$\mathrm{Var}_f(\hat{\gamma}) = \frac{1}{R}\mathrm{Var}_f(g(\underset{\sim}{s}_r)). \tag{2.3}$$

As an example, consider the problem of estimating $\gamma = P(A)$, where A is a specific event. If $P(A) \ll 1$ then A is denoted a *rare event*. Consider A to be e.g. the event that the number of customers in a single server queue reach the capacity N during a busy period.

Substituting $g(\underset{\sim}{s})$ by $I_r(A)$ in (2.2), then:

$$\hat{\gamma} = \frac{1}{R}\sum_{r=1}^{R} I_r(A) \tag{2.4}$$

where

$$I_r(A) = \begin{cases} 1 & \text{if event } A \text{ has occured in cycle } r \\ 0 & \text{otherwise} \end{cases}. \tag{2.5}$$

This can also be expressed as

$$I_r(A) = I(\underset{\sim}{s}_r \in \underset{\sim}{\Omega}_1). \tag{2.6}$$

which gives the relation between the subspace $\underset{\sim}{\Omega}_1$ where the rare events of interest are observed, and the rth sample. This $\underset{\sim}{\Omega}_1$ is denoted a *target* subspace. A visit to $\underset{\sim}{\Omega}_1$ is the event A of interest. For details on the simulation process and the modelling framework, see the description in chapter 4. In appendix B, a list of concepts and symbols is given.

Example 2.2: Consider the queue from example 2.1. The system state, $\underset{\sim}{\omega}$, is the number of customers in the system at a given point in time. The regenerative state space is the

empty system, $\underset{\sim}{\Omega}_0 = \{0\}$, while the target subspace is the full system, $\underset{\sim}{\Omega}_1 = \{N\}$. Each simulated cycle starts and ends in the regenerative state, i.e. $\underset{\sim}{\omega}_0 = \underset{\sim}{\omega}_{n_r} = \{0\}$

It can easily be shown that $\hat{\gamma}$ is an unbiased estimator of γ (the samples $\underset{\sim}{s}$, and hence the events A, are independent, see (2.6)):

$$E_f(\hat{\gamma}) = E_f(\frac{1}{R}\sum_{r=1}^{R} I_r(A)) = \frac{1}{R}\sum_{r=1}^{R} E_f(I_r(A)) = \frac{1}{R}R \cdot P(A) = \gamma . \tag{2.7}$$

The variance is:

$$\mathrm{Var}_f(\hat{\gamma}) = \frac{1}{R}\mathrm{Var}_f(I_r(A)) = \frac{\gamma(1-\gamma)}{R}. \tag{2.8}$$

To retain a certain relative error of the estimates, consider the square of the relative error, $\mathrm{Var}_f(\hat{\gamma})/E_f(\hat{\gamma})^2 \le \varepsilon$. When γ changes, the following number of samples R is required:

$$R \ge \frac{1}{\varepsilon}\left(\frac{1}{\gamma} - 1\right). \tag{2.9}$$

Hence, $R \propto 1/\gamma$, and the simulation efficiency drops dramatically as γ decreases.

As a second example, consider the problem of estimating the steady state probability of state Ω_1. Recall that A is a binomial event that takes on values 0 or 1, and hence the following estimate applies (“Renewal theorem”)

$$\hat{\theta} = \frac{E(\tau_1)}{E(\tau)} = \frac{E(\tau_1 | A) \cdot P(A)}{E(\tau)} \tag{2.10}$$

where $E(\tau_1)$ is the expected time in state Ω_1 in a cycle, and $E(\tau)$ is the expected cycle time. This estimator depends on observations of the same events A as equation (2.4). Hence, (2.10) meets the same challenges as (2.4) with respect to simulation efficiency.

The *simulation efficiency* is defined in terms of the CPU time and the variance. The following measurement will be used in this thesis (the reciprocal of the measure used in [Hee95a] and in (A.16)):

$$m = 1/(t_{\mathrm{cpu}} \cdot \mathrm{Var}(\hat{\gamma})) \tag{2.11}$$

where t_{cpu} is the CPU time required to obtain an estimate $\hat{\gamma}$ with variance $\text{Var}(\hat{\gamma})$. In some sections, an index is added to this estimator to indicate the method used to obtain this estimate, e.g. RESTART (R), importance sampling (IS).

2.2 The need of speedup simulation

Simulation is considered to be a flexible means for performance evaluation of complex data and telecommunication networks. However, when the networks have very strict quality of service requirements, the direct simulation approach is very inefficient. The reason is that the performance measure, e.g. the cell loss probability, depends on rare events to occur, e.g. cell losses (in ATM with probability typically less than $< 10^{-9}$).

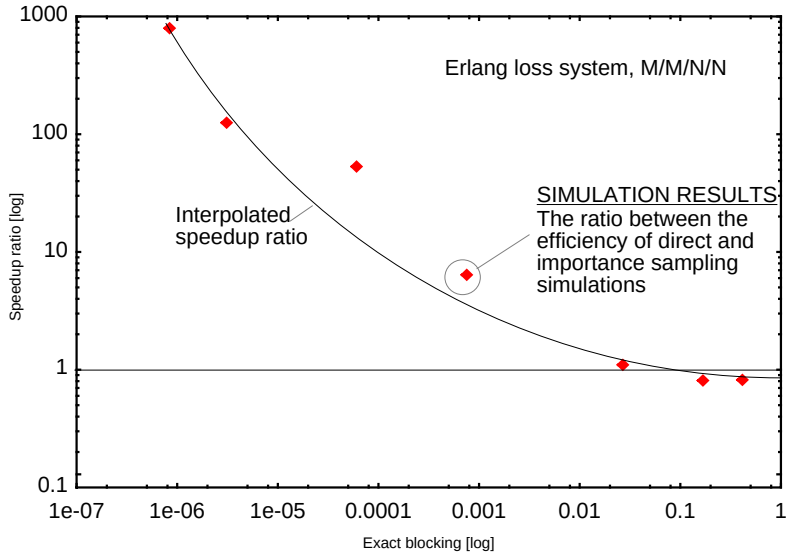


Figure 2.1 : Speedup of importance sampling over direct simulation.

Figure 2.1 illustrates the speedup of importance sampling over direct simulation on a simple Erlang loss system. The number of samples required for direct simulation to retain the same confidence level of the estimates, increases exponentially as the probability of the rare event increases. Using importance sampling (with optimal parameters) the required number is unchanged as long as the model size is unchanged, and hence a significant speedup is observed. Note that already at a loss probability of approximately 5%, importance sampling will, in this example, increase the efficiency.

Importance sampling is not always this efficient, and hence other techniques must also be considered, either as alternatives, as supplements or in combination with importance sampling.

2.3 Overview

This chapter contains several techniques that increase the simulation efficiency, m, relative to an ordinary discrete event simulation experiment. Two orthogonal approaches exist, and the different effect on the original discrete sequence of events is illustrated in figure 2.2(a):

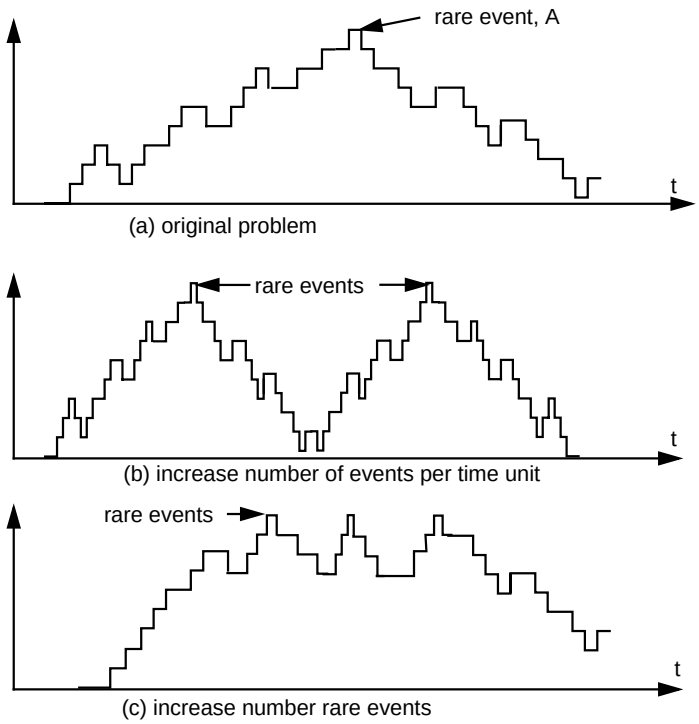


Figure 2.2 : Illustration of sequence of events over real simulation time scale for different speedup approaches.

1. Increase the number of events per time units, either by using a faster machine, see figure 2.2(b), or doing computations in parallel on several processors.

2. Decrease the simulation overhead, i.e. increase the relative number of events of interests, by exploitation of some statistical property of the simulation model, see figure 2.2(c).

In the following sections, a few of the known speedup techniques are included and key references are given. All techniques belong to either of the two orthogonal approaches above. Figure 1.2 contains an overview of the categories used in this chapter:

- *Parallel and distributed simulation* exploits either special hardware and software, or the existence of a cluster of workstations in the organisation. These techniques require some extra administration, apart from expert skills in parallel programming. Hence, only rather computer demanding problems should consider a parallel and distributed simulation approach.

- *Hybrid techniques* combine analytic results with simulation experiments of subproblems.

- *Variance reduction by use of correlation* takes advantage of a known correlation between output and input samples, or introduces such correlation by a controlled simulation setup. This correlation enables variance reduction. These techniques are not efficient enough when the estimates depend on rare events to occur, but can be combined with other techniques.

- *Rare event provoking* techniques include importance sampling and RESTART/splitting. Both techniques increase the frequency of the rare events of interest, but apply two different approaches.

For surveys on variance reductions and speedup simulations, see e.g. [FLS88, KM88, McG92, Hee95c, Hee97b].

2.4 Parallel and distributed simulation

In many computer environments, a large number of processors are available, either as a cluster of workstations or a multiprocessor machine architecture. The *parallel and distributed simulation techniques* exploit such a multiprocessor environment to increase the simulation speedup. The speedups are due to an increase in the number of events per time unit, see figure 2.2(b). The speedup is limited by the number of processors, P, and the technique applied [Lin94]:

- *PADS - P*arallel *A*nd *D*istributed *S*imulation splits the sequential simulation process into parallel subprocesses which are distributed on P processors,

- *PIRS - P*arallel *I*ndependent *R*eplicated *S*imulation makes P independent replications of the sequential simulation process and runs each of them on P separate processors.

These two different approaches are illustrated in figure 2.3.

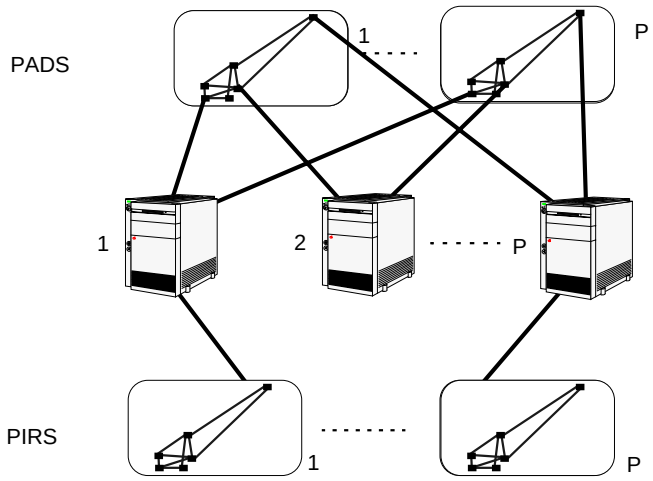


Figure 2.3 : PADS vs. PIRS.

2.4.1 Parallel And Distributed Simulation (PADS)

The PADS identifies sub-models in the sequential simulation model which can be evaluated in parallel on different processors, e.g. in a network model each queue, trunkline or user type can be considered as separate parallel sub-models. In general, it is not simple to identify sub-models that can be run in parallel as separate processes and give significant over direct simulation. The problem is that they are not independent, and hence the processes are slowed down due to delay at interaction points. Several techniques for handling these synchronizations are proposed [Fuj90, KM88]:

- *Conservative synchronisation* approaches assume that all synchronisation points result in an interaction (e.g. exchange of data), and hence the processes cannot proceed until the slowest process has catched up [KRWW93].

- *Optimistic synchronisation* approaches, e.g. used in the Time Warp technique [Jef85], assume that no potential synchronisation point actually results in an interaction and hence the sequence will proceed immediately. If it is discovered that an interaction should have taken place, the subprocess(es) must roll back.

- *Time driven* approaches differ from the event driven approaches above [LR85]. A global clock is incremented on fixed time instances rather than on events. In synchronized networks this has shown a significant speedup.

Due to the slowdown caused by synchronisation, irrespective of which approach that is taken, the total speedup will always be less than the number of processors involved, <P.

2.4.2 Parallel Independent Replicated Simulation (PIRS)

Every stochastic simulation experiment need a number of independent observations to obtain a certain confidence of the estimates. PIRS is a framework to get more observations in shorter time simply by distributing several replicas of the sequential simulation process on P processors. Apart from the gathering of data from different machines, and the post-processing of the final report, the speedup will be P, equal to the number of processors applied.

2.4.3 Pros and cons

The obvious conclusion from the following pros and cons list is that PIRS has a wide applicability in stochastic simulation experiments, while PADS should limit its applications to real time simulations with extensive computation need, such as training simulations for pilot education. See [Lin94] for some comments on PADS, and [NH97] for a framework assisting parallelisation of sequential simulation processes.

2.5 Hybrid techniques

A hybrid technique is any technique that combines analytic results with simulation. Two main approaches exist [FLS88, KM88, LO88]:

- *Conditional sampling* - uses simulation to provide conditions to a given analytic model.

Table 2.1: Pros and cons for PADS and PIRS.

	PADS	PIRS
Special hardware?	Can be exploited	Not applicable
Special software?	Required	Not required
Expert skills required?	Both in parallel programming and simulation	Simulation only
Amount of processing of given problem	Should be large to advocate investment	Any size
Speed up	<P	≈P

- *Decomposition* - identifies independent sub-models (in time or space) for which separate evaluations by means of analytic and simulation approaches are possible.

Both approaches attempt to use analytic results to reduce the variance. They are very model dependent, and the flexibility of simulation models might be reduced to adjust to analytic model assumptions.

2.5.1 Conditional sampling

In conditional sampling, a mathematical approach is taken to reduce the variance by a known functional relation between two random variables, i.e. conditional arguments, conditional expectation, and substructures of the random variables and system response.

Let the expected system response, $g(\underline{s})$ from (2.2), be known if a fixed value of the stochastic variable X is given, i.e. $E(g(\underline{s}|X = x))$. It can be shown that if γ is estimated by taking samples X from $h(x)$ and using the following estimator:

$$\hat{\gamma}_{\mathrm{CS}} = \frac{1}{R}\sum_{r=1}^{R} E(g(\underline{s}|X = x_r)) \tag{2.12}$$

the variance is reduced compared to the direct estimator from (2.2). The estimator is unbiased as the expectation is $E(\hat{\gamma}_{\mathrm{CS}}) = E_h(E_f(g(\underline{s}|X = x))) = E_f(g(\underline{s})) = \gamma$.

The variance is [LO88]:

$$\mathrm{Var}_h(\hat{\gamma}_{\mathrm{CS}}) = \frac{1}{R}\mathrm{Var}_h(E_f(g(\underline{s}|X = x))) = \frac{1}{R}[\mathrm{Var}_f(g(\underline{s})) - E_h(\mathrm{Var}_f(g(\underline{s})|X))]. \tag{2.13}$$

Subtracting (2.13) from (2.3):

$$\mathrm{Var}_f(\hat{\gamma}) - \mathrm{Var}_h(\hat{\gamma}_{\mathrm{CS}}) = \frac{1}{R} E_h(\mathrm{Var}_f(g(\underline{s}|X))) \geq 0 \tag{2.14}$$

then $\mathrm{Var}_f(\hat{\gamma}) \geq \mathrm{Var}_h(\hat{\gamma}_{\mathrm{CS}})$, and hence the variance of $\hat{\gamma}_{\mathrm{CS}}$ is always less than or equal to the variance of $\hat{\gamma}$. Unfortunately, it is often impossible, or at least very difficult, to find a suitable conditioning quantity.

2.5.2 Decomposition

Alternatively, an engineering approach can be taken where the model is decomposed into sub-models, either in time or space. The basic idea is to identify sub-models that need to be evaluated only once, either by analytic solutions or by simulation. The speedup is partly due to the use of analytic results instead of simulations, but mainly due to the reduction of simulation overhead since unnecessary repeated simulations of subsystem are avoided. Figure 2.4 illustrates a hierarchical decomposition over two different time scales. This gives a speedup because unnecessary repeated simulations on low time granularity levels can be avoided, and hence the overhead is reduced.

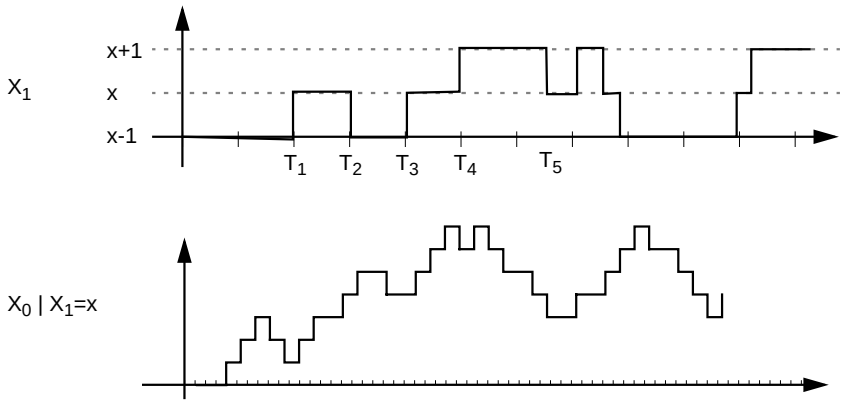


Figure 2.4 : Hierarchical decomposition of system with different time granularity.

Example 2.3: The X_1 in figure 2.4 can be considered to be the number users that are statistically multiplexed on a virtual connection (VC) in ATM. The users are transmitting for a period of typically seconds or minutes. X_0 can be considered to be the number of ATM cells that are transmitted on this VC, given a number $X_1 = x$ users being active.

On a 155 Mbps channel, the cell period is in order of μ-sec. This means that an enormous number of cell periods are generated between every change in the number of multiplexed users.

A sample $\underset{\sim}{s}$ consists of two parts: one part from sub-models with a given (analytic) result, $\underset{\sim}{s}_a$, and another part from simulation models, $\underset{\sim}{s}_s$. The sampling distribution is now $f(\underset{\sim}{s}_a, \underset{\sim}{s}_s)$. To simplify, assume product form, i.e the sub-models to be independent:

$$f(\underset{\sim}{s}_a, \underset{\sim}{s}_s) = f_a(\underset{\sim}{s}_a) \cdot f_s(\underset{\sim}{s}_s). \tag{2.15}$$

An unbiased estimator for γ is:

$$\hat{\gamma}_{\mathrm{HD}} = \frac{1}{R}\sum_{r=1}^{R} g((\underset{\sim}{s}_a)_r) \cdot g((\underset{\sim}{s}_s)_r) \tag{2.16}$$

with variance

$$\mathrm{Var}_{f_a f_s}(\hat{\gamma}_{\mathrm{HD}}) = \frac{1}{R}[\mathrm{Var}_{f_a}(g(\underset{\sim}{s}_a)) + \mathrm{Var}_{f_s}(g(\underset{\sim}{s}_s))] = \frac{1}{R}\mathrm{Var}_{f_s}(g(\underset{\sim}{s}_s)) \tag{2.17}$$

where $\mathrm{Var}_{f_a}(g(\underset{\sim}{s}_a)) = 0$ because $g(\underset{\sim}{s}_a)$ is a fixed response from an analytic result. Hence, equation (2.17) shows that some of the variance reduction is achieved through the use of fixed system response for parts of the system model, instead of only simulations.

2.6 Variance reduction by use of correlation

The traditional variance reduction techniques exploit a known or introduced correlation in input and output samples. The most familiar ones are often described in general textbooks of simulations, [LO88, BFS87]:

- *Antithetic variates* - form output samples as the mean value of two complementary negatively correlated input samples.

- *Common random numbers* - use the same sequence of input samples to induce a correlation in the output samples, particularly suitable for performing comparison simulations.

- *Control variables* - reduce the variance by relating the quantity of interest to a strongly correlated random variable with a known expectation.

The common challenge for all these techniques is to gain sufficient insight in the system behaviour to be able to identify correlated controls, and to foresee the consequences of introducing correlation. If negative correlation is expected, but positive correlation is experienced, the result is a variance *increase* instead of decrease. This is a serious problem for antithetic variates and common random numbers when these are applied to network models of realistic complexity. The reason is that it is very hard to establish a relationship between the input samples and the output (response) samples [Hee95c]. Hence, in the following section, only control variables will be described in more detail.

2.6.1 Control variables

Consider $g(\underset{\sim}{s})$ and $C(\underset{\sim}{s})$ to be two dependent (through $\underset{\sim}{s}$) random variables (system responses) obtained in a simulation experiment. $C(\underset{\sim}{s})$ is a control variable introduced to reduce the variance of the quantity of interest, $g(\underset{\sim}{s})$.

Example 2.4: In figure 2.5, an example of pairs of $g(\underset{\sim}{s}_r)$ and $C(\underset{\sim}{s}_r)$ are plotted as a function of a sample, $\underset{\sim}{s}_r$. The $g(\underset{\sim}{s}_r) = C(\underset{\sim}{s}_r) + \xi_r$, where $C(x) = 1 + \cos(\pi x / 10)$, $\underset{\sim}{s}$ is taken from a uniform distribution $(0, 100)$, and the error term is uniformly distributed $\xi \sim U(-0.33, 0.33)$.

The use of control variables is efficient when the quantity of interest and the controls are strongly correlated as they are in figure 2.5. This means that they have correlated error terms. If the control variable is far from its (known) expectation, then the quantity of interest can also be assumed to be far from its (unknown) mean.

It must be possible to determine the expected value of the control variable, $\theta_C = E(C(\underset{\sim}{s}))$, either by analytic results, by some approximation, or estimated by simulation. An unbiased estimator using *linear control* is then:

$$\hat{\gamma}_{\mathrm{CV}} = \frac{1}{R}\sum_{r=1}^{R}(g(\underset{\sim}{s}_r) + \beta \cdot (C(\underset{\sim}{s}_r) - \theta_C)) \tag{2.18}$$

with variance [LO88]:

$$\mathrm{Var}_f(\hat{\gamma}_{\mathrm{CV}}) = \frac{1}{R}[\mathrm{Var}_f(g(\underset{\sim}{s})) + \beta^2 \mathrm{Var}_f(C(\underset{\sim}{s})) + 2\beta \mathrm{Cov}_f(g(\underset{\sim}{s}), C(\underset{\sim}{s}))]. \tag{2.19}$$

The unknown factor β is chosen to minimise the variance, i.e.

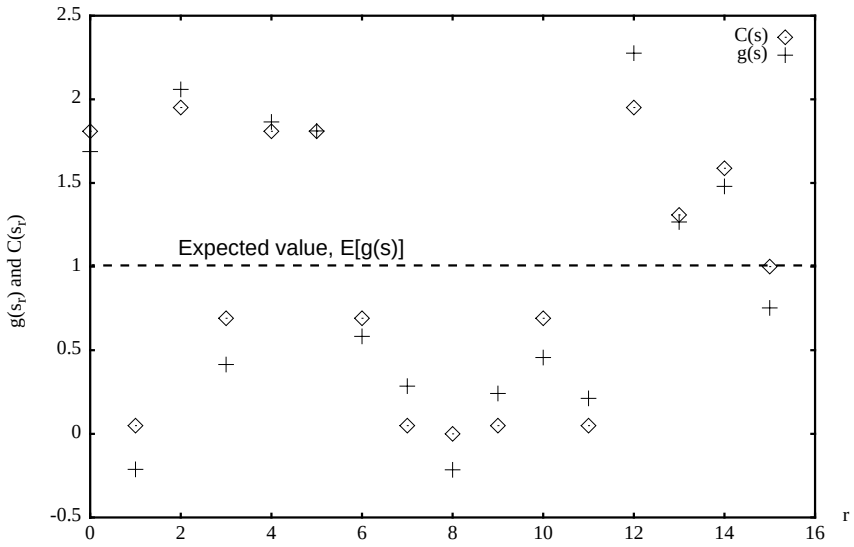


Figure 2.5 : Joint plot of the property of interest and the control variable.

$$\frac{\partial}{\partial \beta}\mathrm{Var}_f(\hat{\gamma}_{\mathrm{CV}}) = 0 \Rightarrow \beta = -\mathrm{Cov}_f(g(\underline{s}), C(\underline{s}))/\mathrm{Var}_f(C(\underline{s}))\,. \tag{2.20}$$

Several control variable techniques apply [LO88]:

- *Simple linear control*, i.e. equation (2.18) with $\beta = -1$,
- *Regression adjusted controls*, i.e. equation (2.18),
- *Multiple controls*, i.e. several control variables are applied, where the control variables should be independent to obtain additional variance reduction.
- *Non-linear controls*, i.e. the control variable(s) and the property of interest are not linearly related. *An example* of an unbiased estimator using non-linear control is:

$$\hat{\gamma}_{\mathrm{CV}} = \frac{1}{R}\sum_{r=1}^{R}(g(\underline{s}_r) + \beta(C^2(\underline{s}_r) - \theta_C^2)) \tag{2.21}$$

with variance [LO88]:

$$\mathrm{Var}_f(\hat{\gamma}_{\mathrm{CV}}) = \frac{1}{R}[\mathrm{Var}_f(g(\underline{s})) + \beta^2 \mathrm{Var}_f(C^2(\underline{s}_r)) + 2\beta \mathrm{Cov}_f(g(\underline{s}), C^2(\underline{s}_r))]. \quad (2.22)$$

The problem with application of control variables is that it is difficult to identify controls that are both correlated to the property of interest, and have a known expectation that is *easily* obtained.

Example 2.5: (example 2.4 continued). The plot in figure 2.5 is part of an experiment of $R = 100$ samples where the estimated correlation factor between $g(\underline{s})$ and $C(\underline{s})$ is $\rho(g(\underline{s}), C(\underline{s})) = 0.96$. In that specific case, the variance was reduced by a factor of 15.5 by using $\hat{\gamma}_{\mathrm{CV}}$ instead of $\hat{\gamma}$ from equation (2.2).

The techniques that reduce the variance by use of correlation are not applicable when the estimates rely on rare events to happen, i.e. $P(A) \ll 1$. The reason is that these techniques do not change the probability of event A. Hence, if A is a rare event, a set containing only 0's might be the result of a simulation. Then, the variance cannot be further reduced and the simulation efficiency is not increased.

2.7 Rare Event Provoking

With a *rare event provoking* technique the speedup is due to changes in the statistical behaviour such that the rare events A are provoked to occur more often. Two main approaches exists:

- *RESTART*[1]*/importance splitting* - identify subspaces from which it is more likely to observe A, and then make replicas of the sequences that reach these subspaces by splitting the simulation process.

- *Importance sampling* - changes the stochastic process to generate sequences of events which make the rare events of interest less rare. The basics of this technique is described in most textbook of simulation that includes a chapter on variance reduction, e.g. [LO88, BFS87].

This section describes these two techniques and shows how they can be combined. Furthermore, this section describes how the underlying sampling distributions are affected by RESTART and importance sampling. In section 2.8, examples of the use of RESTART,

1. REpetitive Simulation Trials After Reaching Thresholds (RESTART).

importance sampling, and the combination of the two, are demonstrated on an M/M/1/N queue and a shared buffer.

2.7.1 RESTART

The basic idea of RESTART is to identify subspaces from which it is more likely to reach the target subspace, i.e. where the rare event A occurs. These subspaces are considered to be *thresholds*. Every time the process reaches a threshold, the current sequence $\underset{\sim}{s}_r$ is split in a number of replicas, all continuing from the splitting state. In this way, the number of rare events will increase, dependent on the number of RESTART thresholds defined and the number of replicas generated. Figure 2.6 illustrates the splitting and restarts of events as a new threshold is reached.

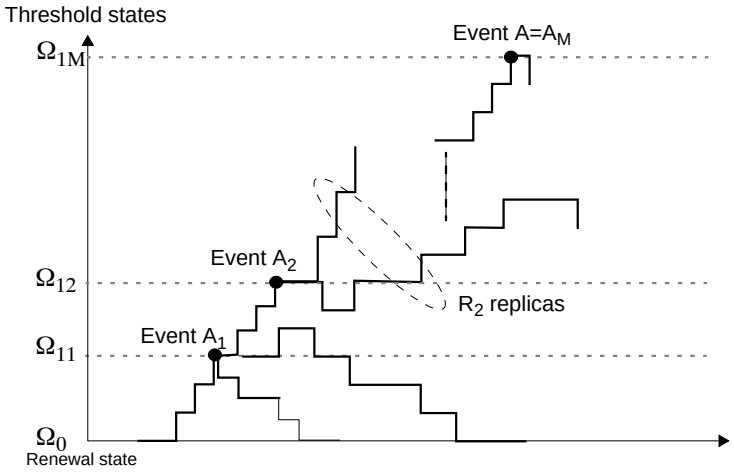


Figure 2.6 : RESTART with splitting at M thresholds.

Consider the event A to be the rare event of interest. Let A_i be the event that state $\underset{\sim}{\Omega}_{1i}$ is visited during cycle $\underset{\sim}{s}_r$. $\underset{\sim}{\Omega}_{1i}$ is the state space constituting threshold i.

For simplicity, substitute $g(\underset{\sim}{s})$ by $I(A)$ in (2.2). Furthermore, define $p_i = P(A_i|A_{i-1})$. Now, $\gamma = P(A)$ can be expressed as (by Bayes formula):

$$\gamma = p_1 p_2 \cdots p_M . \tag{2.23}$$

The *basic* regenerative cycle is the sequence of events, without splitting, that starts from the renewal state Ω_0. This cycle returns a zero-one function $I_r(A_1)$ which is set to 1 if event A_1 occurs, i.e. threshold Ω_{11} is reached, and 0 otherwise. If threshold 1 is reached, i.e. event A_1 has occurred, then R_1 new replicas of the sequence are generated. Each replica is returning $I_r(A_2|\{A_1\})$, $r = 1, \ldots, R_1$. At the nested step m, the corresponding definition is $I_r(A_m|\{A_1, \ldots, A_{m-1}\})$, which is the zero-one function returning 1 if event A_m occurs before returning to Ω_0, and given that events A_1 through A_{m-1} have occurred. Otherwise, $I_r(.) = 0$.

Let $A_i = \{A_1, \ldots, A_i\}$ be the event that all events A_1 through A_i have occurred. An unbiased estimator for γ is then, see [GHSZ96]:

$$\begin{aligned}\hat{\gamma}_R &= \frac{1}{R_1}\sum_{r_1=1}^{R_1} I_{r_1}(A_1)\frac{1}{R_2}\sum_{r_2=1}^{R_2} I_{r_2}(A_2|A_1)\ldots\frac{1}{R_M}\sum_{r_M=1}^{R_M} I_{r_M}(A_M|A_{M-1}) \\ &= \frac{1}{R_1\ldots R_M}\sum_{r_1=1}^{R_1}\ldots\sum_{r_M=1}^{R_M} I_{r_1}(A_1)\ldots I_{r_M}(A_M|A_{M-1})\end{aligned} \tag{2.24}$$

with variance

$$\mathrm{Var}(\hat{\gamma}_R) = (p_1\ldots p_M)^2\sum_{j=1}^{M}\frac{1-p_j}{(p_1R_1)\ldots(p_MR_M)}. \tag{2.25}$$

To optimise the RESTART speedup, the variance $\mathrm{Var}(\hat{\gamma}_R)$ is minimised subject to [VA+94, GHSZ96, Kel96, SG94]:

- R_i - the number of replicas at each threshold => recommended $R_i = 1/p_i$.
- M - the number of thresholds => chosen to give $p_i = p = e^{-2}$ for all $i = 1, \ldots, M$.
- Ω_{1i} - the threshold definition => chosen in accordance to the criteria for M.

RESTART is a flexible and robust approach applicable to transient and steady state simulations [VAVA94]. However, the variance reduction (gain) drops dramatically if R_i is far from optimal [Kel96]. Defining thresholds, Ω_{1i}, becomes difficult when the dimensionality and size of the simulated model is large, and there is no symmetry that can be exploited.

Defining the optimal set of thresholds is generally a great challenge when the simulation model increases in dimensionality. A multidimensional state space, results in multidimensional thresholds, and a huge number of replicas must be generated to get a representative sample at each level. In [GHSZ97], it is pointed out that, except for specific models, it is impossible to identify an optimal splitting of the simulation process in a multidimensional model.

2.7.2 Importance sampling

Importance sampling changes the dynamics of the simulation model. The dynamics must be changed to increase the number of rare events of interests. Let $f(\underline{s})$ be the original sampling distribution where $\underline{s}$ is, for instance, a *sample path* as will be defined in section 4.3.3. This sampling distribution is changed to a new distribution $f^*(\underline{s})$. If $g(\underline{s})$ is the property of interest, the original problem is that the probability of observing $g(\underline{s}) > 0$ is very small when taking samples from $f(\underline{s})$. This probability should be significantly increased with samples from $f^*(\underline{s})$. The observations made must be corrected because the samples $\underline{s}$ are from $f^*(\underline{s})$ and not $f(\underline{s})$.

The property of interest $\gamma = E(g(\underline{s}))$ can now be rewritten:

$$\gamma = E_f(g(\underline{s})) = \sum_{\forall \underline{s}} g(\underline{s}) f(\underline{s}) = \sum_{\forall \underline{s}} g(\underline{s}) \frac{f(\underline{s})}{f^*(\underline{s})} f^*(\underline{s}) = E_{f^*}(g(\underline{s}) \cdot L(\underline{s})) \tag{2.26}$$

where $L(\underline{s}) = f(\underline{s}) / f^*(\underline{s})$ is denoted the *likelihood ratio*. Observe that the expected value of the observations under f is equal to the expected value of the observations under f^* corrected for bias by the likelihood ratio, $E_f(g(\underline{s})) = E_{f^*}(g(\underline{s}) \cdot L(\underline{s}))$.

An unbiased estimator for γ, taking samples $\underline{s}$ from $f^*(\underline{s})$, is:

$$\hat{\gamma}_{\mathrm{IS}} = \frac{1}{R} \sum_{r=1}^{R} g(\underline{s}_r) L(\underline{s}_r) \tag{2.27}$$

with variance

$$\mathrm{Var}(\hat{\gamma}_{\mathrm{IS}}) = \frac{1}{R} \mathrm{Var}_{f^*}(g(\underline{s}) L(\underline{s})) = \frac{1}{R} E_{f^*}((g(\underline{s}) L(\underline{s}) - \gamma)^2). \tag{2.28}$$

The optimal change of measure is given by the $f^*(\underline{s})$ that minimises this $\mathrm{Var}(\hat{\gamma}_{\mathrm{IS}})$, i.e.

$$g(\underline{s})L(\underline{s}) - \gamma = 0\,. \tag{2.29}$$

However, this depends on knowledge of the unknown property of interest, γ. Nevertheless, (2.26) and (2.29) give general guidelines:

- if $g(\underline{s})\,f(\underline{s}) > 0$ then $f^*(\underline{s}) > 0$ (from (2.26)),
- $g(\underline{s})\,f(\underline{s}) = \gamma\, f^*(\underline{s})$ which implies that the sampling should be in proportion to the original importance and likelihood of the samples, i.e $g(\underline{s})\,f(\underline{s}) \propto f^*(\underline{s})$ (from (2.29)).

Because the efficiency of importance sampling is observed to be rather sensitive to the change of measure, a lot of work is done to obtain optimal, or at least good, simulation parameters. In chapter 3, a further treatment of importance sampling and its change of measure will be given, see also [Hei95] which includes an extensive and excellent survey on selecting $f^*(\underline{s})$.

2.7.3 Combination of importance sampling and RESTART

Importance sampling and RESTART can be combined by changing the sampling distribution of the RESTART estimate, see [CHS95] for a similar idea.

An unbiased estimate of $\gamma = P(A)$ for the combination is:

$$\hat{\gamma}_{\text{IS+R}} = \frac{1}{R_1 \ldots R_M} \sum_{r_1=1}^{R_1} \ldots \sum_{r_M=1}^{R_M} I_{r_1}(A_1) L_{r_1}(\underline{s}_{r_1}) \ldots I_{r_M}(A_M | A_{M-1}) L_{r_M}(\underline{s}_{r_M} | A_{M-1}) \tag{2.30}$$

with variance [CHS95]:

$$\begin{aligned} \text{Var}(\hat{\gamma}_{\text{IS+R}}) \equiv \text{Var}(\hat{I}_M) &= \text{Var}(\hat{I}_{M-1} \cdot \hat{p}_M) \\ &= (p_1 p_2 \ldots p_{M-1})^2 \text{Var}(\hat{p}_M) + \text{Var}(\hat{I}_{M-1}) \cdot E(\hat{p}_M^2) \end{aligned} \tag{2.31}$$

where $\hat{I}_i$ and $\hat{p}_i$ are defined as:

$$\hat{I}_i = \frac{1}{R_1 \ldots R_i} \sum_{r_1=1}^{R_1} \ldots \sum_{r_i=1}^{R_i} I_{r_1}(A_1) L_{r_1}(\underline{s}_{r_1}) \ldots I_{r_i}(A_i | A_{i-1}) L_{r_i}(\underline{s}_{r_i} | A_{i-1})\,, \tag{2.32}$$

$$\hat{p}_i = \frac{1}{R_i} \sum_{r_i=1}^{R_i} I_{r_i}(A_i | A_{i-1}) L_{r_i}(\underline{s}_{r_i} | A_{i-1})\,. \tag{2.33}$$

As the results in section 2.8.2.2 indicate, the optimal parameters of importance sampling and RESTART should probably not be obtained separately when the two techniques are combined. Instead,

- the parameter biasing of importance sampling in combination with RESTART should be less than the optimal change of measure in a separate importance sampling experiment, and

- the number of thresholds of RESTART, and the number of replications given by this, should be less than the optimal number obtained without considering importance sampling.

2.7.4 Different impact on sampling distribution

As described in the previous section, both RESTART and importance sampling manipulate the underlying sampling density distribution of $\underset{\sim}{s}$. To demonstrate how the two approaches affect the underlying distribution, a probability measure is defined to describe their ability to provoke rare events. Such a measure is not easy to obtain in the general case. Instead, consider regenerative simulation of a specific M/M/1/N example where the probability of blocking is the property of interest. A rare event is observed in all simulated cycles that includes a visit to state N. This implies that if the maximum state visited during a cycle i is less than N, no rare events are observed. The probability of state i being the maximum state visited during a cycle, or sample path, $\underset{\sim}{s}_r$ with n_r events is denoted P_i $(i = 1, \ldots, N)$:

$$P_i = Pr\left\{\max_{\forall r}(\underset{\sim}{s}_r) = i\right\} = Pr\{(\underset{\sim}{\omega}_x = i) \wedge (\underset{\sim}{\omega}_x \leq i), (x = 1, \ldots, n_r)\}. \tag{2.34}$$

This probability is applied to describe the difference in the two approaches RESTART and importance sampling. The P_i is provides a good indication of how well the two techniques succeed in “pushing” the simulation process towards the targets. In appendix C, the explicit expressions of P_i, valid for an M/M/1/N model, are given for direct, importance sampling, RESTART and combined simulations.

The probabilities P_i are plotted for $i = 1, \ldots, N$ for direct, importance sampling, and RESTART in figure 2.7. The following observations are made:

- *RESTART* increases the rare event probability by increasing the probability of the target state $P_N^{(\mathrm{R})}$ significantly ($P_i^{(\mathrm{R})}$ is the probability measure P_i under the RESTART sampling distribution). However, observe that the $P_i^{(\mathrm{R})}$ is also increased significantly relative to the original distribution for all states $i > \Omega_{11}$, i.e. after the first threshold. As the number of thresholds M increases, the probability measure tends to converge towards a straight line. When $M = N$, each state in the chain is a threshold, and then $P_i^{(\mathrm{R})} \to 1/M$ is a uniform distribution.

- *Importance sampling* with optimal parameters is close to the original distribution for all states $i < N$, while $P_N^{(\mathrm{IS})}$ ($P_i^{(\mathrm{IS})}$ is the probability measure P_i under the importance sampling distribution) is increased significantly, and more than for RESTART, $P_N^{(\mathrm{IS})} \geq P_N^{(\mathrm{R})}$. If the sampled sequence $\underset{\sim}{s}$ under optimal importance sampling does not hit state N, then $P_i^{(\mathrm{IS})} < P_i$. From figure 2.7 it is observed that the ratio $P_i/P_i^{(\mathrm{IS})}$ seems to be constant. This is confirmed analytically in appendix C where the ratio is determined to be $P_i/P_i^{(\mathrm{IS})} = \mu/\lambda$, for all $i = 1, \ldots, N-1$. This is the same as the BIAS -factor under optimal biasing, see section 3.3.3.1. For non-optimal importance sampling the ratio is not constant, and the $P_N^{(\mathrm{IS})}$ is relatively less at state N. For all intermediate states, $1 \leq i < N$, a significant increase to $P_i^{(\mathrm{IS})}$ is observed. This is similar to what was observed for the RESTART.

Even though the plots of $P_{\max}(i)$ in figure 2.7 are generated based on a very simple model, it is expected that the principal differences that are observed are of a general nature.

2.8 Experiments

In [Hee95a, Hee95c, Hee97b], a few comparisons between importance sampling and RESTART on a simple M/M/1/N are reported. In these results, importance sampling with an optimal change of measure will always be at least as good, normally far better, than RESTART with optimal splitting. However, if a non-optimal change of measure is used, then RESTART is sometimes better.

In this section, comparisons between importance sampling, RESTART, and the combination of the two, are made by simulations of an M/M/1/N queue. Furthermore, simulations are also carried out on a model of a shared buffer offered traffic from K user types. When all traffic types are (nearly) equal, the model is said to be *symmetric* or *balanced*. This has previously been pointed out to be challenging for importance sampling with respect to obtaining optimal change of measure [PW89, Fra93].

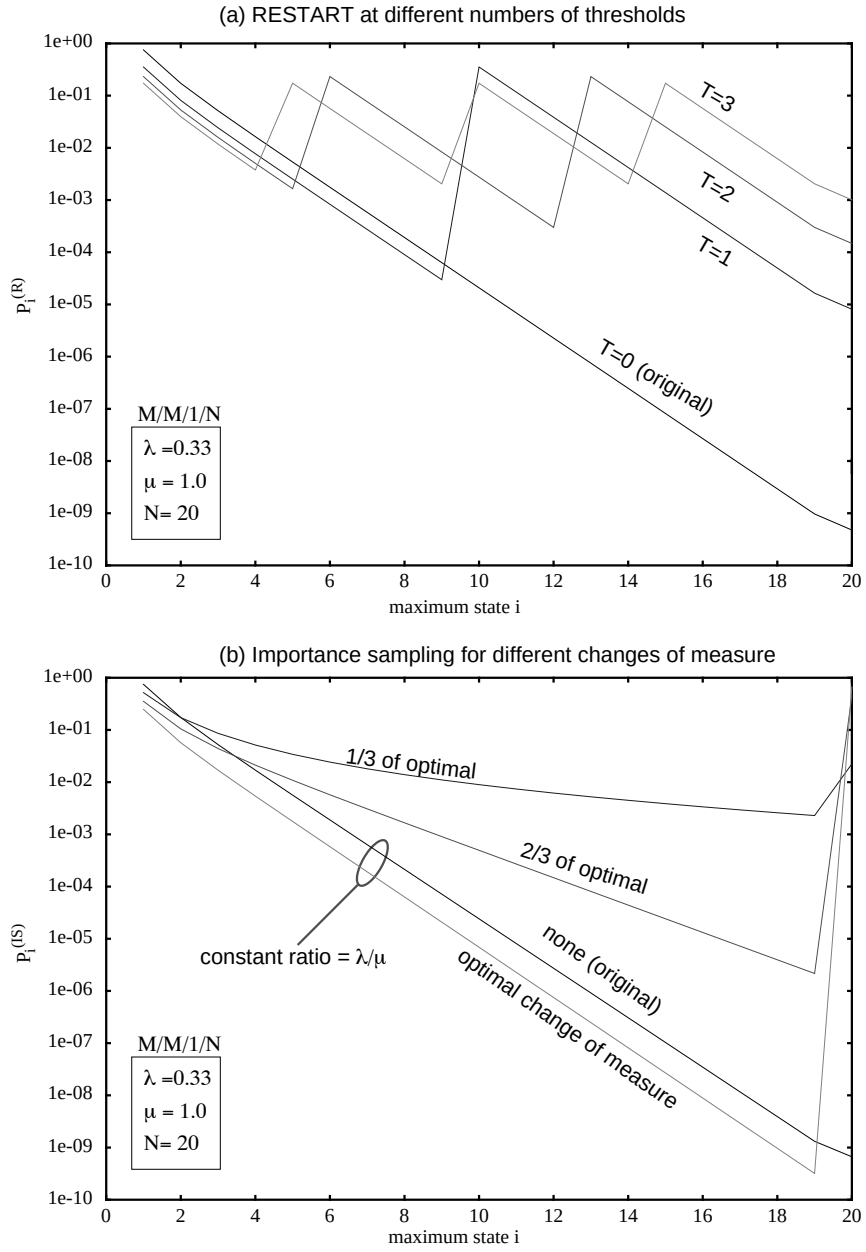


Figure 2.7 : Plots of P_i, the probability of state i being the maximum state visited in a cycle in a simulation of an M/M/1/N queue with λ =0.33, μ =1, N=20.

2.8.1 Simulation setup

In all experiments, the property of interest is the steady state blocking probability obtained by the estimate in (2.10). Each experiment is a *regenerative simulation* of R cycles.

The mean values and their variance are estimated from the results obtained from 20 independent experiments applying the *replication technique* [LO88, BFS87].

2.8.2 Single server queue, M/M/1/N

For a single server queue with Poisson arrivals (λ) and exponential service times (μ^{-1}), it straightforward to obtain exact results. Furthermore, RESTART has well defined optimal thresholds and number of replications, and importance sampling has a known asymptotically optimal change of measure. A state space model of this system is given in figure 2.8.

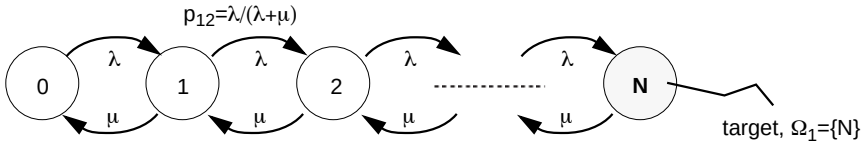


Figure 2.8 : Model of a single server queue, M/M/1/N.

2.8.2.1 Optimal importance sampling vs. optimal RESTART

Optimal importance sampling implies that the change of measure minimises the variance of $\hat{\gamma}_{\mathrm{IS}}$ in (2.28). When simulating steady state behaviour in Markov models, the change of measure need only to change the transition probabilities,

Arrival probability at state i: $\quad p^*_{i,\,i+1} = \lambda^* / (\lambda^* + \mu^*)$ and

Departure probability at state i: $\quad p^*_{i,\,i-1} = 1 - p^*_{i,\,i+1}$.

The optimal choice of parameters for importance sampling simulation of an M/M/1/N queue is to interchange the transition probabilities, see [CFM83, PW89], $p^*_{i,\,i+1} = p_{i,\,i-1}$ and $p^*_{i,\,i-1} = p_{i,\,i+1}$ This, and other changes of measure, will be discussed in further details in the following chapter.

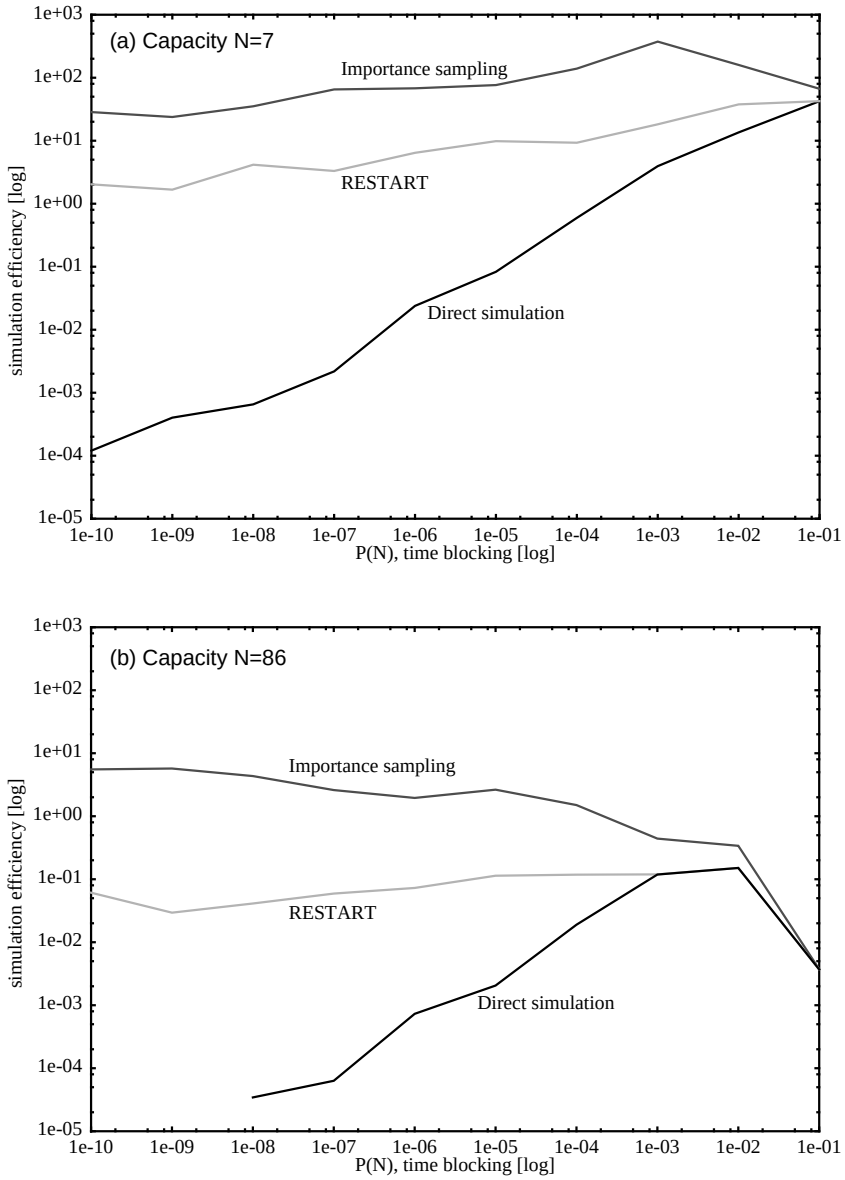


Figure 2.9 : The simulation efficiency for direct and rare event provoking simulations.

Optimal RESTART requires the relative probability between two thresholds to be $p_i \approx e^{-2}$ for all thresholds. The number of replicas should be as close to $R_i = 1/p_i$ as possible [VA+94]. In a state space model with a few states, it is difficult to set up the RESTART experiment because the optimal threshold is likely to be between two states. See the discussion in e.g. [Kel96] where $R_i = (p_{i+1} \cdot p_i)^{-1/2}$ is proposed as an alternative.

In addition to a small example with $N = 7$, a larger one with $N = 86$ is studied. The larger example is included for two reasons:

- the asymptotic optimal change of measure for importance sampling is expected to be inaccurate for small N,
- the optimal number of thresholds of RESTART is not an integer and hence it is difficult to fit the thresholds in a small model.

The results in figure 2.9(a) and (b) show the simulation efficiency of direct simulation, importance sampling, and RESTART for different blocking probabilities:

- *Direct simulation*: as expected from (2.9), the efficiency is proportional to $P(N)$.
- *RESTART*: the efficiency is proportional to $\log P(N)$ [1].
- *Importance sampling*: the efficiency is proportional to $\log P(N)$ [1], but with less slope than for RESTART.
- For the large system with $N = 86$, all techniques are inefficient at *high load* and hence for high blocking probability. This is because the regenerative simulation experiments use the “empty system” state as the renewal state. Hence, a renewal is a rare event. In the network simulation example in chapter 6, an alternative renewal state is applied to eliminate this problem.

The results show that importance sampling with optimal biasing is always at least as efficient as RESTART, normally far better. Even though the results are from experiments on a simple M/M/1/N queue only, this is expected also to be true for more general models. To quote Dr. Phil Heidelberger’s comment on the statement “whenever an optimal change

1. The efficiency of both importance sampling and RESTART are independent of the rarity only if the number of events in each sample is not included in the efficiency measure.

of measure exists for importance sampling, it is always as least as good as RESTART, normally far better", he wrote that [Nic97]: "This pretty much *HAS* to be true since you can't do better than optimal in *ANY* situation (or much better than asymptotically optimal in rare event simulations).". However, to qualify this statement, a few experiments of other, more general models with optimal biasing, are left as further work.

2.8.2.2 Importance sampling, RESTART and the combination

A series of experiments are carried out where importance sampling and RESTART are compared and combined. The experiments are using the same M/M/1/N examples as in the previous sections. The parameter biasing of importance sampling is varied for the purpose of demonstrating that RESTART, in separate simulations or in combination with importance sampling, is more efficient than importance sampling when the biasing is far from optimal.

Let the scaling of the arrival probability vary from no change, up to the optimal change, $p^*_{i,\,i+1} \in [p_{i,\,i+1},\, p_{i,\,i-1}]$. A series of experiments are carried out on an M/M/1/N queue with $N = 10$, $\lambda = 0.1$, and $\mu = 1$. RESTART and importance sampling are combined, and the number of thresholds and the $p^*_{i,\,i+1}$ are varied. The scaled arrival probability is varied from original parameters to optimal parameters, $p^*_{i,\,i+1} \in [0.09,\, 0.91]$.

The second experiment applies the same M/M/1/N queue, now with $N = 20$, $\lambda = 0.33$, and $\mu = 1$. For this example, $p^*_{i,\,i+1} \in [0.25,\, 0.75]$. The number of thresholds is varied from 1 up to 10. The optimal number of thresholds, if only RESTART was applied, is 10.

In figures 2.10 and 1.5, a selection of the experiments, which produced the best results in 3 different regions, are plotted.

- *Region 1*: RESTART is the most efficient,
- *Region 2*: RESTART combined with importance sampling is the most efficient,
- *Region 3*: Importance sampling is the most efficient.

Recall from section 2.7.4 where the differences between RESTART and importance sampling were discussed with respect to their impact on the sampling distribution. Consider the two samples in figure 2.11 marked with a circle. At these points, the simulation efficiency is almost the same for RESTART and importance sampling. The $P_i^{(\mathrm{R})}$ and $P_i^{(\mathrm{IS})}$ are plotted for all $i = 1, \ldots, N$. Note the strong resemblance.

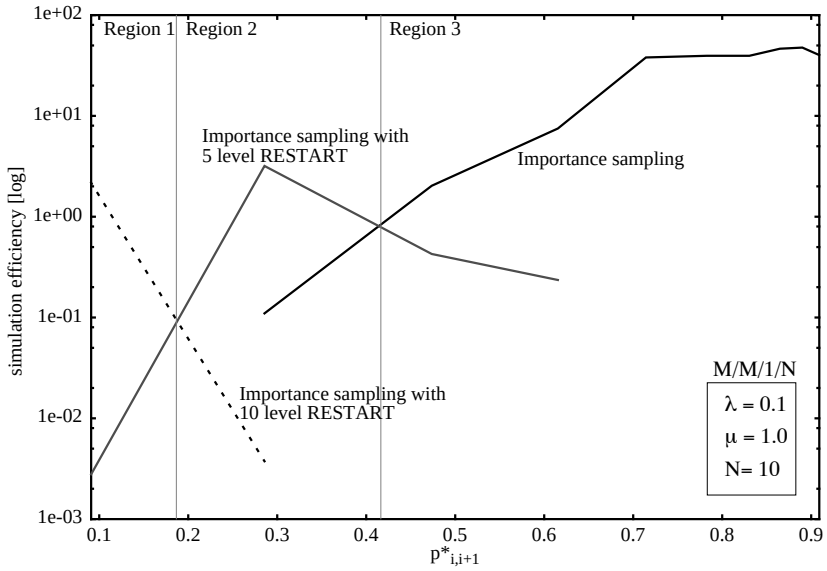


Figure 2.10 : Importance sampling has best performance even if the change of measure is less than optimal.

2.8.3 K traffic types with shared buffer

Experiments are conducted on a shared buffer offered traffic from K user types. The kth user has arrival rate λ_k and a dedicated server with service rate μ_k. In figure 2.12, the mapping from this system to a model with multiple dimensions, is described by the general state, $\underset{\sim}{\omega}$, in the multidimensional state space, $\underset{\sim}{\Omega}$. All transitions out of the state $\underset{\sim}{\omega} = \{\omega_1, \omega_2, \ldots, \omega_K\}$ are described. $\underset{\sim}{1}_k = \{0, 0, \ldots, 1, \ldots\}$ is an *index vector* of size K where element k is 1, and 0 elsewhere. The state space is truncated where when the system capacity is reached, i.e. when all N positions in the shared buffer are occupied. The quantity of interest is the probability of blocking, i.e. $P(x \in \underset{\sim}{\Omega}_1)$.

When $\lambda_k = \lambda$ and $\mu_k = \mu$ for all k, the resulting state space is said to be *symmetric* or *balanced*. Observe that this is *not the same* as a superposition of traffic generators (with rate $k\lambda$) to an M/M/1/N queue. The reason is that the model assumes *dedicated servers* to each generator. This means that the server rate of the queue is dependent in the current combination of users in $\underset{\sim}{\omega}$. If the parameter biasing in importance sampling is based on ignorance of the service rate dependence on the system state, a too strong biasing will occur. This is known from [PW89, Fra93], and will be treated in more detail in chapter 3.

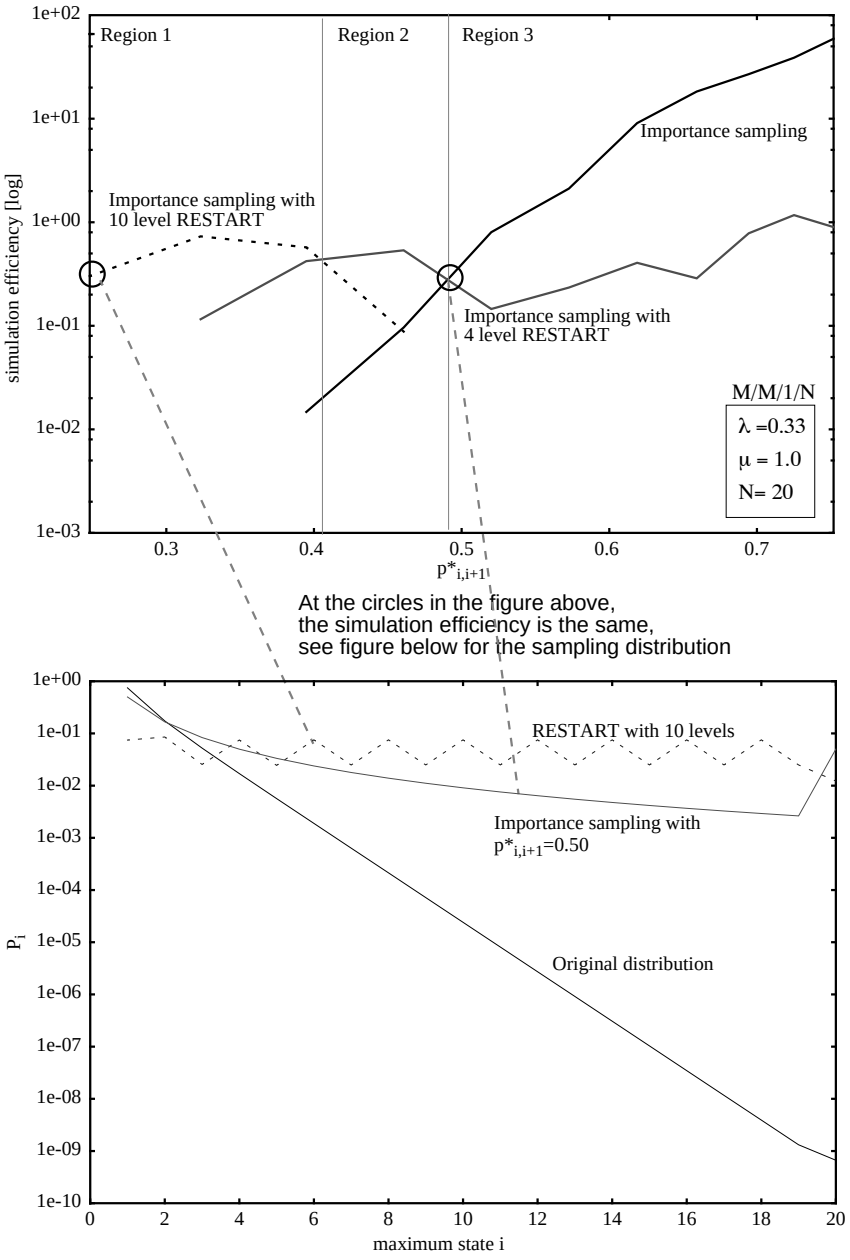


Figure 2.11 : Importance sampling has best performance even if the change of measure is less than optimal. In the lower figure, the probabilities P_i are plotted for every $i = 1, \ldots, N$ for RESTART and a non-optimal importance sampling where the simulation efficiency is almost identical.

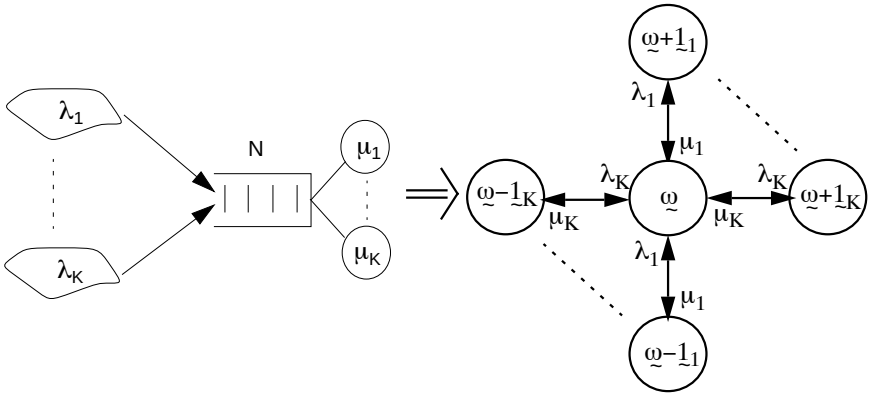


Figure 2.12 : Shared buffer model with the general multidimensional state.

In figure 2.13, the simulation efficiency vs. the number of traffic types K, is plotted for importance sampling and RESTART. The simulation parameters are chosen to have a blocking probability less than 10^{-9} for all values of K.

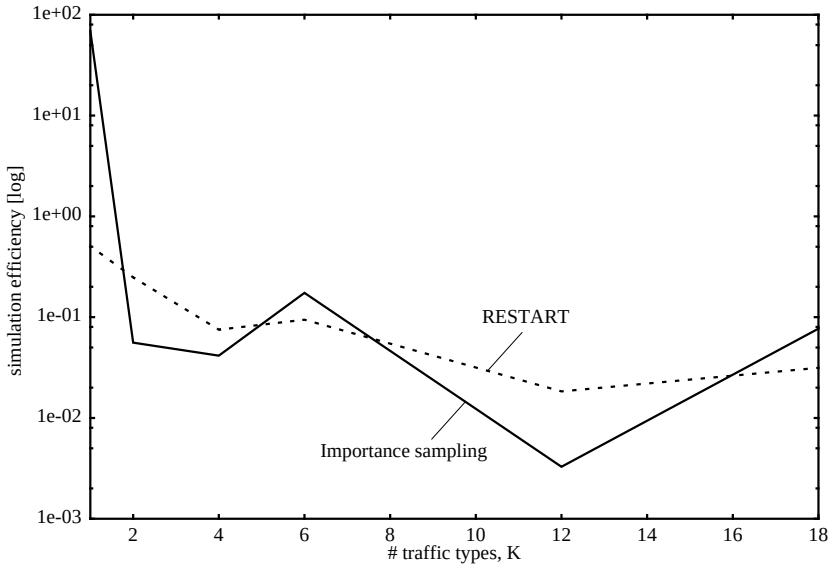


Figure 2.13 : Comparison of importance sampling and RESTART for number of dimensions K.

The results shows that in symmetric or balanced multidimensional state spaces, importance sampling is only more efficient than RESTART in the single dimensional case. RESTART is able to exploit the symmetry in the state space for easy definition of good thresholds in this model, but this is not generally the case. Hence, it is expected that RESTART will be less efficient than importance sampling in a non-symmetric model where the dimensions are different and with several boundaries in the state space.

2.9 Closing comments

This chapter contains a brief overview of 4 different categories of speedup techniques for discrete event simulation. They are not mutually exclusive, and hence should be combined whenever possible. An example of a successful combination can be found in [HH94], where an accelerated measurement and simulation technique for ATM traffic is described. This technique combines, control variables, importance sampling, and parallel independent replicated simulations. Later, a hybrid technique is included to reduce the simulation overhead by exploitation of a known regularity in the traffic pattern.

Parallel simulation can be applied for any simulation problem involving a certain amount of computations, including rare event simulations. For simulation experiments where a large number of replicas are needed, it is recommended to run the sequential simulation processes in parallel on several processors. In contrast to this, the sequential simulation process can identify and distribute parallel sub-processes. However, this requires expert skills in parallel programming, access to special hardware and software, and will result in less speedup than to run sequential processes in parallel.

Both hybrid and variance reduction techniques are model dependent and require insight in the problem at hand. If not, it is not possible to identify submodels with analytic solutions, or to identify correlation between samples that can be exploited for variance reduction. This is a general challenge, and not related to rare event simulation.

Rare event provoking techniques are very efficient because they directly manipulate the dynamics of the simulation process with respect to the rare events of interest. However, they require good insight in the statistical properties of the model, particularly to obtain the simulation parameters involved in making the techniques as efficient as possible.

The emphasis in this section has been on rare event provoking techniques. The differences between RESTART and importance sampling have been pointed out, and a comparison

and combination of these two techniques are made. The experiments were conducted on a simple M/M/1/N system. Importance sampling with optimal change of measure is always more efficient than RESTART. Adding RESTART thresholds to importance sampling will not improve the speedup unless the change of measure is far from the optimal. Even though the results are from experiments on a simple M/M/1/N queue only, this is also expected to be true for more general models. However, to qualify this statement, numerous experiments on more general models, with optimal biasing, should be conducted. This is left as further work.

RESTART and importance sampling are basically different. Their impact on the underlying sampling distribution of an M/M/1/N model is described in this chapter. The study demonstrates that for some parameter values, the RESTART and importance sampling have very similar impact on the sampling distribution. Hence, for this combination, the simulation efficiency was almost equal. These observations are expected to be valid for a broader class of models than M/M/1/N.

Previously, it has been pointed out that importance sampling, using the change of measure obtained by results of large deviation theory, is not efficient in *symmetric* models with multiple dimensions. The experiments in this chapter confirm this, and also demonstrate that the use of RESTART instead is at least as efficient because the symmetry can be exploited. However, for general models with multiple dimensions and no symmetry, it is not easy to set up an efficient RESTART simulation. RESTART will suffer from problems with defining thresholds due to state explosion. In [GHSZ97], it is pointed out that, except for specific models, it is impossible to identify an optimal splitting of the simulation process in a multidimensional model. The reason can be explained with reference to figure 2.14 which is adopted from [GHSZ97]. The rare events of interest are observed when the process reaches threshold A. The most likely path from the origin to this threshold passes through an intermediate threshold B in b'. Correspondingly, the most likely path from origin to this intermediate threshold ends in state b. b and b' are only the same state for symmetric models like the one in section 2.8.3. Hence, care must be taken when establishing thresholds in multidimensional models. The remaining question is how robust the RESTART is, that means how efficient is the simulations, when a sub-optimal path is followed?

RESTART will also experience a significant decrease in the efficiency because a huge number of visits to each threshold is required to obtain a representative sample. In contrast, importance sampling is, to a certain extent, in these models able to exploit the

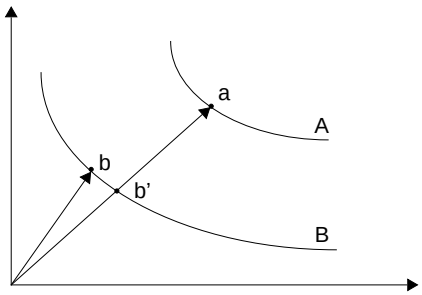


Figure 2.14 : The most likely path to the intermediate set B hits B at b, but the most likely path to the final set A hits B at b'. Figure adapted from [GHSZ97].

identified differences. Efficient simulation can then be established, as will be described in chapter 5.

3

Change of measure in importance sampling

In the previous chapter, an overview of a number of speedup simulation techniques was given. This chapter contains details of *importance sampling*, which is suitable for efficient simulation of probabilities of rare, but important, events. A brief overview is given of previous approaches to make importance sampling efficient and robust. In addition, an easy to implement way of changing the simulation parameters is presented. It is also described how this parameter biasing is done in a multidimensional model with a single resource constraint. For several constraints, the new, adaptive biasing is required. This approach will be described in chapter 5. At the end of this chapter, a summary is given of the most important observations made from quite a few simulation experiments on both simple and multidimensional models. The heuristics based on these observations are also discussed.

3.1 The change of measure

Importance sampling was briefly introduced in chapter 2.7.2, where the principles and equations were given. The basic equation for the expected value of the property $g(\underline{s})$ is repeated for convenience (from (2.26)):

$$\gamma = E_f(g(\underline{s})) = \sum_{\forall \underline{s}} g(\underline{s}) f(\underline{s}) = \sum_{\forall \underline{s}} g(\underline{s}) \frac{f(\underline{s})}{f^*(\underline{s})} f^*(\underline{s}) = E_{f^*}(g(\underline{s}) \cdot L(\underline{s})) . \qquad (3.1)$$

This chapter deals with the *change of measure* from the original sampling distribution $f(\underline{s})$ to a new distribution $f^*(\underline{s})$. As pointed out in chapter 2.7.2, this is the main challenge with respect to making importance sampling efficient and robust. The only restriction to the probability density $f^*(\underline{s})$ observed from equation (2.26) is that $f^*(\underline{s}) > 0$ for all samples $\underline{s}$ where $g(\underline{s}) f(\underline{s}) \neq 0$. This means that the samples $\underline{s}$ with a positive probability $f(\underline{s}) > 0$ and a non-zero contribution $g(\underline{s})$, must have a positive

probability also in the new distribution, $f^*(\underline{s})$. This is a necessary condition which serves as a guideline for choosing $f^*(\underline{s})$. The efficiency of the following importance sampling estimate (from (2.27))

$$\hat{\gamma}_{\mathrm{IS}} = \frac{1}{R}\sum_{r=1}^{R} g(\underline{s}_r)\frac{f(\underline{s}_r)}{f^*(\underline{s}_r)} = \frac{1}{R}\sum_{r=1}^{R} g(\underline{s}_r)L(\underline{s}_r) \tag{3.2}$$

is dependent on an appropriate choice of the new distribution. An unfortunate choice of $f^*(\underline{s})$ may cause the variance of $\hat{\gamma}_{\mathrm{IS}}$ to be larger than the variance of the estimates obtained by direct simulation. For some $f^*(\underline{s})$, the variance may be infinite, as will be discussed in section 3.5. Hence, it is crucial to find a good $f^*(\underline{s})$. The *optimal* $f^*(\underline{s})$ is the one that minimises the variance of $\hat{\gamma}_{\mathrm{IS}}$ (from (2.28))

$$\mathrm{Var}(\hat{\gamma}_{\mathrm{IS}}) = \frac{1}{R}\mathrm{Var}_{f^*}(g(\underline{s})L(\underline{s})) = \frac{1}{R}E_{f^*}((g(\underline{s})L(\underline{s}) - \gamma)^2). \tag{3.3}$$

The variance is minimised to $\mathrm{Var}(\hat{\gamma}_{\mathrm{IS}}) = 0$ when $g(\underline{s})f(\underline{s}) = \gamma f^*(\underline{s})$. But, this requires knowledge of γ, the property of interest. Anyhow, this observation may serve as a guideline stating that $g(\underline{s})f(\underline{s}) \propto f^*(\underline{s})$. This makes efficient application of importance sampling very model specific.

Example 3.1: To demonstrate the effect of different changes of measure, a simple stochastic model with 3 different alternatives of $f^*(\underline{s})$ is introduced as an example. The samples s are taken from a one dimensional Poisson process with rate $\lambda = 1/\nu$, i.e. $f(s) = e^{-\nu}\nu^s/s!$. Observe that in this example the sample is a scalar s and not a vector $\underline{s}$. The property of interest is the probability of observing samples s greater than a certain threshold α, i.e. $\gamma = P(s > \alpha)$. If $\nu = 10$ and the threshold is $\alpha = 30$, then the exact value is $\gamma = 7.98\times10^{-8}$. Hence, to obtain estimates of γ with confidence $1 - \varepsilon = 0.95$ by direct simulation, approximately $R \approx 1/\varepsilon \cdot 1/\gamma = 2.5\times10^{8}$ samples must be taken from $f(s)$, according to (2.9) in section 2.1.

Instead, importance sampling should be applied to increase the efficiency. But, what should the change of measure be? Generally, any probability density function applies that is defined in the non-zero region of γ, i.e. that fulfils the requirements of $f^*(s) > 0$ if $g(s)f(s) > 0$. Three different changes of measure are proposed in this example.

i. *Poisson distribution,* $f_p^*(s)$, with rate $\nu^* = 1/\alpha$. This is an efficient change of parameter for a truncated Poisson distribution, see [Kel86, Man96c].

ii. *Binomial distribution,* $f_b^*(s)$, with $n = 60$ and $p = 0.5$. The binomial distribution approaches Poisson for large n, the parameters are chosen to give $n \cdot p = \alpha$, i.e. to have the same expected mean as $f_p^*(s)$.

iii. *Discrete uniform distribution,* $f_u^*(s)$, with $n = 60$, i.e. defined on the same domain as the Binomial distribution, $s = 0, \ldots, n$.

The guideline from variance minimisation of $\hat{\gamma}_{IS}$ stated that $f^*(s)$ should be proportional to $g(s)f(s)$. In this example $g(s) = 1$ for all samples $s > \alpha$, which means that $f^*(s)$ should be proportional to $f(s)$ for $s > \alpha$. Figure 3.1 shows a plot of the original density and the densities of the three new distributions. Figure 3.1(a) shows the densities over the complete range [0,60], while figure 3.1(b) plots the re-normalised logarithmic densities over $[30, 60]$, the subrange above the threshold α. The latter is included to compare the shapes of the densities in the region where the rare events occur. The Poisson and binomial distributions have their modes centred at the threshold α. The original, Poisson, and binomial distributions are monotonically decreasing in $s > \alpha$. In contrast, all samples $s > \alpha$ in the discrete uniform distribution are equally likely, i.e. $f_u^*(s)$ is not proportional to $g(s)f(s)$ where $g(s) > 0$.

The estimates $\hat{\gamma}_{IS}$ and their *standard error*[1], $S_{\hat{\gamma}_{IS}}$, are included in table 3.1. The experiments include 1000 samples from each of the 3 distributions. As expected, the uniform distribution, which is not proportional to $g(s)f(s)$ o for $s > \alpha$, has the poorest standard error. Table 3.1 also includes the time, t, taken to generate 1000 samples of each distribution, and the efficiency measure, m, from (2.11). Taking the efficiency of the sampling algorithm into account, the uniform distribution is close to being the best alternative (which is the Binomial distribution) for this example. Sampling from a Poisson distribution is time consuming in the *Mathematica* implementation [Wol91].

Table 3.1: Different change of measure.

Distribution, $f^*(x)$	$\hat{\gamma}_{IS}$ [10^{-8}]	$S_{\hat{\gamma}_{IS}}$ [10^{-8}]	t[a]	m [b] [10^{16}]	exact, γ [10^{-8}]
Poisson, λ=0.1	8.59	0.69	95.8	0.0219	7.98
Binomial, n=60, p=0.5	7.47	0.47	26.1	0.1724	
Discrete uniform, n=60	8.07	1.28	3.9	0.1563	

a. Time taken to generate 1000 samples by Mathematica.
b. The efficiency measure from (2.11), $(S^2_{\hat{\gamma}_{IS}} \cdot t)^{-1}$.

1. See appendix B.3.

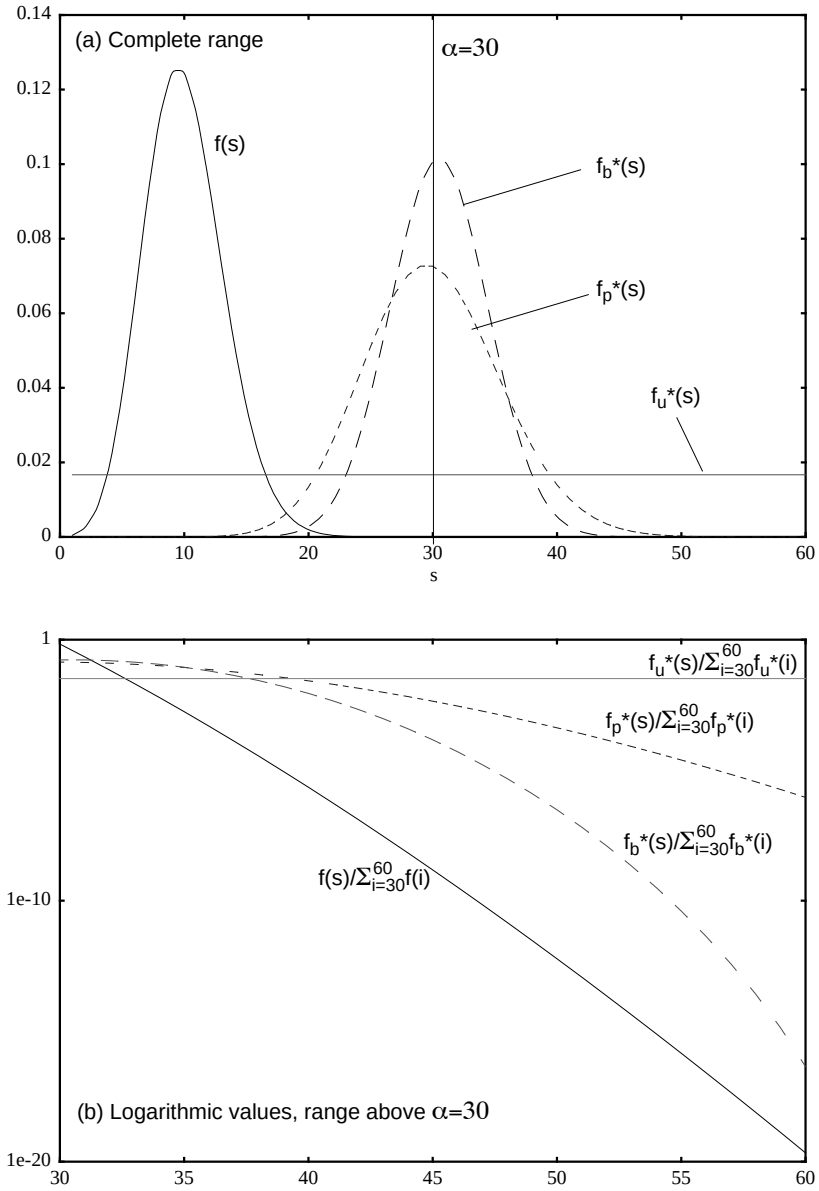


Figure 3.1 : The density functions for the original Poisson, $f(s)$, Poisson with biased rate, $f_p^*(s)$, Binomial, $f_b^*(s)$, and discrete uniform, $f_u^*(s)$, distributions. Figure (a) shows the range [0,60], while (b) shows the logarithmic densities for the range [30,60].

The example shows two things. First that importance sampling can apply a change of measure $f^*(s)$ from various distributions, and is not limited to distributions within the same family of distributions as the original (here Poisson). Furthermore, the results in table 3.1 demonstrate that not only the variance reduction, but also the efficiency of the sample generating algorithm, should be considered when choosing "the best $f^*(s)$" for the problem at hand.

3.2 The simulation process

In this chapter, importance sampling is applied on a simulation process that is assumed to be a *continuous time Markov chain* (CTMC). This section presents the details and restrictions that are necessary to understand the parameter biasing introduced in section 3.4. A more general view of the simulation process will be given in chapter 4, where the building blocks of the framework for network models are presented.

Let $\underset{\sim}{X}(t)$ be a K-dimensional *continuous time Markov chain* (CTMC), defined on a sample space[1] $\underset{\sim}{\Omega}^K = \{0, 1, \ldots, N\}^K$. The state changes in $\underset{\sim}{X}(t)$ are denoted *events*[2] and take place at embedded points in time. A single transition affects only one dimension at a time[3]. If $t_0, t_1, \ldots, t_n$ are n embedded points, then $\underset{\sim}{X}(t)$ can be made discrete in time by

$$\underset{\sim}{X}_i = \underset{\sim}{X}(t_i) = \{\omega_{1;i}, \omega_{2;i}, \ldots, \omega_{K;i}\}, \text{ where } i = 1, \ldots, n. \tag{3.4}$$

This is a *discrete time Markov chain* (DTMC). After event i it is equal to $\underset{\sim}{X}(t_i)$ at embedded point t_i. The $\underset{\sim}{X}_i$ can be expressed by recursion,

$$\underset{\sim}{X}_i = \underset{\sim}{X}_{i-1} + \underset{\sim}{Z}_i. \tag{3.5}$$

$\underset{\sim}{Z}_i$ is the random *event variable* that can take on any (feasible) integer value in one of the K dimensions. An event is e.g. a call arrival, a service completion, a component failure. In section 3.4.2, a possible implementation of importance sampling change of measure is described. This chapter limits the event variable to be defined on {-1,1}, in either of the K dimensions[4],

1. Adding resource limitations to the model, e.g. finite buffer capacity, the feasible region of $\underset{\sim}{\Omega}^K$ will be reduced, e.g. a common resource limitations will cut the corners of this state cube.
2. See chapter 4 for a refinement of the event definition.
3. When preemptive priorities are introduced in chapter 4, this is no longer true.
4. The index vector of size K is $\underset{\sim}{1}_k = \{0, \ldots, 1, \ldots, 0\}$ where element k is 1, and 0 elsewhere.

$$\underset{\sim}{Z}_i = \begin{cases} -\underset{\sim}{1}_k & \text{with probability } q_{k;i},\ \forall k \\ \underset{\sim}{1}_k & \text{with probability } p_{k;i},\ \forall k \end{cases}. \tag{3.6}$$

In chapter 4, a more general view of the simulation process is given.

In the following sections, an overview of the change of measure on $\underset{\sim}{X}(t)$ is given. When estimating *steady state* properties, the simulations are conducted on the discrete time Markov chain conversion $\underset{\sim}{X}_i$. The expected state sojourn (holding) times are used instead of sampling these. This will always give a variance reduction compared to simulations on the original $\underset{\sim}{X}(t)$. This also applies to simulation of *semi-Markov processes* [GSHG92].

3.3 Change of measure in dependability and traffic models

As demonstrated in previous section, the efficiency of importance sampling is sensitive to the choice of the new sampling distribution $f^*(\underset{\sim}{s})$. In figure 3.2, the sensitivity to different parameters is demonstrated by results from simulations on an M/M/1/N queue with blocking probabilities in order of 10^{-9}. The figure shows a typical situation where the precision of the estimator decreases and its variance increases on both sides of the optimal change of measure. The lesson learned from this figure is that for optimal parameters, or parameters close to these, very good results can be obtained. Otherwise, importance sampling may produce poor results. Section 3.5 will return to this and show how the theoretical variance of the estimates rapidly increases (to infinity) on both sides of the optimal change of measure, f^*_{opt}. In a more complex model, such as a network model, similar behaviour is observed. The difference is that the shaded region of figure 3.2 will be narrower, and hence, it is even more important to have good means to obtain optimal parameters.

The optimal change of measure is not easy to obtain. A lot of work has been done on defining (asymptotically) optimal solutions, exact or by pre-simulations. In this section, a brief summary is given, and for a comprehensive survey the reader is referred to [Hei95].

3.3.1 Dependability vs. traffic models

A number of different approaches is proposed for obtaining efficient importance sampling simulations. The solutions are dependent on the nature of the problem. Models describing dependability (or reliability) and traffic (or queuing) aspects are basically very different.

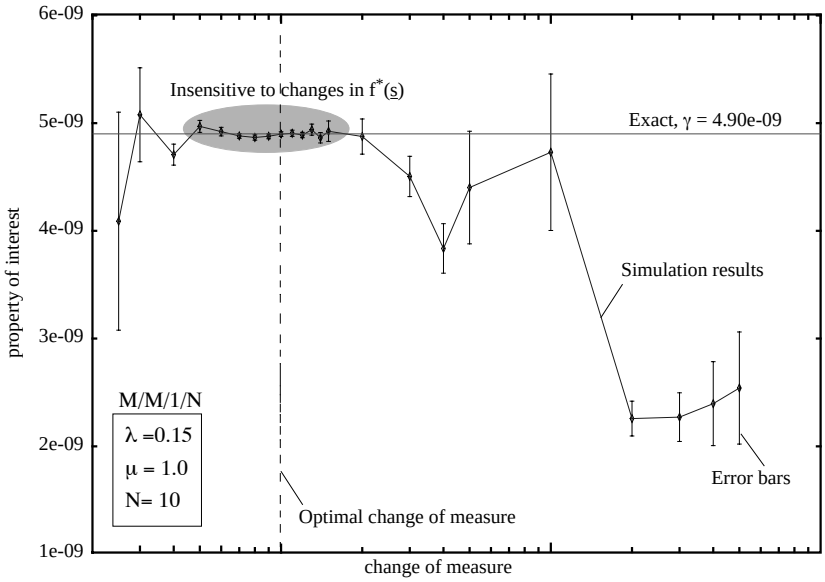


Figure 3.2 : Importance sampling efficiency is dependent on the change of measure.

- *Dependability model* - the events of interest are typically system failures caused by occurrence of a (small) number of element or component failures during a (short) period of time, e.g. the restoration time of the first failure. If each of the component failures are unlikely to happen in the observation period, the system failure is a rare event.

- *Traffic model* - the events of interest are typically buffer overflow or trunkline blocking. These events are due to a (large) number of simultaneous call arrivals in a service period. It is not a *single* call arrival in a service period that is the rare event, but the occurrence of a large number of simultaneous call arrivals. This becomes a rare event when the queue or trunk line capacity are large, and/or the offered load is low and server distributions are heavy tailed.

To demonstrate the different limiting behaviour of the dependability and traffic models, consider a *basic event* with probability ε. A basic event is, in a dependability model, a component failure during a restoration period, and in a traffic model, arrival of a call in a service period. Let the number of basic events leading to the rare event of interest be n.

Assume that the probability of rare events are exponential in the number of basic event, ε^n. This probability goes to 0 in two different ways:

- In *dependability* models: $\varepsilon^n \to 0$ as $\varepsilon \to 0$ if n is constant, or
- In *traffic* models: $\varepsilon^n \to 0$ as $n \to \infty$ and $\varepsilon < 1$.

Example 3.2: The difference is illustrated in figure 3.3. A sequence of basic events is leading to a rare event of interest within a specified observation period, $T = 1000$. In the dependability model, a system failure is observed after a few basic events, while in the traffic model a large number of basic events is leading to the occurrence of blocking or overflow.

This is the main reason why an optimal, or at least good, change of measure in a traffic model does not necessarily directly apply in a dependability model, and vice versa. For instance, one of the optimal solutions for the traffic models assumes a large number of basic events to substitute the discrete sequence of events by a continuous trajectory. This approximation is required to apply asymptotic results from the large deviation theory. In a dependability model with small n, this approximation is too rough.

3.3.2 Approaches for dependability models

The change of measure in dependability models is normally done by changing either the failure rates, the restoration or repair times, or the observation period. The rare event of interest is typically the system failure caused by occurrence of a specific combination (or sequence) of component failures. In a real sized system the number of components is large and so also the number of different system failure modes. In general, the objective of the change of parameters is to sample the most likely paths to failures. This means that the most important system failure modes must be identified, and their corresponding sequence of events leading to this failure.

3.3.2.1 Simple failure biasing

A straightforward approach is the *simple failure biasing*, thoroughly treated in [Nak94]. The failure and repair rates are scaled by a common factor, denoted p in this section. The idea is to increase the frequency of failures by increasing the failure rates and reducing the repair rates in proportion to the original rates. Let the original failure and repair rates for

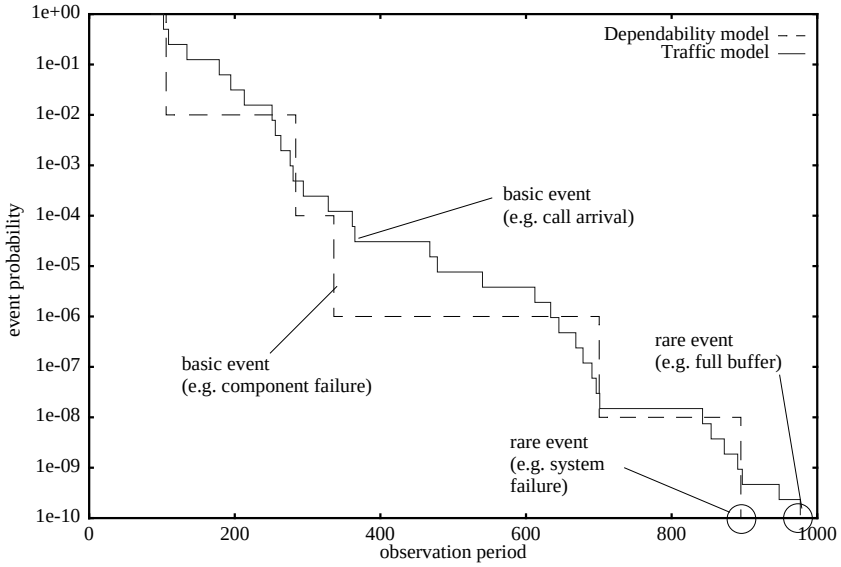


Figure 3.3 : The rare events in a dependability model occur typically after a small number of basic events, while in traffic models the number of basic events that results in a rare event is large.

component k at state $\underset{\sim}{\omega}$ be denoted $\lambda_k(\underset{\sim}{\omega})$ and $\mu_k(\underset{\sim}{\omega})$, respectively. The *simple failure biasing* changes the failure rates to $\lambda_k^*(\underset{\sim}{\omega})$ and repair rates to $\mu_k^*(\underset{\sim}{\omega})$ as follows:

$$\begin{aligned}\lambda_k^*(\underset{\sim}{\omega}) &= \lambda_k(\underset{\sim}{\omega}) \cdot p / \sum \lambda_k(\underset{\sim}{\omega}) \\ \mu_k^*(\underset{\sim}{\omega}) &= \mu_k(\underset{\sim}{\omega}) \cdot (1-p) / \sum \mu_k(\underset{\sim}{\omega}) .\end{aligned} \tag{3.7}$$

Example 3.3: Consider a component k which is either “ok” or “defect”. The failure and repair rates are:

$$\lambda_k(\underset{\sim}{\omega}) = \begin{cases} \lambda_k & \underset{\sim}{\omega} = \text{ok} \\ 0 & \underset{\sim}{\omega} = \text{defect} \end{cases},$$

$$\mu_k(\underset{\sim}{\omega}) = \begin{cases} 0 & \underset{\sim}{\omega} = \text{ok} \\ \mu_k & \underset{\sim}{\omega} = \text{defect} \end{cases}.$$

According to [Hei95], the factor p is typically $0.25 < p < 0.9$. However, figure 3.4 shows the results from simulations of the same 2 component parallel system that was used

in [CG87]. The simulation results with error bars indicate that a factor $p^* = 0.99$ should be chosen, not $p^* = 0.50$ as was recommended in [CG87].

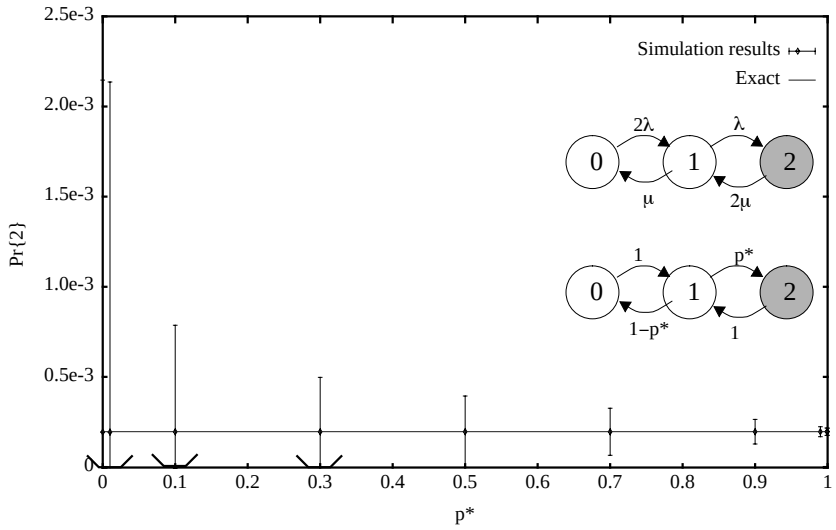


Figure 3.4 : The effect of failure biasing on the example from [CG87]. The optimal change of measure is λ□BIAS= (μ/λ)·λ = μ = 0.9901. (λ=0.01, μ=1.0).

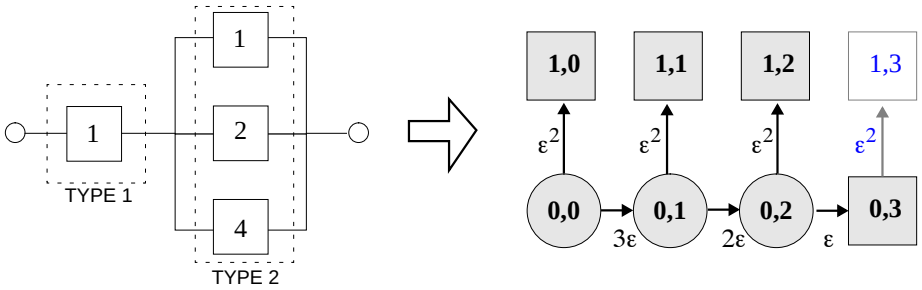
The simple failure biasing is efficient when the system is *balanced*, i.e. the number of failure transitions to a system failure is the same for all component types. It is proven [HSN94, Nak94], that simple failure biasing has *bounded relative error*[1], r.e.$(\hat{\gamma}_{IS}) < \infty$, i.e. a fixed number of samples is required to estimate $\hat{\gamma}_{IS}$ to a certain level of confidence, irrespective of the probability of the rare event of interest.

3.3.2.2 Balanced failure biasing

For an example of an *unbalanced* system, consider the system in figure 3.5 taken from [Hei95]. The system consist of 2 component types. The system failure occurs either as a result of 1 failure of type 1, or 3 failures of type 2. Let the failure rate of type 1 be ε^2, and for type 2, ε. Assume that no repair actions take place. Hence, the system is denoted *unbalanced* because the dependability, e.g. measured by the MTFF, is dominated by failures of the single component. However, when $\varepsilon^2 \ll \varepsilon \ll 1$, simulation by use of simple failure biasing, will most frequently generate a path towards the 3 component failure. This

1. Relative error is the ratio between standard error and sample mean, r.e.$(\bar{X}) = S_{\bar{X}} / \bar{X}$, see appendix B.3.

is not the most likely path. Application of simple biasing may result in estimates having *unbounded relative error*.



(a) Block diagram structure (b) State space model

Figure 3.5 : Example of unbalanced system, taken from [Hei95].

Instead, *balanced failure biasing* is proposed [GSHG92]. Uniform weights are now put on the failure rates of the different component types, i.e.:

$$\begin{aligned} \lambda_k^*(\omega) &= \lambda_k(\omega) \cdot p / K \\ \mu_k^*(\omega) &= \mu_k(\omega) \cdot (1-p) / \sum \mu_k(\omega) . \end{aligned} \tag{3.8}$$

This means that after scaling, the failure rates are equal for each of the K component types. The repair rates are scaled similar to the simple failure biasing case, see (3.7). Now, the most likely path in the example of figure 3.5 will be sampled more frequently during simulation. However, when simulating a balanced system, simple failure biasing is more efficient than balanced biasing, e.g. see results in [LS95].

3.3.2.3 Pre-simulations for determination of the *p*-factor

The optimal scaling factor p from (3.7) and (3.8), is not easy to obtain analytically. In some cases it is considered to be impossible. Instead, [DT93] proposes an experimental approach where a series of *short* pre-simulations is conducted for different p. The relative error, r.e.$(\hat{\gamma}_{IS})$, is plotted for different p, like for instance in figure 3.2. The optimal value of p, is the factor providing the minimum relative error. An optimisation technique, based on the *simulated annealing* technique, is applied to the series of results to determine the optimal factor from the experimental data. Consider, for instance, the results plotted in

figure 3.2 to be estimates obtained from a series of pre-simulations. Then, a p factor within the shaded region would have been chosen as the optimal p.

3.3.2.4 Failure distance biasing

The failure biasing approaches in (3.7) and (3.8), propose a fixed scaling factor p for the entire simulation experiment. An alternative approach is called *failure distance biasing* [Car91]. The scaling factor is changed during the experiment and is adapted to the current state of the system. The idea is to change the failure rates to give most weight to the components involved in the system failure which is closest to the current state. *Closest* in the sense that it is involving the least number of failure events/transitions to a system failure. When these components are determined from the fault tree of the system and the current state, the transition rates are changed using an "on-the-fly" optimisation, see [Car91] for details.

The adaptive parameter biasing technique, that will be described in chapter 5, is inspired by this failure distance biasing. However, instead of counting the number of transitions to each system failure, the new idea is to assign *probabilities* to each sequence of transitions from the current state up to a given system failure, and also consider the value of the *contribution* to the system dependability given that this failure has occurred.

3.3.3 Approaches for traffic models

As pointed out in section 3.3.1, the rare events in traffic models occur as a result of a large number of basic events. The sequence of events is denoted a *path*. In chapter 4, the concept of a path is presented together with other details of the modelling framework, see also appendix B for an overview. Typically, very efficient simulations are achieved when an optimal change of measure is available. On the other hand, changing to parameters far from the optimal, poor simulation results are produced.

3.3.3.1 Borovkov heuristics applied to GI/GI/1

A few (asymptotically) optimal changes of measures exist for a limited class of models. This section will focus on one model in particular, the *general single server queue*, GI/GI/1/N. The optimal change of measure is obtained by the use of results from the *large deviation theory*, see [Buc90] for a rigorous description. This result is one of the fundaments of the adaptive technique that will be proposed in chapter 5.

Let S_n/n be the sample mean of n samples s taken from the distribution $f(s)$, where $S_n = \sum_{r=1}^{n} s_r$. Roughly, combining Cramér's theorem with Chernoff bounds [Buc90, RME96], the probability of this sample mean being greater than or equal to a certain value y, is:

$$P(S_n/n \geq y) \approx e^{-nI(y)} \tag{3.9}$$

where the *Cramér's transform* or the *entropy function* is:

$$I(y) = \sup_{\theta \in R} (\theta y - \log M(\theta)). \tag{3.10}$$

$M(\theta)$ is the moment generating function of $f(s)$.

Example 3.4: If $f(s) = \lambda e^{-\lambda s}$, i.e. the exponential distribution, the moment generating function is $M(\theta) = E_f(e^{sa}) = \lambda/(\lambda - a)$, and then the Cramér's transform becomes $I(y) = \lambda y - 1 - \log \lambda y$. In [RME96] more examples are given.

In figure 3.6 the Cramér's transform for the exponential distribution with mean value 10 ($\lambda = 1/10$) is plotted. Observe the exponential increase on both sides of the mean value. Equation (3.9) describes how the probability decreases as the sample mean deviates from the expected value.

In [PW89], (3.9) was heuristically applied to obtain an asymptotically optimal change of measure for GI/GI/1 and tandem queues, using the following reasoning. The queue is observed over a period of time (0,T]. The queue is empty at time $t = 0$, and reaches the capacity N for the first time at $t = T$, without returning to empty for t in (0,T]. The observed arrival and departure rates in this interval are constant and denoted λ' and μ', respectively. The queue grows with a rate $\lambda' - \mu'$ ($\lambda' > \mu'$) and reaches N at time T, i.e. $N = T(\lambda' - \mu')$. The probability of observing an arrival rate of λ' over an interval (0,T], is equal to the probability of observing a mean interarrival of $1/\lambda'$ in $\lambda' T$ samples. The Cramér's theorem now applies, substituting $y = 1/\lambda'$ and $n = \lambda' T$ into (3.9). Correspondingly for the observed departure rate μ'. The probability of observing λ' and μ' is then, substituting T by $N/(\lambda' - \mu')$:

$$p(\lambda', \mu') = e^{-T(\lambda' I_\lambda(1/\lambda') + \mu' I_\mu(1/\mu'))} = e^{-N\frac{\lambda' I_\lambda(1/\lambda') + \mu' I_\mu(1/\mu')}{\lambda' - \mu'}} \tag{3.11}$$

where the $I_\lambda(1/\lambda')$ is the Cramér's transform for the arrival distribution.

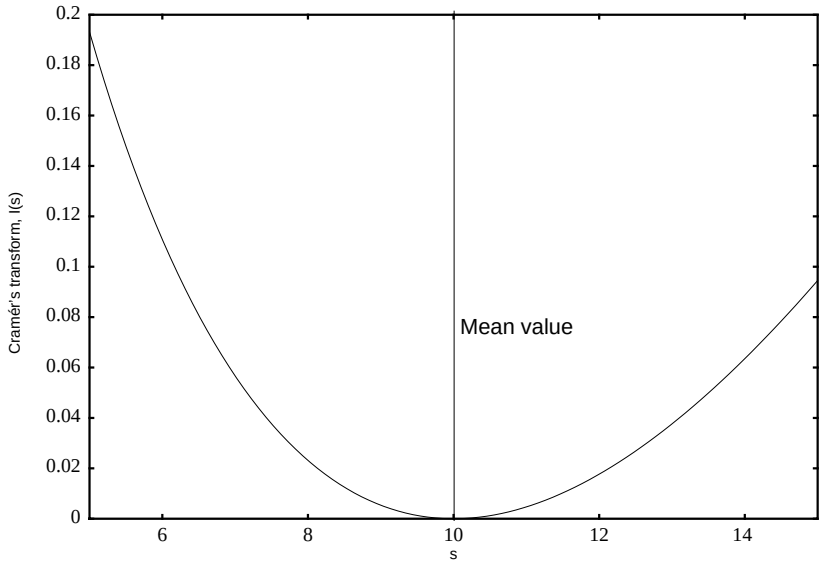


Figure 3.6 : The Cramér's transform for the exponential distribution with rate λ=1/10.

The objective is to maximise $p(\lambda', \mu')$ with respect to λ' and μ', which is the same as minimising the exponent of (3.11), without the constant term $-N$:

$$\beta(\lambda', \mu') = \frac{\lambda' I_{\lambda}(1/\lambda') + \mu' I_{\mu}(1/\mu')}{\lambda' - \mu'}. \tag{3.12}$$

Example 3.5: Consider an M/M/1/N queue where the objective is to estimate the probability of reaching the buffer capacity N. The Cramér's transforms is

$$I_{\lambda}(1/\lambda') = \lambda/\lambda' - 1 - \log\lambda/\lambda'$$
$$I_{\mu}(1/\mu') = \mu/\mu' - 1 - \log\mu/\mu'$$

substituted into (3.12)

$$\beta(\lambda', \mu') = \frac{\lambda - \lambda' - \lambda'\log\lambda/\lambda' + \mu - \mu' - \mu'\log\mu/\mu'}{\lambda' - \mu'}.$$

The $\beta(\lambda', \mu')$ is plotted in figure 3.7 for different values of λ' and μ'. The non-trivial[1] solution to $\min(\beta(\lambda', \mu'))$, is the interchange of the original arrival and departure rates:

$$\lambda' = \lambda^* = \mu \text{ and}$$
$$\mu' = \mu^* = \lambda. \tag{3.13}$$

The result in example 3.5 is well known and has also been derived by use of *slow-random walk*, [CFM83]. This assumes that N is large to make the sequence of arrival and departures a continuous trajectory in the state space. The arrival and departure rates need not be constant over the entire trajectory as was assumed in the heuristics described above. The relation between this and slow-random walk approach is given in [PW89].

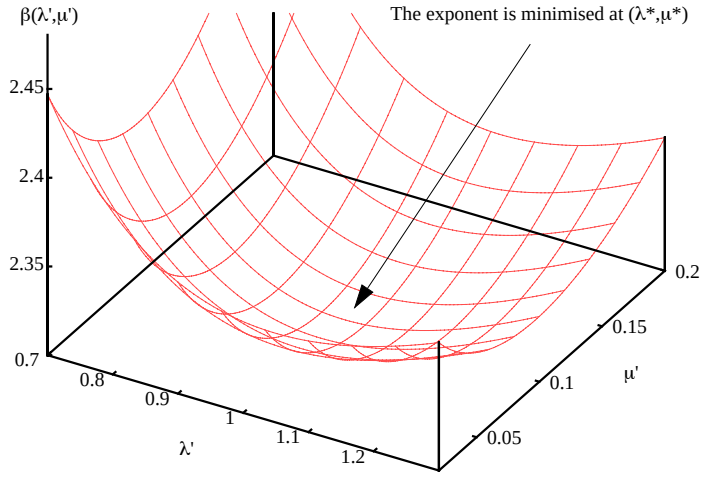


Figure 3.7 : The exponent $\beta(\lambda', \mu')$ *for various* λ' *and* μ' *where* $\lambda = 0.1$ *and* $\mu = 1$.

3.3.3.2 Change of measure based on large deviation results

Most of the work on successful application of importance sampling to traffic models has used large deviation results to determine (asymptotically) optimal changes of measure. The results from [CFM83, PW89] for the GI/G1/1 queue are extended to multi-server queues in [Sad91, KW92, KW93, Man95]. Similar results are obtained for Erlang loss systems with batch arrivals [Man96b] and for fluid flow models [Man96a].

1. The trivial solution is to substitute the original parameters, $\lambda' = \lambda$ and $\mu' = \mu$.

The extension of the results to tandem queues are only efficient if the load on queues is significantly dominated by a single queue, [PW89, GK95, FA94]. Otherwise, using the scaling as proposed by application of large deviation, will result in an inefficient simulation. In [GK95], conditions are developed for the applicability of results. For Jackson networks [Fra93], it is required that the performance is constrained by a single queue or node.

Generally, in a well engineered network, the resource utilisation is balanced, which means that no single bottleneck dominates the system performance. This implies that using the results mentioned above will not give an efficient simulation. For some specific (and small) network structures, large deviation results exists, for example *intrees* [CHJS92], and *feedforward network* [Maj97]. However, these results are not applicable to more general topologies with balanced utilisation of resources. Hence, an engineering approach is developed, which combines previous large deviation results with the ideas of failure distance biasing [Car91]. This adaptive biasing approach will be described in Chapter 5, and is also published in [Hee97a, Hee98].

Large deviation results are applied to set up efficient importance sampling simulation of the cell loss ratio in packet switched network like the ATM, see [CHS95] and for consecutive cell losses in ATM [NH95].

The following section introduces a scaling of the parameters similar to the failure biasing of section 3.3.2, that will be used in the remaining of this thesis. The scaling is denoted *parameter biasing*, and its use is demonstrated on a few models.

3.4 The change of measure

For easy implementation of the large deviation results to simulations with importance sampling, this section will introduce a scaling of the arrival and departure rates of the simulation model, $\underset{\sim}{X}(t)$. This parameter biasing is applied to several queuing and Erlang loss models. The relation between these biasing factors, and the optimal change of measure obtained by large deviation results, is described.

3.4.1 The change of transition rates

The *change of measure* refers to the change of the sampling distribution from $f(\underline{s})$ to $f^*(\underline{s})$. The *biasing* approach described in this section, applies to $\underset{\sim}{X}(t)$, or its embedded process $\underset{\sim}{X}_i$. When simulating the steady state properties, only the event probabilities in

$\underset{\sim}{Z}_i$ will be changed. This is described in section 3.4.1.1. When transient properties are of interest, the $X(t)$ is simulated. Then both the underlying event probabilities and the state sojourn times are changed. The latter is described in section 3.4.1.2.

3.4.1.1 The sample path distribution

Let $\underset{\sim}{s}$ be the sample representing a sequence of events, e.g. arrivals and departures in the $\underset{\sim}{X}_i$. The original arrival and departure rates for dimension k are denoted, $\lambda_k(\underset{\sim}{\omega})$ and $\mu_k(\underset{\sim}{\omega})$ respectively. The distribution of the sample path $f(\underset{\sim}{s})$ is defined as:

$$f(\underset{\sim}{s}) = \prod_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} P_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}} \text{with} P_{\underset{\sim}{x}, \underset{\sim}{y}} = \begin{cases} \lambda_k(\underset{\sim}{x}) / G(\underset{\sim}{x}) & \text{if } \underset{\sim}{y} = \underset{\sim}{x} + \underset{\sim}{1}_k \\ \mu_k(\underset{\sim}{x}) / G(\underset{\sim}{x}) & \text{if } \underset{\sim}{y} = \underset{\sim}{x} - \underset{\sim}{1}_k \wedge \underset{\sim}{y} \notin \underset{\sim}{\Omega}_0 \\ 1 & \text{if } \underset{\sim}{y} \in \underset{\sim}{\Omega}_0 \end{cases} \quad (3.14)$$

The $\underset{\sim}{\Omega}_0$ is the regeneration state space, as was mentioned in section 2.1 and will be described in more details in section 4.3. The *normalisation* factor is $G(\underset{\sim}{x}) = \sum_{k=1}^{K} \lambda_k(\underset{\sim}{x}) + \sum_{k=1}^{K} \mu_k(\underset{\sim}{x})$.

Changing the $f(\underset{\sim}{s})$ means changing the transition rates with respect to determination of $P_{\underset{\sim}{x}, \underset{\sim}{y}}$. The new sampling distribution is:

$$f^*(\underset{\sim}{s}) = \prod_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} P^*_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}} \text{ with}$$

$$P^*_{\underset{\sim}{x}, \underset{\sim}{y}} = \begin{cases} \lambda^*_k(\underset{\sim}{x}) / G^*(\underset{\sim}{x}) & \text{if } \underset{\sim}{y} = \underset{\sim}{x} + \underset{\sim}{1}_k \\ \mu^*_k(\underset{\sim}{x}) / G^*(\underset{\sim}{x}) & \text{if } \underset{\sim}{y} = \underset{\sim}{x} - \underset{\sim}{1}_k \wedge \underset{\sim}{y} \notin \underset{\sim}{\Omega}_0 \\ 1 & \text{if } \underset{\sim}{y} \in \underset{\sim}{\Omega}_0 \end{cases} \quad (3.15)$$

now $G^*(\underset{\sim}{x}) = \sum_{k=1}^{K} \lambda^*_k(\underset{\sim}{x}) + \sum_{k=1}^{K} \mu^*_k(\underset{\sim}{x})$.

The *likelihood ratio* from equation (2.26) is

$$L(\underset{\sim}{s}) = \frac{f(\underset{\sim}{s})}{f^*(\underset{\sim}{s})} = \frac{\prod_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} P_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}}}{\prod_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} P^*_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}}} = \prod_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} \frac{P_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}}}{P^*_{\underset{\sim}{\omega}_i, \underset{\sim}{\omega}_{i+1}}}. \quad (3.16)$$

3.4.1.2 The state sojourn times

While simulating transient properties, the change of arrival and departure rates will also affect the state sojourn time distribution. The t_i is the simulation time at embedded

point i, and $(t_{i+1} - t_i)$ is the sampled time between two embedded points i and $i+1$, i.e. the state sojourn time of state $\underset{\sim}{\omega}_i$. The likelihood ratio is now:

$$L_t(\underset{\sim}{s}) = L(\underset{\sim}{s}) \cdot \prod\nolimits_{\forall(\underset{\sim}{\omega}_i \in \underset{\sim}{s})} \frac{G(\underset{\sim}{\omega}_i) e^{-G(\underset{\sim}{\omega}_i)(t_{i+1} - t_i)}}{G^*(\underset{\sim}{\omega}_i) e^{-G^*(\underset{\sim}{\omega}_i)(t_{i+1} - t_i)}} \tag{3.17}$$

where $L(\underset{\sim}{s})$ is given in (3.16).

In the following section, a scaling factor of the arrival and departure rates is introduced.

3.4.2 Biasing the parameters

A common factor is applied to scale up the arrival rates and to scale down the departure rates of the simulation model. The factor is denoted BIAS factor in this paper. With state dependent rates, the BIAS-factor will also be state dependent, $\mathrm{BIAS}(\underset{\sim}{\omega})$, i.e. the biasing will change as the system state changes. Furthermore, if the system can observe rare events in J different queues, an individual BIAS factor must be assigned for every j, $\mathrm{BIAS}_j(\underset{\sim}{\omega})$. Finally, the BIAS factor may depend on the resource requirements c_{kj} assigned to an event triggered by the traffic type described by generator k, $\mathrm{BIAS}_{kj}(\underset{\sim}{\omega})$. The BIAS factor is:

$$\begin{aligned} \lambda^*_k(\underset{\sim}{\omega}) &= \lambda_k(\underset{\sim}{\omega}) \cdot \mathrm{BIAS}_{kj}(\underset{\sim}{\omega}) \\ \mu^*_k(\underset{\sim}{\omega}) &= \mu_k(\underset{\sim}{\omega}) / \mathrm{BIAS}_{kj}(\underset{\sim}{\omega}). \end{aligned} \tag{3.18}$$

The following sections give the heuristics for choosing the BIAS factor in a few application examples.

3.4.2.1 Biasing in one dimensional models

An optimal change of measure exists for a GI/GI/1, see e.g. [PW89]. For an M/M/1 queue, the optimal change of measure has an explicit form:

$$\lambda^* = \mu \text{ and } \mu^* = \lambda. \tag{3.19}$$

Substituting (3.19) into (3.18), gives a state independent biasing:

$$\mathrm{BIAS}_{11}(\underset{\sim}{\omega}) = \mathrm{BIAS} = \lambda / \mu. \tag{3.20}$$

3.4.2.2 Biasing in multidimensional models

Recall from section 2.8.3, that a model with a *shared buffer* was described. Generally, a network model will consist of J shared buffers. Each of the K traffic types that are offering traffic to this network, will request capacity from an arbitrary set of buffers. From section 3.2, it is known that the simulation process, $X(t)$, operates on the state space Ω. The objective is to force this process towards either of J different subspaces, Ω_j where rare events of interest occur. Each of these subspaces corresponds to a shared buffer. In section 2.1 these subspace are denoted *target* subspaces, whilst chapter 4 describes the complete framework for modelling networks.

To explain the approach that is applied for biasing the parameters in such a model, an alternative interpretation of the optimal biasing in (3.13) is given in terms of *drift* in the simulation process. At current state ω the process experiences a *positive drift* towards each Ω_j, denoted $\delta_{j+}(\omega)$, ($j = 1, \ldots, J$). The total positive drift towards Ω_j is induced by the dimensions $k \in \Gamma_j$, i.e. all dimensions which have a transition that moves the process closer to Ω_j. The positive and negative drifts are defined as follows (c_{kj} is the number of resources of type j requested by the traffic type k, see chapter 4 and appendix B):

$$\begin{aligned} \delta_{j+}(\omega) &= \sum_{k \in \Gamma_j} c_{kj} \lambda_k(\omega) \\ \delta_{j-}(\omega) &= \sum_{k \in \Gamma_j} c_{kj} \mu_k(\omega). \end{aligned} \tag{3.21}$$

In the importance sampling distribution, $f^*(\underline{s})$, the positive drift must be *increased* to make the rare events of interest to occur more often. The heuristics, based on the result in (3.19), is that the positive drift under an importance sampling model, $\delta_{j+}^*(\omega)$, should be interchanged with the total *negative drift* under the original model, $\delta_{j-}(\omega)$, for every state ω, namely:

$$\begin{aligned} \delta_{j+}^*(\omega) &= \sum_{k \in \Gamma_j} c_{kj} \lambda_k^*(\omega) = \sum_{k \in \Gamma_j} c_{kj} \mu_k(\omega) = \delta_{j-}(\omega) \\ \delta_{j-}^*(\omega) &= \sum_{k \in \Gamma_j} c_{kj} \mu_k^*(\omega) = \sum_{k \in \Gamma_j} c_{kj} \lambda_k(\omega) = \delta_{j+}(\omega). \end{aligned} \tag{3.22}$$

One possible non-trivial solution to (3.22) is the following BIAS factor:

$$\mathrm{BIAS}_{kj}(\omega) = \mathrm{BIAS}_j(\omega) = \frac{\sum_{k \in \Gamma_j} c_{kj} \mu_k(\omega)}{\sum_{k \in \Gamma_j} c_{kj} \lambda_k(\omega)} = \frac{\delta_{j-}(\omega)}{\delta_{j+}(\omega)}. \tag{3.23}$$

To ensure that the drift towards the target is never reduced, the BIAS-factor must be greater than or equal to 1[1] for every state ω, and hence the following is used:

$$\mathrm{BIAS}_{kj}(\omega) = \max(1, \frac{\delta_{j^-}(\omega)}{\delta_{j^+}(\omega)}). \tag{3.24}$$

By letting $K = 1$, then (3.24) becomes $\mathrm{BIAS}_{11}(\omega) = \mu_1(\omega)/\lambda_1(\omega) = \mu/\lambda$, and the result in (3.20) is recognised.

The $\mathrm{BIAS}_{kj}(\omega)$ factor is applied to scale the arrival and departure rates of all $k \in \Gamma_j$, using (3.18), as follows:

$$\lambda_k^*(\omega) = \begin{cases} \lambda_k(\omega) \cdot \mathrm{BIAS}_{kj}(\omega) & \text{if } (k \in \Gamma_j) \\ \lambda_k(\omega) & \text{otherwise} \end{cases},$$
$$\mu_k^*(\omega) = \begin{cases} \mu_k(\omega) / \mathrm{BIAS}_{kj}(\omega) & \text{if } (k \in \Gamma_j) \\ \mu_k(\omega) & \text{otherwise} \end{cases}. \tag{3.25}$$

This will induce a positive drift in the simulation process towards a specific target subspace Ω_j.

3.4.2.3 Biasing in loss systems

Biasing in Erlang loss systems is a special case of the shared buffer model. Now, a *buffer* is a set of *trunklines*, and the arrival and departure rates are:

$$\lambda_k(\omega) = \lambda_k,$$
$$\mu_k(\omega) = \min(\omega_k, S_k)\mu_k, (S_k > 1) \tag{3.26}$$

where S_k is the number of dedicated servers for type k traffic (in the shared buffer model $S_k = 1$). For this specific system, an alternative scaling is described in [Kel86, Man96c] which is efficient when the number of trunklines is large. This section shows the relation between the scaling in the two approaches.

1. The BIAS factor may be allowed to take values below 1 if the objective is e.g. to force the process towards an empty system. In a large system, reaching empty system is a rare event which can be provoked by importance sampling. In this thesis, the empty state is not necessarily applied to complete the regenerative cycle, see description in chapter 6.

The approach in [Kel86, Man96c] is scaling the *traffic load*, i.e. the ratio λ/μ, offered to the system. According to [Man96c] the solution is:

$$\lambda_k^*/\mu_k^* = (\lambda_k/\mu_k)e^{-c_{kj}\cdot x_j} \tag{3.27}$$

where x_j satisfies $\sum_{\forall(k\in\Gamma_j)} c_{kj}\lambda_k/\mu_k \cdot e^{-c_{kj}\cdot x_j} = N_j$. The N_j is the capacity of resource type j, e.g. the number of channels in trunkline j.

To compare with the use of BIAS factors, substitute the arrival and departure rates from (3.26) into (3.23):

$$\lambda_k^*(\omega)/\mu_k^*(\omega) = (\lambda_k(\omega)/\mu_k(\omega))\mathrm{BIAS}_j^2(\omega) = \left(\frac{\lambda_k}{(\min(\omega_k, S_k)\mu_k)}\right)\mathrm{BIAS}_j^2(\omega)\,. \tag{3.28}$$

Then, the difference in two scaling approaches from (3.27) and (3.28) is given by comparing the two factors:

$$e^{-c_{kj}\cdot x_j} \text{ versus } \mathrm{BIAS}_j^2(\omega)/\min(\omega_k, S_k)\,. \tag{3.29}$$

The scaling in (3.27) is state independent but dependent on the resource requirements, c_{kj}, while (3.28) is only state dependent. A few simulations have been carried out to compare the two approaches. For large models, (3.27) is most efficient, while (3.28) is the most efficient for small models.

3.5 Heuristics and observations

A large number of simulation experiments with importance sampling is conducted on a variety of models. In chapter 6, simulations of complex, multidimensional network examples are reported. The heuristics presented in this section are based on observations from these simulations, in addition to simulations of simple one and two dimensional models. These models are studied because their analytic solutions are easily obtained, and comparisons between analytic and simulation results can be made.

This section contains three main results. *Firstly*, the experience with the use of the observed likelihood ratio as indication of goodness of simulation results is presented. It is observed that when the simulations produce good and stable estimates of the property of interest, the corresponding observed likelihood ratio is close to 1 (its expected value) and

with a small relative error. *Secondly*, it is observed from the basic equation of importance sampling, that it is possible to let $f^*(\underline{s}) = 0$ for every sample $\underline{s}$ where $g(\underline{s})f(\underline{s}) = 0$. This is implemented and tested on a few examples. It was discovered that a variance reduction was not guaranteed. *Thirdly*, the relation between the variance of the estimates and the changes in the BIAS factor is examined. In a model of an M/M/1/N queue, it is feasible to establish the bounds of a stable region of the BIAS factor. Outside this region, the variance is unbounded (infinite). This serves as an explanation to the observed simulation behaviour when (too) large BIAS factors have been applied. It has been observed that simulation of a finite number of replications under heavy biasing, will frequently result in a sample mean which is much less than the expected value. The reason is that the samples are found to be taken from a *heavy tailed distribution*.

All theoretical results used in this section are described in details in appendix D. The results are valid for models that are described in section 3.2. For one dimensional Markov chains the variance expressions can be rearranged to enable efficient calculations in large models. Appendix D includes numerical results from 3 different one dimensional models, and 1 example of a two dimensional model.

It is important to keep in mind that this section only addresses the accuracy of the importance sampling estimates with respect to the change of measure. No discussions relative to simulation stopping rules, simulation run length, transient periods, block vs. regenerative simulation, etc. are included. This is outside the scope of this thesis and the interested reader is referred to any textbook on simulation, e.g. [BFS87, LO88, BCN96].

3.5.1 The use of likelihood ratio for validation

As for all types of simulation, it is essential to ensure that the results of an importance sampling simulation are good. With *good* it is traditionally meant to produce accurate estimates close to the expected value. When the true values cannot be established elsewhere, neither through analytic solutions nor running a direct simulation, some indication of the goodness of the importance sampling estimates is required. It is important to note that a useful estimate in rare event simulations may be an estimate that is within the same order of magnitude as the expected value. This is generally a much weaker requirement to what a good simulation result is than the traditional one.

It has been discovered that some simulation results which apparently are good, because the relative error was low, have estimates that are *much less than the expected value*. This

is an observation that tells the simulation analyst that precautions must be taken when the expected value is not known. An incorrect conclusion about the correctness of estimates may be drawn if only the sample mean and its relative error are considered for validation of results. Figure 3.8 includes typical observations of the sample mean and its standard error produced by importance sampling simulations using a *too strong parameter biasing*. The estimates are all much below the exact value. Hence, the key point is to discover when a too strong biasing is applied, and some indication of this is required.

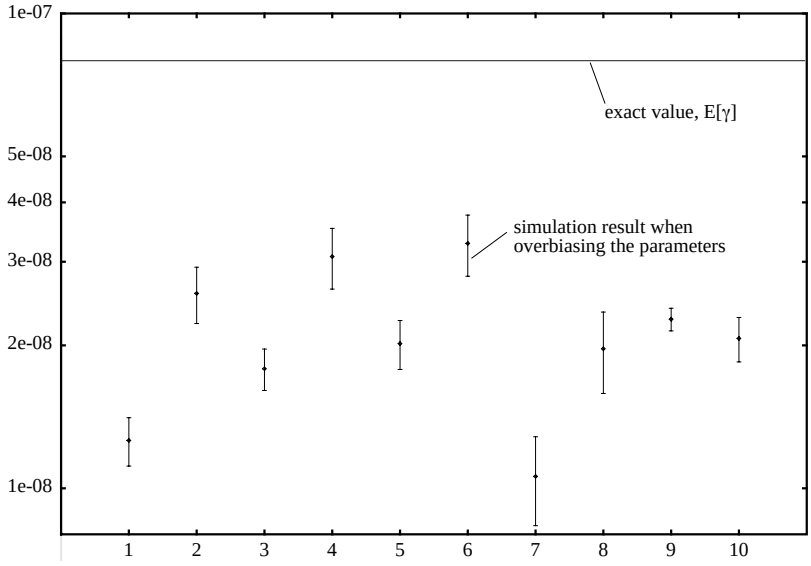


Figure 3.8 : When the parameter biasing is too strong, the estimated mean is often observed to be below the exact value but has small relative error.

Recall from (3.2) that $\hat{\gamma}_{IS}$ is obtained by observing several samples of the product of the property $g(s)$ and the likelihood ratio $L(s)$. Intuitively, this indicates that the observed likelihood ratio in a simulation can serve as an indication of how precise and stable $\hat{\gamma}_{IS}$ is. The expected value of the likelihood ratio is known for every change of measure and model, namely $E(L) = 1$. Hence, the observed mean value should be close to 1, $\bar{L} \approx 1$. The variance of L is dependent on the BIAS factor as the results in appendix D.7 demonstrates. The standard error of $\bar{L}$ is small when the BIAS factors are greater than 1 (BIAS=1 is direct simulation where the likelihood ratio is always 1 with standard error equal to 0). The standard error increases slowly with increasing BIAS factor. Above the optimal value of the BIAS factor, and then a *rapid increase* is observed. This means that a too strong biasing is possible to discover by studying the likelihood ratio.

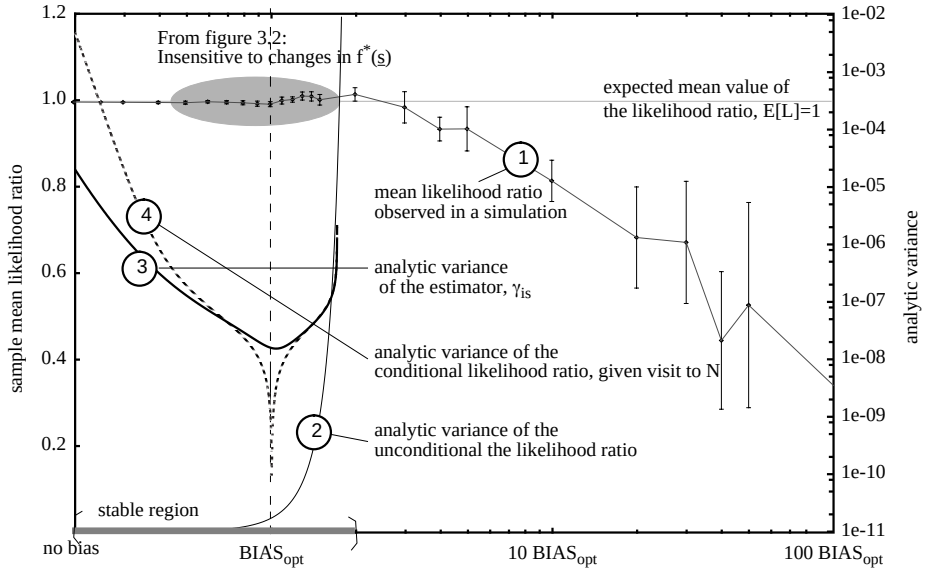


Figure 3.9 : The observed likelihood ratio drops below the expected value when biasing outside the stable region is applied
(all results are from an M/M/1/N queue with λ=0.15, μ=1.0, N=10).

Example 3.6: Consider an M/M/1/N queue with an arrival rate $\lambda = 0.15$ and departure rate $\mu = 1$. The capacity is $N = 10$. A series of simulations are conducted for different BIAS factors and the results are plotted in figure 3.9. The mean observed likelihood ratio $\bar{L}$ is plotted with the standard error, $S_{\bar{L}}$, as error bars. The optimal change of measure is $\text{BIAS}_{opt} = \mu/\lambda = 6.67$. At BIAS_{opt}, the observed likelihood ratio is $\bar{L} \approx 1$ with r.e.$(\bar{L}) = S_{\bar{L}}/\bar{L} \ll 1$. The same is observed for the BIAS factors below this optimal value. For some BIAS factor > BIAS_{opt}, the typically observed likelihood ratio is $\bar{L} < 1$ with r.e.$(\bar{L}) \ll 1$, and in some rare cases is $\bar{L} \approx 1$ with r.e.$(\bar{L}) > 1$.

The same effect as observed in this example, is observed in many other importance sampling simulations. To explain this, consider the theoretical variance of the likelihood ratio, $Var(L)$, given in (D.38). The value of $Var(L)$ for different parameter biasing, is added as plot 2 in figure 3.9. The $Var(L)$ shows a rapid increase when the BIAS factor becomes larger than the optimal value. Outside the *stable region*, the bounds are given in section 3.5.3, the variance grows to infinity. This implies that when the BIAS is too large, an infinite number of samples must be taken to estimate a likelihood ratio close to 1, i.e. the likelihood ratio follows a heavy tailed distribution.

Figure 3.9 also includes the analytic variance of $\hat{\gamma}_{\text{IS}}$ (plot 3), and the *conditional* likelihood ratio, given that a visit to state N is observed, $Var(L|N)$ (plot 4). These two plots show that the variance increases rapidly on both sides of the BIAS_{opt}. From (3.2), it is observed that the $\hat{\gamma}_{\text{IS}}$ is strongly correlated with the conditional likelihood ratio. This means that for estimators like $\hat{\gamma}_{\text{IS}}$, the conditional likelihood ratio contains the same information as the $\hat{\gamma}_{\text{IS}}$ with respect to the variance.

In figure 3.10, the probability density for the likelihood ratio is plotted for the model in figure 3.9 for a specific BIAS=20. This is a biasing outside the stable region. The plot shows the relation between a specific observed value of the likelihood ratio and its probability. This is obtained by considering all possible paths from the origin (the regenerative state $\underset{\sim}{\Omega}_0 = 0$) to the target subspace ($\underset{\sim}{\Omega}_1 = N$) of an M/M/1/N queue. The most likely path contains N arrivals and no departures, the second most likely path contains $N + 1$ arrivals and 1 departure. The "extra" arrival and departure constitutes a *loop* in the Markov chain in figure 3.12. Each plot to the left in figure 3.10, represents the relation between the likelihood ratio for a path with i loops and absorption in $\underset{\sim}{\Omega}_1 = N$, and the corresponding probability of this path. In figure 3.11(a) and (c) the underlying data of these plots are given. To the right in figure 3.10, the plots of the corresponding relation between the likelihood ratio and its probability for the paths with i loops and absorption in state 0. This is the paths where no visits to the target subspace are observed. In figure 3.11(b) and (d) the underlying data of these plots are given.

The most important observation is that the density of the likelihood ratio is *heavy tailed*. Evidently, from figure 3.11, it becomes even more heavy tailed when the BIAS factor increases beyond the stable range. In a heavy tailed distribution, a very large number of samples are required to obtain a sample mean that is close to the expected value[1].

Example 3.7: Taking sample paths $\underset{\sim}{s}$ from a heavy tailed distribution causes problems with estimations of the properties of interest. The majority of samples in a simulation experiment will then be a *direct path* from the origin at state *0* up to state *N*. Obviously, the likelihood ratio $L(\underset{\sim}{s})$ has the same value when the sample path $\underset{\sim}{s}$ is the same. The sample mean, $\bar{L}$, will become much less than 1, and the standard error of sample mean will be small. However, once in a while (very rarely), a sample path with no visits to state N is observed. These paths have a (very) large contribution to the estimate of the

1. The *Pareto distribution* is an example of a heavy tailed distribution which, for some parameters, has no finite expectation.

likelihood ratio. Now, the L may be close to 1 due to this single contribution, but the corresponding standard error will be large.

If the likelihood ratio follows a heavy tailed distribution, then the same will be the case for the estimates of the property of interest. This is expected to be the case under more general model assumptions based on observations from simulation of other, more complex, systems. In figure 3.11 the details behind figure 3.10 are given, illustrating that the distribution becomes increasingly heavy tailed as the BIAS factor increases.

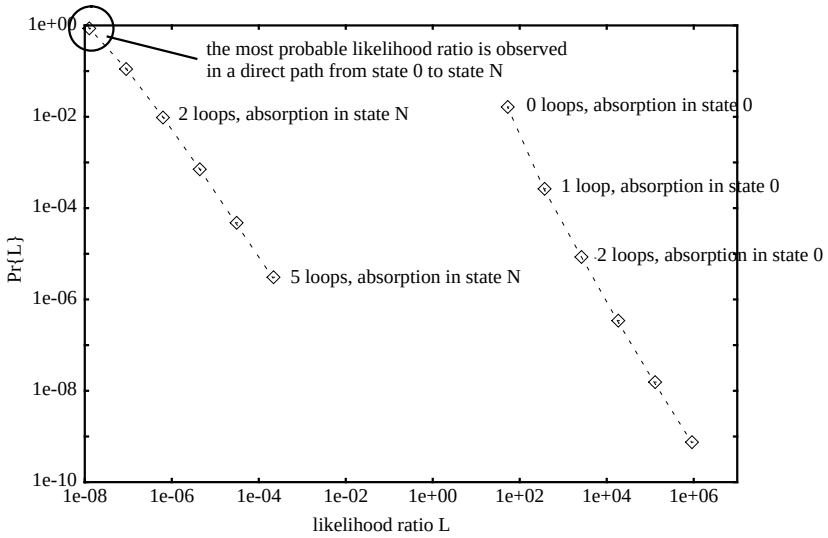


Figure 3.10 : The sampling distribution of the likelihood ratio is heavy tailed in the region where too strong biasing is applied. The plot is for an M/M1/N system with λ*=0.15,*μ*=1.0, N=10 with* BIAS $= 20$*, number of loops j=0,...,5.*

To summarise, consider the following two cases.

Case 1: The likelihood ratio is $\bar{L} \approx 1$ with r.e.$(\bar{L}) \ll 1$:

==> this indicates that the estimate $\hat{\gamma}_{IS}$ is *good* if its relative error r.e.$(\hat{\gamma}_{IS}) \ll 1$.

Case 2: The likelihood ratio is $\bar{L} \ll 1$ or r.e.$(\bar{L}) > 1$:

==> this indicates that the estimate $\hat{\gamma}_{IS}$ is *poor* even if the r.e.$(\hat{\gamma}_{IS}) \ll 1$.

The analytical results from the models in appendix D, confirm that the indication in case 1 is correct. The same has also been observed for a more complex model, see example 3.8. However, it does not provide the requested *guarantee* in the general case, see the counter example 3.9 below.

Example 3.8: Consider the non-trivial network example in section 6.3 with 10 user types and 12 nodes. The call blocking is estimated very accurate compared to the exact values. In this example $\bar{L} = 0.984$ with r.e.$(\bar{L}) = 0.0067$ and r.e.$(\hat{\gamma}_{\mathrm{IS}}) = 0.0308$, which confirms the indication in case 1.

Example 3.9: Recall the shared buffer example from section 2.8.3 with K user types with identical load. Consider one plot in this example where $K = 6$ and $\lambda = 0.30$. The likelihood ratio is $\bar{L} = 0.948$ with relative error r.e.$(\bar{L}) = 0.011$. The relative difference between the true value and the estimator, $(\gamma - \hat{\gamma}_{\mathrm{IS}})/S_{\hat{\gamma}_{\mathrm{IS}}} = 4.21$, i.e. more than 4 times the standard error. Hence, the estimates is not very accurate even though the relative error is low, r.e.$(\hat{\gamma}_{\mathrm{IS}}) = 0.161$. Note, however, that for engineering purposes this $\hat{\gamma}_{\mathrm{IS}}$ may still be of some value because the estimate is in correct order of magnitude (10^{-10}).

Case 2 is analytically confirmed on M/M/1/N queues and shared buffer models in appendix D. All results from simulation experiments, including more complex models, give reason to believe that this generally holds. However, the opposite is not necessarily true. For instance, a direct simulation gives poor estimates, but in this case the $\bar{L} = 1$ and r.e.$(L) = 0$.

3.5.2 Conditional return transition to regenerative sub-states

The only restriction for choosing $f^*(\underset{\sim}{s})$ given by (3.1), is that the new density must be $f^*(\underset{\sim}{s}) > 0$ for all samples $\underset{\sim}{s}$ where $f(\underset{\sim}{s}) \cdot g(\underset{\sim}{s}) \neq 0$. As a direct consequence, this means that it is possible to let $f^*(\underset{\sim}{s}) = 0$ for all samples where $f(\underset{\sim}{s}) \cdot g(\underset{\sim}{s}) = 0$, even if the original distribution $f(\underset{\sim}{s}) > 0$ for this sample.

This can be exploited in simulation of regenerative cycles. A sample paths $\underset{\sim}{s}$ is a regenerative cycle that starts and ends in $\underset{\sim}{\Omega}_0 = 0$, see figure 3.12. All paths which do not include a visit to $\underset{\sim}{\Omega}_1 = N$ should be disabled. The *disabling* is implemented simply by making the transition back to $\underset{\sim}{\Omega}_0$ dependent on whether a visit to $\underset{\sim}{\Omega}_1$ is observed or not. The transition back to $\underset{\sim}{\Omega}_0$ completes a regenerative cycles which is the observation period, or *sample*, of the simulation. In figure 3.12 below, this is illustrated by a one dimensional

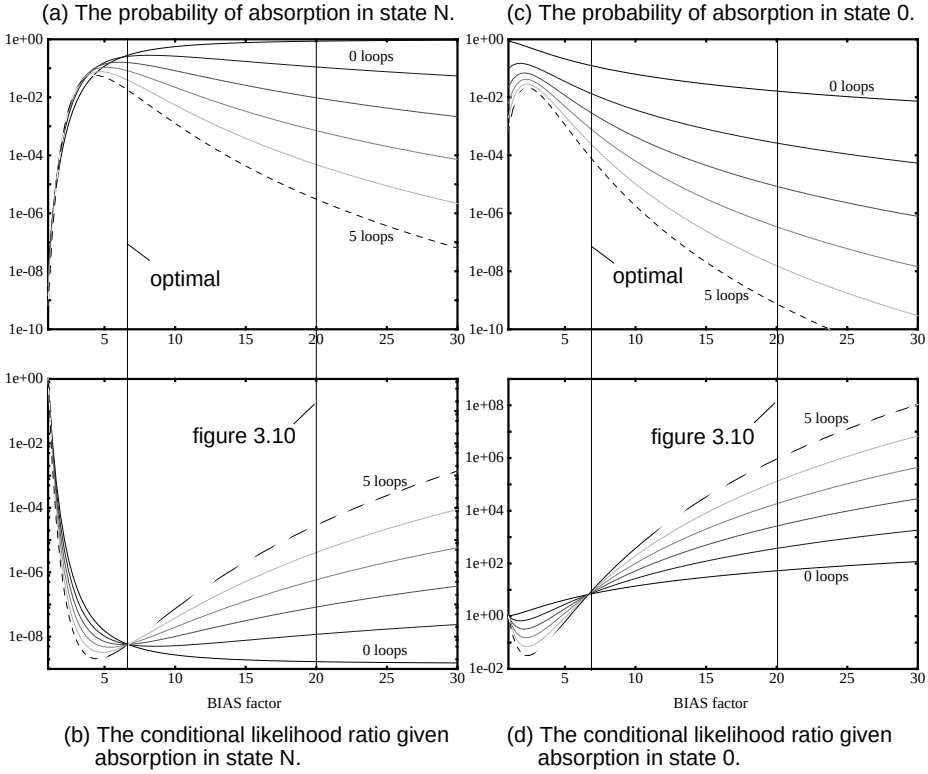


Figure 3.11 : The observed likelihood ratio has very large variability as the BIAS factor increases outside the stable region. (plot of M/M1/N system with λ=0.15,μ=1.0, N=10).

Markov chain. The state transition probability between state 1 and the regenerative state 0, is assigned to 0 as long as no visit to the target subspace N is observed. Then the transition probability between state 1 and 2 is 1. This implies that for all sampled paths, a rare event will be observed, because at least one visit to N will be included before the cycle is completed.

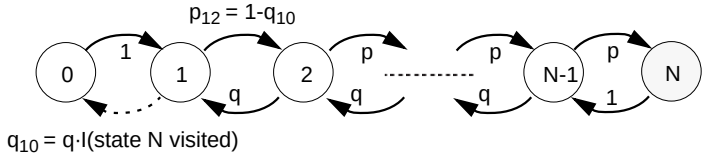


Figure 3.12 : Simple Markov chain with conditional return to regenerative state.

Simulation experiments have been conducted with a condition on the state transition back to the regenerative state. Denote this approach *conditional*, and the approach where no paths are disabled *unconditional*. Figure 3.13 shows the relative error r.e.$(\hat{\gamma}_{IS})$ of the estimates $\hat{\gamma}_{IS}$ for the conditional and unconditional cases. A minor variance reduction is observed for all BIAS factors in the conditional case. However, in other simulation experiments a variance *increase* is observed for some BIAS factors. To explain why variance reduction is not guaranteed in the conditional case, consider the variance expression of $\hat{\gamma}_{IS}$ from (D.41):

$$Var(\hat{\gamma}_{IS}) = E(L^2|N)p^*(N) - (E(L|N)p^*(N))^2 = E(L^2|N)p^*(N) - \gamma^2 . \tag{3.30}$$

In the conditional case, the probability of visiting state N in a cycle is $p^*(N) = 1$, and now (3.30) is:

$$Var_{cut}(\hat{\gamma}_{IS}) = E(L^2|N) - (E(L|N))^2 = Var(L|N) . \tag{3.31}$$

This implies that a comparison of the variance of the conditional versus unconditional case is the same as comparing the variance of the importance sampling estimates $Var(\hat{\gamma}_{IS})$ and the variance of the conditional likelihood ratio, given a visit to state N, $Var(L|N)$. This is known from appendix D where explicit expressions are developed. Figure D.7 shows that only for small systems, e.g. with $N = 10$, a variance reduction is observed $Var(\hat{\gamma}_{IS}) > Var(L|N) = Var_{cut}(\hat{\gamma}_{IS})$ for all BIAS factors. For larger systems, this inequality holds only for biasing close to the optimal BIAS factor.

In conclusion, it is not generally recommended to disable any paths, e.g. put conditions on the state transition back to Ω_0, because this does not give significant variance reduction, and it may cause a variance increase.

3.5.3 The stable region of the BIAS factor

As demonstrated in section 3.5.1, the variance of the likelihood ratio and the estimate of γ, are very sensitive to the change of measure. The same has been observed in many importance sampling simulations. In this section, the upper bound of the parameter biasing is determined for M/M1/N queues. Beyond this upper bound, the variance grows to infinity. The lower bound of the biasing region is direct simulation, i.e. BIAS=1.

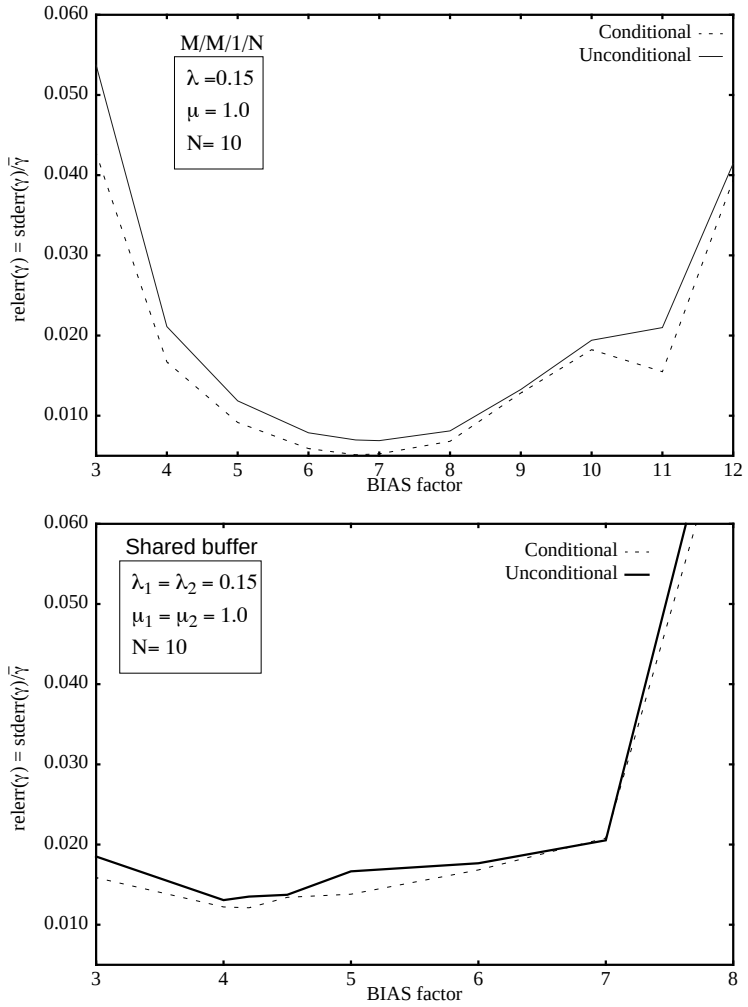


Figure 3.13 : Simulation results for two models where variance reduction is achieved by using conditional transition to $\underset{\sim}{\Omega}_0$ instead of unconditional transition.

In appendix D, the variance of the likelihood ratio in a single dimensional Markov chain is derived. As a part of the derivation, it was observed that the following condition must be fulfilled

$$\phi_2(\nu) = 4((pq)^2/p^*q^*)\cos^2(\nu\pi/N) < 1 \tag{3.32}$$

for all $\nu = 1, \ldots, N$. The p and q are the arrival and departure probabilities under original distribution, respectively, and the p^* and q^* are the corresponding probability under importance sampling distributions:

$$\begin{aligned} p^* &= \lambda^*/(\lambda^* + \mu^*) \text{ and} \\ q^* &= 1 - p^* \end{aligned} \tag{3.33}$$

To ensure $E(L^2|N) < \infty$ then

$$\max(\phi_2(\nu)) = \phi_2(1) < 1 \tag{3.34}$$

is a sufficient (but not necessary) condition, see section D.4.3 for details. Hence, substituting (3.32) into (3.34), the variance is finite when:

$$4((pq)^2/p^*q^*)\cos^2(\pi/N) < 1\,. \tag{3.35}$$

For the specific example given in figure 3.9, the numerical upper bound is $\text{BIAS} = 11.38$.

In a general system, it is very difficult to obtain similar expressions to provide an exact maximum limit of the BIAS factor on an explicit form. However, it is likely that the variance of the estimates grows to infinity also in more complex models. Case 2 in section 3.5.1 provides an indication that can serve as a means for detecting that the biasing of the importance sampling parameters is too strong.

3.6 Closing comments

This chapter introduces the basics of the importance sampling and contains a brief overview of some approximations for the change of measure. The focus is on approaches and results that are of importance to the adaptive parameter biasing that will be presented in chapter 5. In an example, it was demonstrated that the change of measure in importance sampling can be any distribution that fulfils the requirements of $f^*(\underline{s}) > 0$ for all samples $\underline{s}$ where $f(\underline{s})g(\underline{s}) \neq 0$. It was also pointed out, in the same example, that the computational complexity of the sampling algorithm should be considered in obtaining the “optimal change of measure”.

To change the sampling distribution in a Markov simulation, a BIAS factor is defined to scale the arrival and departure rates of the simulation processes. A few examples of the

use of this BIAS factor are included, and the relation to previously known biasing results are established. An extension of this BIAS factor is proposed for scaling in multidimensional models. This result is essential for the adaptive biasing technique in chapter 5.

The experiments reported in this chapter are based on simulations of simple models which can be compared with analytic results. These experiments, combined with the results of simulations of other more complex models, see chapter 6, can be summarised as follows:

1. The density of the likelihood ratio is *heavy tailed*. This means that a very large number of samples are required to obtain a sample mean that is close to the expected value.

2. It is not recommended to disable the transition back to Ω_0 as part of the importance sampling strategy. This reason is that this never gives significant variance reduction, and it may cause a variance increase.

3. In a general system, it is very difficult to obtain similar explicit expression to provide an exact maximum limit of the BIAS factor. However, it is likely that the variance of the estimate grows to infinity for a too strong biasing also for more complex models. Hence, it is very important to have a good indication of whether the biasing is too strong or not.

As an heuristic based on observations in 1 and 3, the *observed likelihood ratio* is proposed to serve as an indication of the accuracy of the importance sampling estimates.

4

Modelling framework for network simulations

In the previous chapter, a brief overview is given of some approaches for changing the measure of importance sampling. Most of them are valid for simple models only. In this thesis, network systems and their corresponding multidimensional models are the main focus. This chapter presents the details of the framework that will be applied for modelling traffic and dependability aspects of a communication network.

In section 4.1, a typical network is used as an example to motivate the various aspects of the framework that will be presented in section 4.3. The assumptions made regarding the underlying stochastic process are given in section 4.2. Section 4.4 summarises the modelling concepts and comments on what the challenges are with respect to the use of importance sampling simulation on models with multiple dimensions. The modelling concepts presented in this chapter are based on [Hee95b], and the extensions in [Hee97a, Hee98].

4.1 A typical network

Figure 4.1 shows the topology of a fictitious communication network covering most of Norway. This network will serve as an example to motivate the framework and to illustrate the various modelling concepts introduced.

The network, which may be regarded as a backbone, consists of 6 nodes that are interconnected by 10 links. The links contain the *resources* that are requested by the users that are offering traffic to this network. The network transports a traffic mixture from users with very different requirements, such as telephony, data transfer, cable-TV, Internet applications like email, web-browsing, and file transfer. This means that the users have different capacity requirements. They also require different quality of service, ranging from accept-

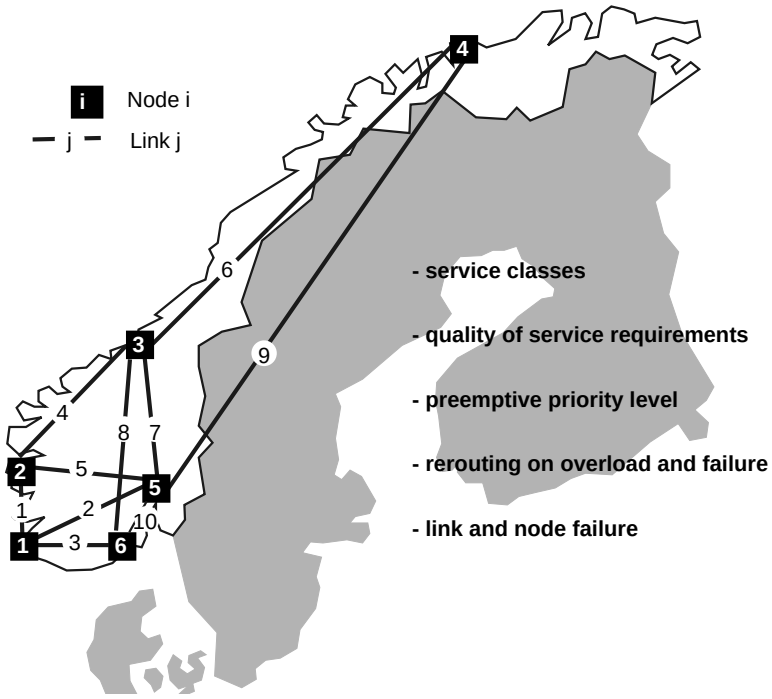


Figure 4.1 : A fictitious backbone network covering Norway.

ing no delay or no loss of information, to (almost) no limits regarding the quality of service tolerance, e.g. “World Wide Wait”. The price they are willing to pay should be in proportion to their requirements. The integration of such traffic requires a priority mechanism to distinguish between the users. This is important in overload situations where the users that are paying a high price for some guaranteed service must be given priority. As will be presented in the following, a preemptive priority mechanism will be introduced. The high priority users are given access to preempt the low priority users when insufficient capacity is available. The preempted calls are not resumed but are lost.

The basic building blocks of the modelling framework will be described in section 4.3. A key concept, defined in section 4.3.1, is the *generator* which is a collection of users with equal attributes. The user attributes are the traffic parameters (arrival rate and holding time, population size, resource capacity, etc.) and the network routing. The routing of a specific user is described as a sequence of links from which capacity is requested to setup a connection to a destination node.

To increase the quality of service that is offered by a network, a topology should be specified to make route redundancy feasible. This means that a pair of access nodes is interconnected by at least two link disjoint routes, see figure 4.1 for a topology example. The modelling framework defines a mechanism that allows the users to have alternate routes which may be invoked when the primary route is unavailable. A route is unavailable either due to traffic overload, or because either of the links of nodes along the route have failed.

The framework is also prepared for modelling of dependability aspects like link and node failures. This allows evaluation of a system considering both dependability and traffic aspects simultaneously. In section 4.3.6, a few comments on the applicability of the framework are made.

A performance analysis of steady state properties, typically includes:

- probability of blocking of a user with a specific priority level p,
- fraction of calls disconnected due to overload,
- fraction of calls disconnected due to a link or node failure,
- resource utilisation of the various links,
- number of rerouting events,
- etc.

The network has users with very high quality of service requirements. Hence, the performance measures depend on the occurrence of *rare events* like call blocking, rerouting, disconnections, etc. In the following chapter, an adaptive parameter biasing is described which makes the use of importance sampling simulation feasible to systems that are modelled by the framework presented in this chapter.

The underlying simulation process is briefly presented in section 4.2. When introducing importance sampling to increase the number of the rare events, this process must be changed, see description in chapter 3 for simple models. In chapter 5, a new, adaptive biasing will be introduced.

4.2 The simulation process

In its most general form, the model presented in this chapter is prepared for a simulation process like the *generalised semi Markov process* (GSMP) $\{X(t);(t \geq 0)\}$, see the description in [Gly89, Dam93]. This process operates on the system *state*, $\underset{\sim}{\omega}(t)$. At certain time epochs, an *event* is generated that triggers a possible change in the system state according to the *state transition* or *event probabilities*. The process stays in this new state for a certain amount of time, according to some *lifetime* distribution. The events in GSMP will occur at *embedded* points, but these need not be *renewal* points.

To simplify the importance sampling strategy in the following chapter, a less general simulation process is applied. The same process is described in section 3.2 and the description is repeated in this section for the sake of completeness. It is assumed that every embedded point, where the events occur, is a renewal point. Hence, the simulation process $\{\underset{\sim}{X}(t);(\underset{\sim}{\Omega}^K, t \geq 0)\}$ is a K-dimensional *continuous time Markov chain* (CTMC). It is defined on a discrete state space[1] $\underset{\sim}{\Omega}^K = \{0, 1, \ldots, M_k\}^K$. (When this is not ambiguous, the superscript K is removed). The events in $\underset{\sim}{X}(t)$ take place at embedded points in time. Let $t_0, t_1, \ldots, t_n$ be n embedded points, and then $\underset{\sim}{X}(t)$ can be discretizised in time by:

$$\underset{\sim}{X}_i = \underset{\sim}{X}(t_i) = \{\omega_{1;i}, \omega_{2;i}, \ldots, \omega_{K;i}\}, \text{ where } i = 1, \ldots, n. \tag{4.1}$$

The process $\{\underset{\sim}{X}_i;(\underset{\sim}{\Omega}, i > 0)\}$ is a *discrete time Markov chain* (DTMC) where the $\underset{\sim}{\omega}_i$ is the state occupancy after event i. The $\underset{\sim}{X}_i$ can be expressed by the following recursion,

$$\underset{\sim}{X}_i = \underset{\sim}{X}_{i-1} + \underset{\sim}{Z}_i \tag{4.2}$$

where $\underset{\sim}{Z}_i$ is the random event variable that describes the possible transitions out of state i, i.e. what events that can occur at this state. The example below shows a regular expression of the $\underset{\sim}{Z}_i$. In a model with preemptive priorities, the event variable becomes more complex, see example 4.4.

Example 4.1: Let an event affect only one dimension at a time. Thus, the event variable will assign values of either -1 or +1 in one of the K dimensions, and the following regular expression applies[2]:

1. Adding resource limitations to the model, e.g. finite buffer capacity, the feasible region of $\underset{\sim}{\Omega}^K$ will be reduced, e.g. a common resource limitations will cut the corners of this state cube.
2. The index vector of size K, $\underset{\sim}{1}_k = \{0, \ldots, 1, \ldots, 0\}$, is 1 at position k and 0 elsewhere.

$$\underset{\sim}{Z}_i = \begin{cases} -\underset{\sim}{1}_k & \text{with probability } q_{k;i},\ \forall k \\ \underset{\sim}{1}_k & \text{with probability } p_{k;i},\ \forall k \\ 0 & \text{with probability } 1 - \sum_{\forall k}(p_{k;i} + q_{k;i}) \end{cases} \tag{4.3}$$

The $q_{k;i}$ and $p_{k;i}$ are the state dependent transition probabilities.

In a two dimensional model, $K = 2$, like the example in figure 4.2, the process is operating on the state space $\underset{\sim}{\Omega}^2 = \{0, 1, \ldots, M_1\} \times \{0, 1, \ldots, M_2\}$. The system states are changed at every event according to (4.2) and by use of the following event variable:

$$\underset{\sim}{Z}_i = \begin{cases} \{-1, 0\} & q_{1;i} \\ \{1, 0\} & p_{1;i} \\ \{0, -1\} & q_{2;i} \\ \{0, 1\} & p_{2;i} \\ \{0, 0\} & 1 - (q_{1;i} + p_{1;i} + q_{2;i} + p_{2;i}) \end{cases} \tag{4.4}$$

The event probabilities, $q_{k;i}$ and $p_{k;i}$, $k = 1, 2$, depend on the system state $\underset{\sim}{\omega}_i$ after event i.

All simulations of *steady state* properties, e.g. call blocking probabilities, are conducted on the discrete time Markov chain (DTMC), $\underset{\sim}{X}_i$ conversion of the continuous time Markov chain (CTDM), $\underset{\sim}{X}(t_i)$. The expected lifetimes (state sojourn times) are used instead of a random sample from the lifetime distribution. This will always reduce the variance compared to simulations on the original $\underset{\sim}{X}(t_i)$, [GSHG92]. But, simulations of *transient* properties require that the continuous process $\underset{\sim}{X}(t_i)$ is applied.

4.3 Flexible simulation model framework

This section describes the flexible modelling framework. Section 4.3.1 presents the basic building blocks constituting a model. The system state of the model is defined on a state space as given in section 4.3.2. In section 4.3.3, it is described how the simulation process from the previous section operates on the model. The boundaries of the state space, as imposed by the resource capacities, are defined as target subspaces in section 4.3.4.

The flexibility of the framework is discussed in section 4.3.6 where the applicability to modelling of dependability aspects are presented. Finally, in section 4.3.7, the extensions to the framework are described, adding preemptive priority and rerouting mechanisms.

Figure 4.2 shows an example of the mapping between a sub-system of the example in figure 4.1, and its state space description.

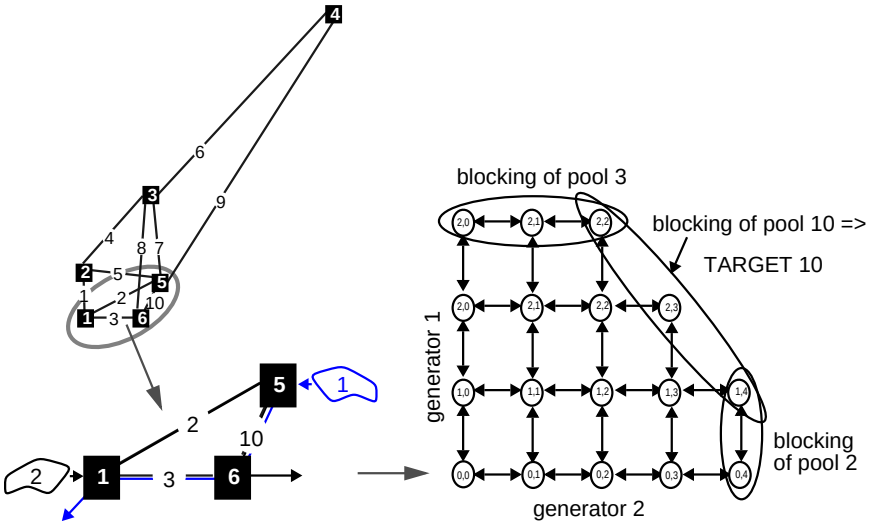


Figure 4.2 : The mapping of a system example to a state space model.

Example 4.2: Consider node 1, 5, and 6 of the example in figure 4.1. Traffic is offered from two different user types. Type 1 generates calls from node 5 to 1, via node 6. This means that channels, or bandwidth, from link 10 and 3 are required. Correspondingly, user type 2 sets up connections between node 1 and 6, via node 5. Now, resources from link 2 and 10 are required. Link 10 is then a common resource constraint for user type 1 and 2, in addition to the individual constraints given by link 3 and 2, respectively. The state space model is given to the right in the figure. Each traffic type corresponds to a dimension, and each link is a boundary that restricts the state space expansion.

4.3.1 Building blocks

The basic building blocks of the model are the K *generators* of *entities* which request resources from J *resource pools*. Figure 4.2 shows an example with generators added to

node 1 and 5 in the network example of figure 4.1. The resulting state space description is also included in the figure.

The building blocks are defined as follows:

Resource pool: a finite set of identical and interchangeable resources (e.g. a link in a network is considered to be a *pool* of channels, where the channels are *resources*):

- N_j is the total capacity of pool j, e.g. the number of channels or the total bandwidth on a link.

- $\Gamma_j = \{k | c_{kj} > 0 \wedge 1 \le k \le K\}$ is the set of generators with pool j as a constraint (see below for definition of c_{kj}).

Entity, e_k, $k = 1, \ldots, K$: an item that holds c_{kj} resources from pool j, (e.g. a connection generated by generator k between A and B parties holds c_{kj} channels from each link j along the route between A and B).

Generator (of entities): a component which explicitly models a process that generates the events operating on the entities:

- e_k is an entity from generator k,

- $\lambda_k(\underset{\sim}{\omega})$ is the state dependent arrival rate of generator k,

- M_k is the population size,

- $\mu_k(\underset{\sim}{\omega})$ is the state dependent departure rate of generator k,

- S_k is the number of servers for generator k entities,

- c_{kj} is the capacity requested from pool j by entities of generator k,

- $\Phi_k = \{j | c_{kj} > 0 \wedge 1 \le j \le J\}$ is the routing set, a fixed set of resource pools.

The capacity c_{kj} requested by an entity e_k, is in a *connection-oriented* network considered to be either a fixed number of communication channels on link j, or a specific fraction of the total bandwidth N_j. In a *connection-less* network, e.g. ATM or Internet, the c_{kj} is the *equivalent bandwidth* for an accepted call on link j.

The relations between the basic building blocks and

- *network* aspects like topology, resource capacities and service rates,
- *traffic* aspects of the users, including parameters like arrival rates, capacity constraints, priority, etc.

are described table 4.1. Observe that the topology of the network model is implicitly defined through the sets Φ_k and Γ_j. Note also that the departure rates μ_k are associated with the generators as part of the traffic model. This is because μ_k^{-1} is considered to be the mean duration of a call or the mean repair time.

Table 4.1: Relations between traffic and network models and the basic building blocks.

Queueing network model	Generator, k	Resource pool, j
Network model	Routing set, Φ_k	Resource pool capacity, N_j Generator constraint set, Γ_j
Traffic model	Population size, M_k Arrival rate, λ_k Number of servers, S_k Departure rate, μ_k Capacity requirement, c_{kj}	

4.3.2 The state space

The global state space, $\underset{\sim}{\Omega}$, consists of a set of *system states*, $\underset{\sim}{\omega} \in \underset{\sim}{\Omega}$, defined as follows:

System state, $\underset{\sim}{\omega} = \{\omega_k\}_{k=1}^{K}$, a representation of the number of entities at any time, i.e. where $\omega_k = \#e_k$ is the number of entities of generator k, (e.g. $\#e_k$ is the number of end-to-end connections of source type k).

The number of generators is the same as the number of dimensions in the state space. This means that in a model with K generators, the state space description has K dimensions. The boundaries of the state space are determined by the relations between the generators and the resource pools, given as the routing sets, Φ, and the resource pool capacity, N_j.

4.3.3 The model dynamics

The dynamics of the model are given by the underlying simulation process that operates on $\underset{\sim}{\Omega}$ as described in section 4.2. During the simulation experiment, a sample *path* is constructed that consists of a sequence of *events*, where:

Event: an occurrence that triggers a *request* or *release* of c_{kj} resources (e.g. a connection attempt is an event that requests c_{kj} resources). A request event results in a new entity if sufficient number of resources are available for all $j \in \Phi_k$. A release event removes an entity.

Path, $\underset{\sim}{s} = \{\underset{\sim}{\omega}_0, \underset{\sim}{\omega}_1, \ldots, \underset{\sim}{\omega}_{n-1}, \underset{\sim}{\omega}_n\}$: any sequence of events, where $\underset{\sim}{\omega}_i$ is the system state after event i and n is the total number of events in path $\underset{\sim}{s}$. A regenerative cycle is a path where $(\underset{\sim}{\omega}_0 = \underset{\sim}{\omega}_n) \in \underset{\sim}{\Omega}_0$, where $\underset{\sim}{\Omega}_0$ is the regenerative state. Figure 4.3 gives an example of a sample path.

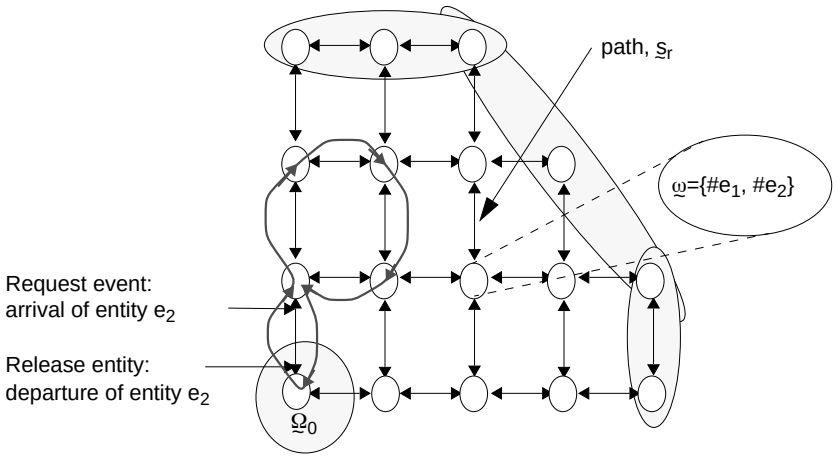


Figure 4.3 : The sequence of events constituting a sample path.

Observe that a request event not necessarily results in a change in the system state. For instance, if a new call arrives to a loss system where all trunks are occupied, the call is rejected. The call arrival is a request event. The system state holds the number of calls of each type, and it will not be updated. The request is nevertheless considered to be an *event*, because it may be a part of the statistics, e.g. counting the number of lost calls.

Generally, each event in a path causes the system state $\underset{\sim}{\omega}_i$ to be updated according to the event variable $\underset{\sim}{Z}_i$ and the corresponding event probabilities. For example, $p_{k;i}$ is the probability of a request of an event e_k.

4.3.4 The target

This chapter only considers performance measures dependent on the network constraints given by the resource pool capacities, e.g. time blocking, rerouting probability, etc. Estimation of these properties requires observations of visits to a subspace $\underset{\sim}{\Omega}_j$, denoted *target subspace* j.

Target subspace, $\underset{\sim}{\Omega}_j$: a subspace of $\underset{\sim}{\Omega}$ where the remaining capacity of resource pool j is less than the maximum capacity requested by the generators in Γ_j. This means that at least one of the generators in Γ_j is blocked:

$$\underset{\sim}{\Omega}_j = \left\{ \underset{\sim}{\omega} | \left(\sum_{k \in \Gamma_j} \omega_k c_{kj} > N_j - \max_{k \in \Gamma_j} (c_{kj}) \right) \right\}. \tag{4.5}$$

An alternative and simplified definition can be given by considering the number of resources occupied of pool j:

$$B_j = \left\{ x | \left(N_j - \left(\max_{k \in \Gamma_j} (c_{kj}) < x \leq N_j \right) \right) \right\}. \tag{4.6}$$

Observe that B_j is uniquely determined from $\underset{\sim}{\Omega}_j$, but the opposite is not possible because B_j contains no details about the entity permutations in $\underset{\sim}{\omega}$. Many combinations of entities will result in the same number of allocated resources, x, i.e. each number in B_j will have a mapping to several system states in $\underset{\sim}{\Omega}_j$.

Rare event: a visit to a target when the probability of this event is low, $P(\underset{\sim}{\Omega}_j) \ll 1$.

Single target model: is a network with only one (dominating) resource pool.

Multiple target model: is a network with several pools that have significant contributions to the property of interest.

4.3.5 State dependent transition rates

State dependent transition rates redefine the arrival and departure rates to $\lambda_k(\omega_k)$ and $\mu_k(\omega_k)$, respectively. This is a restriction of the general definition in section 4.3.1. This implies that the transition rates of a specific generator are dependent on the state of this generator only. The following functional relations apply for the arrival and departure rates:

$$\lambda_k(\underline{\omega}) = \lambda_k(\omega_k) = \begin{cases} (M_k - \omega_k)\lambda_k & M_k < \infty \\ \lambda_k & \text{otherwise} \end{cases}, \tag{4.7}$$

$$\mu_k(\underline{\omega}) = \mu_k(\omega_k) = \min(\omega_k, S_k)\mu_k . \tag{4.8}$$

The restrictions are made to simplify the adaptive algorithm in the following chapter.

Example 4.3: An M/M/1/N queue has only one generator, $K = 1$. The arrival process is Poisson, i.e. with infinite number of sources, $M_1 = \infty$, and constant arrival rates, $\lambda_1(\omega_1) = \lambda_1$. The queue have a single server, $S_1 = 1$, and exponential service times with rate $\mu_1(\omega_1) = \mu_1$, active in all states where at least one customer is present $\omega_1 > 0$.

4.3.6 Application to dependability modelling

An example of the generality of the framework is given in table 4.2. This illustrates how to model both traffic and dependability aspects of the network example from figure 4.1. For instance, observe that a link failure is modelled as a generator where each event requests the entire capacity of a link.

The concepts were originally defined for traffic models, but are later observed to be suitable also for modelling dependability aspects:

- A *link failure* is modelled as a generator where each event requests the entire capacity of a link.

- *Failure propagation* can be modelled by making the failure rate of a generator dependent on state changes in other generators. This require a more general definition of state dependent rates than was described in section 4.3.5.

- *Common mode failures*, a fault that may cause several failures is modelled by application of routing sets. All links that are included in the routing set will fail on occurrence of a single event.

- A *node failure* is modelled by defining all adjacent links as resource pools and including them in the Φ_k of the failure generator k.

- A *partial failure* or *graceful degradation* of a resource is where a failure affects parts of a link or node. This can be modelled by describing the failure process with more than two states (“ok” and “defect”), and where each event will affect some fraction of the resource pool, i.e $c_{kj} < N_j$.

As described in section 4.1, integration of users with different quality of service requirements is only possible when the model has some preemptive priority mechanism. Otherwise it is not possible to distinguish between the users. Furthermore, a preemptive priority mechanism is also required to model the failure events which are preemptive by nature. Without such a mechanism, for instance, the failure events that affect link j are not able to preempt the entities that occupy the resources of link j, and the failure will be rejected.

Alternative routing should be provided to model redundancy, e.g. to let high priority users change its routing if the primary route is overloaded or disconnected due to a link or node failure.

Table 4.2: Mapping of discrete event model concepts and simulation models, an example.

concepts	traffic simulation model	dependability simulation model
entity	connection	failure
generator	source type	link or node failure type
request event	connection arrival	failure of a link or node
release event	connection completion	repair of a link or node
capacity requirement	$1 \le c_{kj} \le N_j$	$c_{kj} = N_j$ [a]
quantity of interest	blocking probability	unavailability

a. In a partial link or node failure the capacity is reduced, not blocked, and the $c_{kj} < N_j$.

Preemptive priority and rerouting mechanisms are described as extensions to the framework in the following section.

4.3.7 Model extensions

This section includes descriptions of the extensions added to the framework to handle preemptive priority and rerouting mechanisms.

4.3.7.1 Preemptive priority

Preemptive priority implies that a request event from a generator with a high priority level are allowed to preempt an entity with a lower priority level.

- p_k is the priority level of generator k, where $p = 0$ is the highest priority level. Observe that if $p_i < p_j$ then generator i has *higher* priority and may preempt entities coming from generator j.

- $\Gamma_j^{(p)}$ is the *generator constraint set* where the generators with priority level p have resource j as a constraint.

- $\underset{\sim}{\Omega}_j^{(p)}$ is the target subspace for priority level p.

$$\underset{\sim}{\Omega}_j^{(p)} = \left\{ \underset{\sim}{\omega} \,|\, \sum_{k \in \Gamma_j^{(p)}} \omega_k c_{kj} > N_j - \max_{k \in \Gamma_j^{(p)}} (c_{kj}) \right\}. \tag{4.9}$$

In figure 4.4 it is shown how the preemptions add new transitions to the state space. With no spare capacity on arrival of a high priority event, two simultaneous updates take place:

- a sufficient number of entities with low priority is identified and disconnected, hence the corresponding #e is decreased,

- the arriving high priority entity from generator k increases $\#e_k$ by 1.

This means that arrival of an entity will in some cases affect more than one generator because low priority entities are preempted and removed from the state vector, $\underset{\sim}{\omega}$.

Example 4.4: The model in figure 4.4 has two traffic generators with different priority levels. Generator 2 has higher priority than generator 1, $p_2 = 0$ and $p_1 = 1$. Entities from generator 1 will be preempted on arrival of a new entity from generator 2 if all resources of link 1 are occupied (here $c_{1j} = c_{2j} = 1$). Let the state after event i be $\underset{\sim}{\omega}_i = \{2, 2\}$. This is a state in $\underset{\sim}{\Omega}_1^{(1)} = \{\{4, 0\}, \{3, 1\}, \{2, 2\}, \{1, 3\}, \{0, 4\}\}$, the target subspace of priority level 1, which in this example is involving generator 1 only.

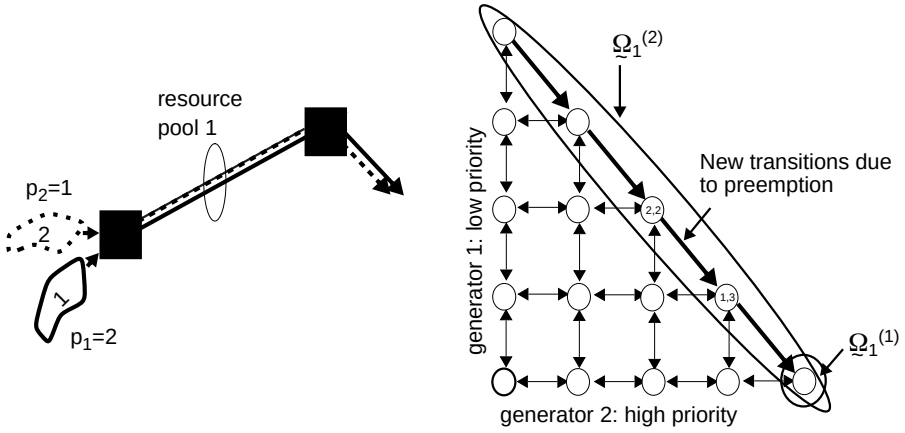


Figure 4.4 : Influence on state transitions by adding preemptive priority, an example.

The target subspace for priority level 0, involving generator 2 only, is $\Omega_1^{(0)} = \{\{0, 4\}\}$. The event variable for ω_i is:

$$Z_i = \begin{cases} \{-1, 0\} & q_{1;i} \\ \{0, 0\} & 1 - (q_{1;i} + q_{2;i} + p_{2;i}) \\ \{0, -1\} & q_{2;i} \\ \{-1, 1\} & p_{2;i} \end{cases} \tag{4.10}$$

This means that when an entity from generator 2 arrives in state $\omega_i = \{2, 2\}$, one of the two entities of generator 1 must be preempted. Thus, the state is changed from $\omega_i = \{2, 2\}$ to $\omega_{i+1} = \{1, 3\}$.

4.3.7.2 Rerouting

Rerouting sets are defined to allow the generator k to allocate resources from an alternative set of resources.

- R_k is the number of rerouting alternatives for generator k.
- $\Phi_k = \{\Phi_{k0}, \ldots, \Phi_{kR_k}\}$ is the set of alternative routes for generator k.

- $\underset{\sim}{\omega} = \{\{\omega_{kr}\}_{r=0}^{R_k}\}_{k=1}^{K}$ is the extended system state, where $\omega_{kr} = \#e_{kr}$ is the number of entities of generator k that follows the r th route, $(r = 0, \ldots, R_k)$.

- $\underset{\sim}{\Omega}_j$ is the target subspace of j:

$$\underset{\sim}{\Omega}_j = \left\{ \underset{\sim}{\omega} \mid \sum_{k=1}^{K} \sum_{r=0}^{R_k} \sum_{j \in \Phi_{kr}} \omega_{kr} c_{kj} \geq N_j \right\}. \tag{4.11}$$

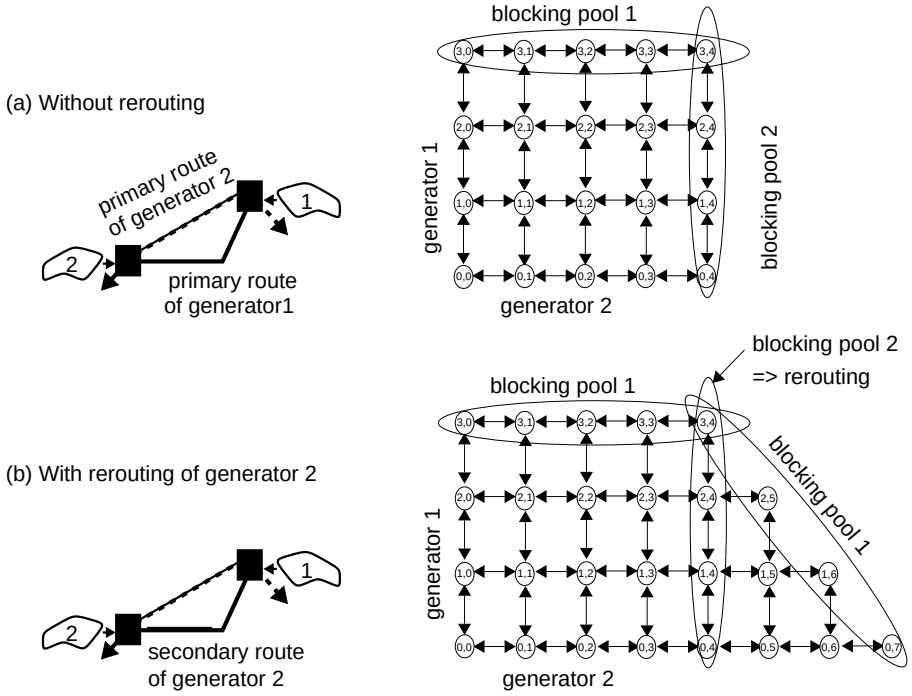


Figure 4.5 : Influence on state space limitations by adding rerouting, an example.

A generator switches to an alternative route only if the primary route $r = 0$ is not available. The alternative routes are checked in sequence from $r = 1$ up to R_k until a route with sufficient capacity on all links is found. If no route is available, a blocking has occurred, i.e. a visit to one, or several, target subspaces, $\underset{\sim}{\Omega}_j$, is observed.

Example 4.5: Consider a network with 2 nodes and 2 links. Traffic is offered by two user types, modelled by generator 1 and 2. The primary route for generator 1 is link 1, $\Phi_{10} = \{1\}$, and for generator 2 it is link 2, $\Phi_{20} = \{2\}$. With no rerouting, the situ-

ation is as described in figure 4.5(a). The generators have independent resource constraints. An overlap between the two target subspaces is only present when the total capacity both links are independently, and simultaneously occupied. The target subspace of the two links are:

$$\begin{aligned} \Omega_1 &= \{\{3,0\},\{3,1\},\{3,2\},\{3,3\},\{3,4\}\} \\ \Omega_2 &= \{\{0,4\},\{1,4\},\{2,4\},\{3,4\}\} \end{aligned} \tag{4.12}$$

Now, let generator 2 have an alternative route on link 1, $\Phi_{21} = \{1\}$. The two generators compete for the capacity on link 1 when link 2 is not available. The situation is described in figure 4.5(b), showing that the state space is extended. The new states are added to describe the situation when entities from generator 2 occupy resources on link 1. It is assumed that generator 2 will not preempt entities from generator 1 when an alternative route is chosen. The target subspaces with rerouting are:

$$\begin{aligned} \Omega_1 &= \{\{3,0\},\{3,1\},\{3,2\},\{3,3\},\{3,4\},\{2,5\},\{1,6\},\{0,7\}\} \\ \Omega_2 &= \varnothing \end{aligned} \tag{4.13}$$

In this case, by definition, lost traffic generator 2 is not due to blocking on link 2 only.

4.4 Closing comments

This chapter introduces the basic building blocks of a flexible framework for the modelling of communication networks. The concepts of generators and resource pools are introduced. A generator can be interpreted as, for instance, a collection of users with the same attributes, while a resource pool may be considered as communication channels on a link. These concepts were originally defined for traffic models, but is later also used for modelling of dependability aspects like link or node failure.

The communication network is assumed to handle a mixture of traffic stemming from users with very different resource capacity and quality of service requirements. To be able to distinguish between users, and to provide different quality of service to them, a preemptive priority mechanism is introduced. Furthermore, to increase the offered quality of service, a mechanism for switching between alternate routes is provided to the users. This means that if the network topology has link disjoint routes between the end nodes, a user may switch to an alternate route if the primary route is occupied or disconnected. The description of rerouting given in this chapter, does not combine this with preemptive pri-

orities. This is done to clarify the presentation. The framework is defined to let a high priority generator preempt a low priority entity when conflicts occur after a rerouting. It is no conceptual difference between arrival of a rerouted entity and an entity following the primary route. The preemptive priority and rerouting mechanisms are also crucial for modelling of link or node failures.

The simulation process that adds the dynamics to the model, is assumed to be a *continuous time Markov chain, CTMC*. A *discrete time Markov chain, DTMC* is embedded on this process. All simulations of *steady state* properties can use the discrete process with the expected state sojourn times instead of taking samples from a distribution. Simulations of transient properties require that the continuous process is used.

The modelling framework may use more general processes than the CTMC, for instance a *generalised semi Markov process, GSMP*, as described in [Gly89]. In this process, the events take place at embedded points which are not necessarily renewal points. The event variables $\underset{\sim}{Z}_n$ from (4.3) are far more complicated. Alternatively, defining a DTMC process embedded on all events of the GSMP, and not only on the renewals, the importance sampling strategy will be far more complicated, see e.g [GSHG92].

A model of a real-sized, well-engineered network, is large and multidimensional. The model is not easily reduced when the resource utilisation is balanced, i.e. with no significant bottleneck. This is also a problem with respect to definition of the parameters for importance sampling simulation. A rare event of interest may occur in several links in the network. In the following chapter, this is discussed in further detail and a new adaptive strategy for changing the parameters is proposed.

5

Adaptive parameter biasing in importance sampling

This chapter presents the adaptive change of measure that applies to the models described in chapter 4. As will be addressed in section 5.1, these models pose new challenges to the change of measure, or parameter basing, in importance sampling. Hence, previous proposed approaches do not longer suffice. The general ideas of the adaptive biasing are presented in section 5.2, and section 5.3 adds details. In section 5.4, the robustness of the adaptive biasing is commented. The idea was first presented in [Hee95b], and later improved in [Hee96] and further generalised in [Hee97a]. The description given in this chapter can also be found in [Hee98]. For applicability to networks, see the feasibility studies in chapter 6.

5.1 The challenge of parameter biasing in network models

The variance of the importance sampling estimate is given in (3.3) in chapter 3. There it was pointed out that the optimal change of measure is the $f^*(\underline{s})$ that minimises this variance, $Var(\hat{\gamma}_{IS})$. The variance is 0 when $g(\underline{s})f(\underline{s}) = \gamma f^*(\underline{s})$. Unfortunately, this involves the unknown property of interest, γ. This minimum variance restriction, however, serves as a general guideline implying:

$$g(\underline{s})f(\underline{s}) \propto f^*(\underline{s}). \tag{5.1}$$

This means that the new distribution $f^*(\underline{s})$ should be proportional to the importance of sample $\underline{s}$ to minimise the variance of $\hat{\gamma}_{IS}$. The importance of the sample $\underline{s}$ is the product of the original sampling distribution, $f(\underline{s})$, and the contribution, $g(\underline{s})$.

Now, consider a multidimensional network model involving J resource pools. The property of interest is associated with these pools. In the model a *target subspace* $\underline{\Omega}_j \subset \underline{\Omega}$ is defined for each pool in which the rare events, that contributes to the property of interest,

occur. Each sampled path $\underline{s}$ has J contributions, $g_j(\underline{s})$, one for each target subspace. Returning to the guideline of (5.1), it is heuristically assumed that the optimal change of measure should be proportional to the importance of each of these J targets:

$$g_j(\underline{s})\, f(\underline{s}) \propto f_j^*(\underline{s}). \tag{5.2}$$

This is only possible to achieve when the system model has one of the following two properties:

1. *A single dominating target*. This means that there exists a target j_{max} where, for all (at least the most likely) sample paths, $\underline{s}$: $g_{j_{\max}}(\underline{s}) > g_j(\underline{s})$, $\forall j \neq j_{max}$. Then the optimal change of measure can be approximated by using the optimal change of measure with respect to the target j_{max}. This approach has been used, for instance, in tandem queues [PW89] and loss networks [Man96c].

2. *Single visit paths.* When a sample path never includes visits to more than one target, i.e.

   $$g_j(\underline{s}) \cdot g_i(\underline{s}) = 0, \forall j \neq i, \forall \underline{s} \tag{5.3}$$

   then the R replications of a simulation experiment can be divided into J independent stratas, see [Fra93]. Each strata contains $R_j = p_j \cdot R$ replications where the change of measure is relative to target j. The optimal strata probability is $p_j = G^{-1} \cdot \sum_{\forall \underline{s}} g_j(\underline{s})\, f(\underline{s})$ where $G = \sum_{j=1}^{J} \sum_{\forall \underline{s}} g_j(\underline{s})\, f(\underline{s})$. This is computationally demanding in a multidimensional model.

In a more realistic system, the target subspaces will be overlapping and the single visit condition from (5.3) will no longer be satisfied. Now, using an approach based on (5.3), the estimates of the system performance are biased because $\sum_{j=1}^{J} g_j(\underline{s})\, f(\underline{s}) > g(\underline{s})\, f(\underline{s})$. An alternative approach, see section 3.4.2.2, is reversion of the drift including *all* K generators, irrespective of the target subspaces in the model. This increases the positive drift towards all target subspaces simultaneously, at least the drift towards the regeneration subspace, Ω_0. This approach has shown poor performance through a series of simulation experiments. It seems like with no indication of “best direction” to change the drift of the simulation process, the sampled path tends to follow an unlikely path to a rare event.

Example 5.1: (continued from example 4.2). To demonstrate why the two approximations above will fail in a balanced network, consider evaluation of the model of figure 4.2. The property of interest is the probability that either of the two generators

is blocked. The blocking is due to insufficient capacity on link (resource pool) 2, 3 or 10. Let the total blocking be γ and the blocking of pool j be γ_j, $j = 2, 3, 10$. Now, make the following assumptions about the exact blocking probabilities, $\gamma = 10^{-7}$, $\gamma_2 = \gamma_{10} = 5 \cdot 10^{-8}$, and $\gamma_3 = 6 \cdot 10^{-8}$. Observe that $\gamma \neq \gamma_2 + \gamma_3 + \gamma_{10}$. First, let the importance sampling simulation change the drift towards pool 3 because this has the largest blocking probability, $\gamma_3 = \max(\gamma_2, \gamma_3, \gamma_{10})$. Then, an accurate estimate $\hat{\gamma}_3$ of the blocking on pool 3 is obtained. However, the result of this simulation is that the overall blocking estimate is $\hat{\gamma} \approx \hat{\gamma}_3$ because only visits to Ω_3 are provoked. Instead, importance sampling combined with stratified sampling is applied as described in approach 2. Good and accurate estimates of $\hat{\gamma}_2$, $\hat{\gamma}_3$, and $\hat{\gamma}_{10}$ are likely to be obtained, but the $\hat{\gamma} = \hat{\gamma}_2 + \hat{\gamma}_3 + \hat{\gamma}_{10}$ estimate is unbiased because the condition in (5.3) is violated.

A well engineered network has a balanced utilisation of the resources in the system. This means that the overall property of interest, e.g. the call blocking, will in practice be determined by restrictions imposed by more than one resource pool. Furthermore, the resource pools are shared between several generators, as defined in the routing set Φ_k. The assumption from (5.3) is no longer valid. To overcome this limitation, a new parameter biasing strategy is proposed that guides the simulation process to focus on the *most important targets* at *every step* along a simulated path. This will be described in the following section.

5.2 General idea of the adaptive parameter biasing

The general idea of the biasing is to adapt the change of measure to every state along the simulated path $\underset{\sim}{s}$, and to induce a positive drift towards the most important target seen from the current state, $\underset{\sim}{\omega}$.

The *importance* of a target j is generally defined as:

$$H_j = \sum_{\forall \underset{\sim}{s}} g_j(\underset{\sim}{s}) f(\underset{\sim}{s}) = E_f(\gamma_j). \tag{5.4}$$

This can be interpreted as the expected contribution γ_j from target j to the property of interest, γ. Observe that in a realistic system the $E_f(\gamma) \neq \sum_{j=1}^{J} E_f(\gamma_j)$, because the single target visit condition from (5.3) is violated, see example 5.1. The importance of target j, given that a specific state $\underset{\sim}{\omega}$ is visited in $\underset{\sim}{s}$, is defined as

$$H_j(\omega) = \sum_{\forall \xi \wedge (\omega \in \xi)} g_j(\xi) f(\xi). \tag{5.5}$$

The ideas are implemented in the following steps:

1. Estimate the *importance* $\hat{H}_j(\omega)$ for all targets j at current state ω. An estimate of this importance is presented in section 5.3.2. The relative importance is denoted the *target distribution*, and estimated by $\hat{p}_j(\omega) = \hat{H}_j(\omega) / [\sum_{j=1}^{J} \hat{H}_j(\omega)]$.

2. Sample a tentative target from the estimated target distribution, $\hat{p}_j(\omega)$, ($j = 1, \ldots, J$).

3. Change the measure by changing the rates of the generators $k \in \Gamma_j$ as was described in (3.25), according to the tentative target j that was sampled in the previous step.

4. Sample the next event in the simulation process, $\underset{\sim}{X}_i$, using the biased parameters from step 3.

Example 5.2: (continued from example 5.1). After a few events in the simulated path, the current state is $\omega = \{1, 1\}$. The positive drifts towards the target subspaces under the original sampling distribution are

$$\begin{aligned} \delta_{2+}(\omega) &= c_{12} \cdot \lambda_1(1) \\ \delta_{3+}(\omega) &= c_{23} \cdot \lambda_2(1) \\ \delta_{10+}(\omega) &= c_{1\,10} \cdot \lambda_1(1) + c_{2\,10} \cdot \lambda_2(1) \end{aligned} \tag{5.6}$$

The importance of each target is estimated to be[1]:

$$\begin{aligned} \hat{H}_2(\omega) &= 2.5 \\ \hat{H}_3(\omega) &= 4.0 \\ \hat{H}_{10}(\omega) &= 3.5 \end{aligned} \tag{5.7}$$

which results in the following target distribution:

$$\begin{aligned} \hat{p}_2(\omega) &= 0.25 \\ \hat{p}_3(\omega) &= 0.40 \\ \hat{p}_{10}(\omega) &= 0.35 \end{aligned} \tag{5.8}$$

1. All numerical values are freely chosen to demonstrate the point of a balanced network, not a result of estimations using a set of specific parameters, although it could have been.

This means that at state $\omega = \{1, 1\}$, the change of measure is, with probability 0.40, reversion of the drift towards pool 1. The BIAS factor scales the arrival and departure rates as described in (3.25), and is $\text{BIAS}_3(\omega) = \delta_{3\text{-}}(\omega)/\delta_{3+}(\omega) = \mu_2(1)/\lambda_2(1)$. With probability 0.25, the drift is reversed towards pool 2, and with probability 0.35 towards pool 10. In the latter case, the arrival and departure rates of both generators are scaled. This implies that the drift toward pool 2 and 3 is also increased, in addition to pool 10. However, the BIAS factor is not the same as $\text{BIAS}_2(\omega)$ and $\text{BIAS}_3(\omega)$:

$$\text{BIAS}_{10}(\omega) = \frac{\delta_{10\text{-}}(\omega)}{\delta_{10+}(\omega)} = \frac{c_{1\ 10} \cdot \mu_1(1) + c_{2\ 10} \cdot \mu_2(1)}{c_{1\ 10} \cdot \lambda_1(1) + c_{2\ 10} \cdot \lambda_2(1)}. \tag{5.9}$$

In [Car91], an idea similar to the adaptive biasing was presented, denoted *failure distance biasing*. In this biasing strategy, the *importance*, H_j, is the minimum number of transitions from the current state up to a system failure mode j (which is the target). This is, in general, not a very precise estimate of (5.4). A more computational demanding change of measure is applied in failure distance biasing than in (3.25).

It is important to emphasise that the target distribution obtained in step 1 above, is only used in step 2 to decide which target should be in focus for the next change of measure in step 3. This *implicit* influence on the simulation process dynamics is assumed to make the simulation efficiency less sensitive to the accuracy of the target distribution estimate. If, instead, the $\hat{p}_j(\omega)$ had been used as an estimate of the event probabilities from section 3.2 directly, it is expected that the simulation efficiency would have been far more sensitive to the accuracy. In section 5.4, a few comments are made on the robustness and weakness of the target distribution estimate, and the effect on the simulation efficiency.

The computation of $\hat{p}_j(\omega)$ will significantly influence the total computational overhead added to the simulations by the adaptive strategy. The reason is that the estimation must be repeated for *every step along the simulated path*, because the relative importance will change as the system state changes.

Example 5.3: To see why the importance changes, consider the example in figure 5.1. This shows a state space consisting of 3 target sub-spaces. The sub-spaces are made disjoint to simplify the illustration. The contour lines represent the “iso-likelihoods” of each target, i.e. every state at a specific contour line has the same conditional target likelihood. The contour intervals are logarithmic with base 10, i.e. the iso-likelihoods represent probabilities at every order of magnitude, i.e. 10^{-1}, 10^{-2}, 10^{-3}, … .

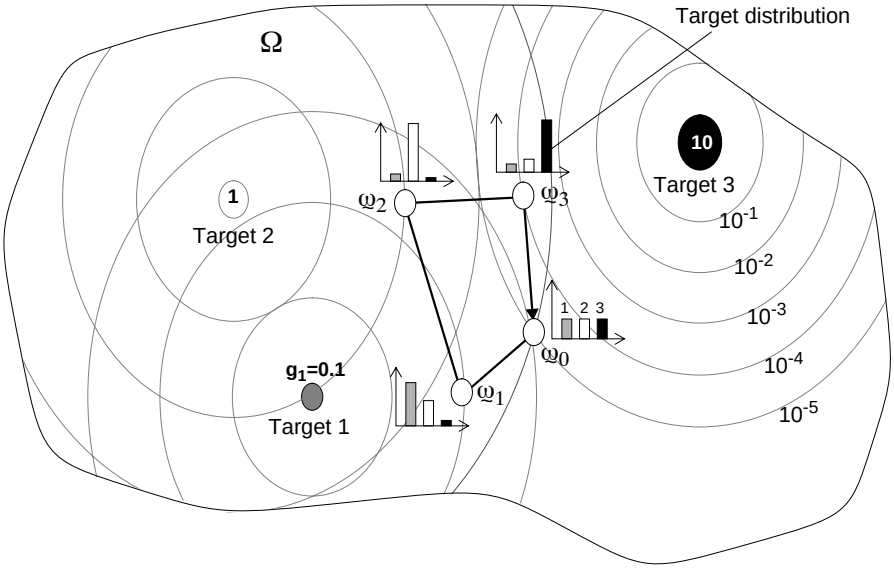


Figure 5.1 : Changes in target distribution illustrated by iso-likelihood curves along an unlikely path in the subspace.

The sampled path depicted in the figure, $\underline{s} = \{\omega_0, \omega_1, \omega_2, \omega_3\}$, demonstrates how the relative importance changes. At ω_0 all targets have the same importance. This means that the next target to be in focus will be chosen from a uniform distribution, see the histogram next to the system state ω_0. Stepping to ω_1, the situation is changed. Now the process has approached target 1, and the target distribution reflects this by increasing the probability of target 1 relative to the targets 2 and 3. This means that the most likely sampling distribution is now optimal with respect to target 1, $f_1^*(\underline{s})$. Correspondingly, at state ω_2 and ω_3, the most likely targets are 2 and 3, respectively. The change of measure at state ω can be expressed as:

$$f^*(\underline{s}) = \begin{cases} f_1^*(\underline{s}) & \text{with probability } \hat{p}_1(\omega) \\ f_2^*(\underline{s}) & \text{with probability } \hat{p}_2(\omega) \\ f_3^*(\underline{s}) & \text{with probability } \hat{p}_3(\omega) \end{cases} \tag{5.10}$$

Even if target 1 is the target with the greatest importance at state $\underset{\sim}{\omega}_1$, it is possible that the sampling from the target distribution results in a change of measure with a drift towards one of the two other targets. Furthermore, even with a simulation process that has an expected positive drift towards target 1, the next event in the sampled path may move the process towards target 2 or 3, or even back to the regenerative state. The path in figure 5.1 is, in fact, an example of a very unlikely path where the simulation process in neither of the states moves in the direction of the most important target.

It is not a trivial matter to determine the target importance and the corresponding target distribution. In general, the conditional iso-likelihoods do not have the nice circular contour lines as the example in figure 5.1. The following section describes an estimate of the target distribution, $p_j(\omega)$.

5.3 Target distribution

Several algorithms and approximations to the target distribution, $p_j(\omega)$, have been proposed since the adaptive biasing was first introduced, see [Hee95b, Hee96, Hee97a]. Common for all these, is that the target distribution is the *relative target importance*, i.e.

$$p_j(\underset{\sim}{\omega}) = H_j(\underset{\sim}{\omega}) / [\sum_{j=1}^{J} H_j(\underset{\sim}{\omega})]. \tag{5.11}$$

The target importance $H_j(\underset{\sim}{\omega})$ is given in (5.5). An approximation of the target distribution, $p_j(\omega)$, must be:

1. sufficiently close to the exact target distribution of (5.11),

2. *robust* to changes in parameters and state space structures, and

3. computationally *efficient*.

Observe that the estimate of the target distribution depends on the estimates of the *relative*, and not the absolute importance. This can, under some conditions, be exploited to significantly reduce the computational effort, as described in the following.

5.3.1 Simplified estimate of target importance

The importance of a target consists of an infinite number of contributions from sample paths, $\underset{\sim}{s}$, as given in (5.5). Not all samples have (significant) contributions to target j.

Hence, an approximation of the importance is proposed where only the n largest contributions are included:

$$H_j^{(n)}(\underset{\sim}{\omega}) = \sum_{i=1}^{n} g_j(\underset{\sim}{s}_j^{(i)}) f(\underset{\sim}{s}_j^{(i)}) \leq H_j(\underset{\sim}{\omega}). \tag{5.12}$$

$\underset{\sim}{s}_j^{(i)}$ is the sample path $\underset{\sim}{s}$ with the ith largest contribution to the importance of target j. Observe that the approximation in (5.12) is systematically underestimating the exact value. But, recall that the target distribution depends on good estimates of the *relative*, and not the *absolute,* importance. The simplest approximation is then to let $n = 1$, which means that only the *path with the largest importance,* $\underset{\sim}{s}_j^{(1)}$, is included. The target distribution is then approximated by:

$$\tilde{p}_j(\underset{\sim}{\omega}) \approx H_j^{(1)}(\underset{\sim}{\omega}) / [\sum_{j=1}^{J} H_j^{(1)}(\underset{\sim}{\omega})]. \tag{5.13}$$

To obtain an estimate of the target importance in (5.13), it is required to identify the most important sample path, $\underset{\sim}{s}_j^{(1)}$, which includes both the current state $\underset{\sim}{\omega}$, and at least one visit to the target subspace $\underset{\sim}{\Omega}_j$. A *subpath* of the *complete* path, $\underset{\sim}{s}$, as was defined in section 4.3.2, will be identified. This subpath includes only the sequence of events from $\underset{\sim}{\omega}$ to the first entrance to the target subspace $\underset{\sim}{\Omega}_j$. A large number of sample paths has to be considered to identify this subpath, because the *importance* of a path consists of the product of both the *likelihood* and *contribution*. Instead of including both the likelihood and the contribution simultaneously, the following estimate is proposed:

$$\hat{H}_j^{(1)}(\underset{\sim}{\omega}) = \hat{f}_j(\underset{\sim}{\omega}) \cdot \hat{g}_j(\underset{\sim}{\omega}) \tag{5.14}$$

where the *likelihood,* $\hat{f}_j(\underset{\sim}{\omega})$, and the *contribution,* $\hat{g}_j(\underset{\sim}{\omega})$, are estimated separately. In section 5.3.2, it is described how the target likelihood is established. The contribution associated with this subpath is briefly presented in section 5.3.3.

The estimation of the target importance are in the following presented for a single target only. It must be repeated for all $j = 1, \ldots, J$ to provide an estimate of the target distribution, and finally substituting (5.14) for each target j in the estimate of the target distribution in (5.13).

5.3.2 Target likelihood

To provide an estimate of the *target likelihood*, $\hat{f}_j(\underset{\sim}{\omega})$ in (5.14), a subpath from the current state $\underset{\sim}{\omega}$ up to a the first entrance to the target subspace $\underset{\sim}{\Omega}_j$ must be identified. Define this subpath, consisting of n_j events:

$$\underset{\sim}{\sigma}_j(\underset{\sim}{\omega}) = \{\underset{\sim}{\omega}_i | (\underset{\sim}{\omega}_0 = \underset{\sim}{\omega}) \wedge (\underset{\sim}{\omega}_{n_j} \in \underset{\sim}{\Omega}_j) \wedge (\underset{\sim}{\omega}_i \notin \underset{\sim}{\Omega}_j, i < n_j)\}. \tag{5.15}$$

Event i is generated by $k_i \in \Gamma_j$, i.e. one of the generators constrained by resource pool j. Every *event* in $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ is assumed to be an *arrival* of an entity. No release events are included because $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ should represent the *most likely* sequence of events to the target. An *entity* from generator k_i is allocating $c_{k_i j}$ resources. The state $\underset{\sim}{\omega}_i$ after event i in $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ is given by the following recursion:

$$\underset{\sim}{\omega}_i = \underset{\sim}{\omega}_{i-1} + \underset{\sim}{1}_{k_i} = \underset{\sim}{\omega} + \sum_{l=1}^{i} \underset{\sim}{1}_{k_l} \tag{5.16}$$

where $\underset{\sim}{1}_k = \{0, 0, \ldots, 1, \ldots, 0\}$ is an *index vector* of size K where element k is 1, and all other elements are 0. Observe from (5.16) that the subpath is uniquely determined by the current state $\underset{\sim}{\omega}$ and the sequence given by $\underset{\sim}{k} = \{k_1, k_2, \ldots\}$.

The events in the subpath $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ are generating a sequence of resource allocations. Let x_j denote the number of resources from pool j allocated at state $\underset{\sim}{\omega}$. The number of resources at the state $\underset{\sim}{\omega}_i$ after event i, is $(x_j)_i$. The $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ can be described as a sequence of $(x_j)_i$'s instead of, or in addition to, the $\underset{\sim}{\omega}_i$ from (5.16). Since the $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ contains only arrivals, then $(x_j)_i > (x_j)_{i-1}$ for all $i = 1, \ldots, n_j$. To simplify the following description, let the index j on x_j be omitted. Then x_j becomes x, and $(x_j)_i$ becomes x_i. This is unambiguous because in this section only the subpath to target j is described.

Recall from section 4.3.4, the target subspace was defined in terms of the system states on the state space $\underset{\sim}{\Omega}_j$, but also in a simpler form in terms of the number resources, x:

$$B_j = \left\{ x | (N_j - \max_{k \in \Gamma_j} c_{kj}) < x \leq N_j \right\}. \tag{5.17}$$

This is derived from the $\underset{\sim}{\Omega}_j$ definition in section 4.3.4. For the purpose of searching for the most likely subpath, it is convenient to use this description of the target subspace. The sequence of resource allocations in a specific subpath, starting from x_0, is:

$$x_i = x_0 + \sum_{l=1}^{i} c_{k_l j}. \tag{5.18}$$

The number of resource allocations of the last state in the subpath is $x_{n_j} \in B_j$.

In appendix E an efficient search algorithm is described where the *maximum likelihood* of a subpath from current state $\underset{\sim}{\omega}$ up to a state with x resources. The maximum likelihood is denoted π_x, and is obtained for every $x_0 < x \leq N_j$. The algorithm exploits the Markov properties of the simulation process to successively determine the π_x by reuse of previous values, $\pi_{x - c_{kj}}$. As input, the search algorithm requires:

- the current state, $\underset{\sim}{\omega}$,
- the generators which are constrained by pool j, $k \in \Gamma_j$,
- the target subspace, either given as $\underset{\sim}{\Omega}_j$ or B_j.

The algorithm also returns the specific path $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ that is associated with each π_x, $(x = x_0 + 1, \ldots, N_j)$.

Assume that a set of π_x is provided by the search algorithm. Then, the target likelihood will be assigned to any of the π_x where $x \in B_j$. This represents the likelihood of a subpath from current state up to a state where the number of allocated resources x is in the target subspace.

The *maximum likelihood* is:

$$\hat{f}_j(\underset{\sim}{\omega}) = \max_{x \in B_j} (\pi_x) \tag{5.19}$$

while, the *maximum importance* is estimated by taking the contribution, given by the end state of the subpath, into account (see section 5.3.3 for estimation of target contribution):

$$\hat{H}_j^{(1)}(\underset{\sim}{\omega}) = \max_{x \in B_j} (\pi_x \cdot \hat{g}_j(\underset{\sim}{\omega}^{(x)})). \tag{5.20}$$

The estimate from (5.20) is substituted in (5.13) to determine the target distribution.

Introducing the rerouting and preemptive priority mechanisms will only affect the input arguments to the search algorithm. The following two sections briefly describe what the arguments should be.

5.3.2.1 Target likelihood with rerouting

The mechanism for alternative routing was described in section 4.3.6. The routing sets Φ_k, were extended. The Γ_j set depends on the current state through the given routing:

$$\Gamma_j = \{k | j \in \Phi_{kr_k}\} \tag{5.21}$$

where r_k $(0 \le r_k \le R_k)$ is the index of the route used by generator k in state ω.

The current number of allocated resources from pool j, x_0, is changed according to the extended system state description in section 4.3.6:

$$x_0 = \sum_{k=1}^{K} \sum_{r=0}^{R_k} \sum_{j \in \Phi_{kr}} \omega_{kr} c_{kj}. \tag{5.22}$$

Let the Γ_j in (5.21), the ω as defined in section 4.3.6, and the x_0 from (5.22) be the input arguments to the search algorithm. This will provide the requested set of π_x for $x_0 < x \le N_j$. The target subspace B_j is defined by using (5.21) in (5.17), and the target likelihood and importance are determined by (5.19) and (5.20), respectively.

This approach will provoke the rare events that cause blocking on the current route of a generator only, even if the generator has several alternative routes that all have to be blocked for the generator to be blocked. This can be justified by the following example.

Example 5.4: Consider a generator k with a secondary route, $R_k = 1$. At the start of a regenerative cycle, the simulation parameters are biased to provoke a blocking in the primary route, Φ_{k0}. When this route is blocked, generator k will route new calls via the secondary route, Φ_{k1}. Now, the rare events in the targets $j \in \Phi_{k1}$ will be provoked. Since the generator is not able to use the secondary route before the primary is blocked, it makes no sense to provoke rare events in the secondary route when the primary route is not blocked.

5.3.2.2 Target likelihood with preemptive priority

In a model with several preemptive priority levels, the search algorithm is applied to determine a subpath for each priority level, $\sigma_j^{(p)}$. Each subpath includes the generators in Γ_j that have a specific priority level p, or higher. Although it is the likelihood of the subpath of priority level p that shall be determined, the generators with higher priority than p have

to be included because they cannot be preempted like the low priority generators have been. Let $\Gamma_j^{(p)} \subseteq \Gamma_j$ be the set of generators which has priority p, or higher:

$$\Gamma_j^{(p)} = \{k | (p_k \leq p) \wedge j \in \Phi_k\}. \tag{5.23}$$

The corresponding initial resource allocations are:

$$x_0^{(p)} = \sum_{k \in \Gamma_j^{(p)}} \omega_k c_{kj}. \tag{5.24}$$

Since the subpath $\sigma_j^{(p)}$ is associated with priority level p, the target subspace must be defined as the subspace where generators of priority level p is blocked:

$$B_j^{(p)} = \left\{ x | \left(N_j - \max_{k \in (\Gamma_j^{(p)} - \Gamma_j^{(p-1)})} c_{kj} \right) < x \leq N_j \right\} \tag{5.25}$$

where $\Gamma_j^{(p)} - \Gamma_j^{(p-1)} = \{k | (p_k = p) \wedge j \in \Phi_k\}$ is the set of generators with priority level p.

Now, as input to the search algorithm, use $\Gamma_j^{(p)}$ from (5.23), and $x_0^{(p)}$ from (5.24). The algorithm will then provide a set of $\pi_x^{(p)}$ for $x_0^{(p)} < x \leq N_j$. It is not necessary to include some sophisticated preemption priority mechanism into the search algorithm because the lower priority generators are already removed from $\Gamma_j^{(p)}$. The target likelihood, $\hat{f}_j^{(p)}$, and the importance, $\hat{H}_j^{(p)}$, for priority level p, are obtained by substituting π_x by $\pi_x^{(p)}$ in (5.19) and (5.20), respectively.

The final target likelihood and importance are determined by taking the maximum likelihood and importance over the different priority levels:

$$\hat{f}_j(\underline{\omega}) = \max_{\forall p}(\hat{f}_j^{(p)}), \tag{5.26}$$

$$\hat{H}_j^{(1)}(\underline{\omega}) = \max_{\forall p}(\hat{H}_j^{(p)}). \tag{5.27}$$

5.3.3 Target contribution

The target *contribution* is the property of interest, $g(\underset{\sim}{s})$, observed by using the subpath $\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})$ instead of a complete path $\underset{\sim}{s}$. For instance, if the steady state probability of target j is the property of interest, then

$$\hat{g}_j(\underset{\sim}{\omega}) = g(\underset{\sim}{\sigma}_j(\underset{\sim}{\omega})) = \left(\sum_{k=1}^{K} \lambda_k(\underset{\sim}{\omega}_{n_j}) + \mu_k(\underset{\sim}{\omega}_{n_j})\right)^{-1} \tag{5.28}$$

is the contribution, i.e. the expected *sojourn time* in the first visited state in the *target subspace.*

5.4 Closing comments

This chapter presents the ideas of an adaptive parameter biasing for importance sampling. The goal is to change the drift of the simulation process. The process should move towards the parts of the state space where the most important rare events occur. These subspaces are denoted target subspaces. A measure is provided to guide the process to change the drift towards the most important of these target subspace. However, this importance will not be the same in every state of the state space. This means that some measure of the target importance must be developed that adaptively adjusts to the current state of the simulation process. The ideas, and an implementation of these, are described in this chapter.

The requested measure of target importance is denoted the *target distribution*. Generally, it is not feasible to obtain the exact distribution, at least it cannot be done efficiently. Thus, a rough, but efficient, approximation of the target distribution is proposed. This target distribution will demonstrate its feasibility on several network examples in the following chapter.

The estimated target distribution has changed several times since the adaptive biasing was first introduced. Even the simulation results presented in the first approach produced accurate estimates in reasonable time. This is an indication of that the adaptive biasing is rather robust with respect to the accuracy of the target distribution estimate.

The target distribution changes for every event in the simulated sample path. However, the change is not significant for every event so the same $p_j(\underset{\sim}{\omega})$ is applicable for several events. In the network simulations in the following chapter, the distribution is only recalculated

when the accumulated sample path probability, see (3.14), has changed more than one order of magnitude.

Some additional work can be done on the adaptive biasing. The target distribution can be further tested on new models to get increased insight in what the strengths and weaknesses are. It is also interesting to make a computational profile of the efficiency of each step in the adaptive strategy for new models. This to see which parts of the approach that is the most computer demanding, and to determine the total computational overhead introduced by the adaptive biasing. See example 6.2 for a simulation profile from a network system.

6

Importance sampling in network simulations

This chapter contains the results of several network simulations. The purpose of these experiments is to demonstrate the feasibility of the adaptive parameter biasing proposed in the previous chapter. In section 6.1 an overview of the simulation experiments and their objectives is given. The regenerative simulation setup applied to the network simulations are presented in section 6.2. The simulation results and observations are given in sections 6.3-6.5, while section 6.6 summarises the experiments and proposes new experiments.

6.1 Simulation objectives

It is important to keep in mind that the main purpose of the simulation experiments in this chapter is to demonstrate the feasibility of the adaptive technique, rather than to provide insight in the properties of a specific communication network. The examples are all fictitious networks which are described by the modelling framework of chapter 4. Both the preemptive priority and the rerouting mechanisms are included in the examples.

The objective is to demonstrate the feasibility of the proposed adaptive strategy. For this purpose, the following 3 network examples are constructed:

- *Case 1: No priority nor alternative routing.*
  When all the generators have the same priority and have a fixed routing, it is feasible to obtain exact blocking probabilities, e.g. by the *convolution method* [Ive87], when the model is of moderate size. Hence, the simulation results from this case can be compared with exact values. The efficiency and precision of the simulation method are demonstrated.

- *Case2: Improving the quality of service by rerouting.*
  This case demonstrates the use of rerouting. Two simulation series are produced, one with, and one without, an alternative (secondary) route. The improvement of the quality of service in terms of reduced call blocking is estimated.

- *Case 3: Disturbing low priority traffic.*
  To demonstrate the use of the preemptive priority mechanism, low priority traffic is added to the case 2. Three simulation series are produced where the blocking probabilities of low priority traffic are estimated. First the network is simulated with only low priority traffic generators, then mixed with high priority traffic, and finally mixed with both high priority traffic and affected by link failures.

In all three cases, the quantities of interest are the blocking probabilities of all, or a selection of, user types, i.e. the generators. No exact solutions are analytically provided for the models in case 2 and 3. Instead, the rough blocking approximation from appendix F is applied for comparisons. The approximation assumes individual blocking on each link which is expected to result in an upper bound of the blocking probabilities. In addition, the importance sampling estimates are compared to the estimates obtained from the direct simulations given in appendix G.

The results from the experiments are found in sections 6.3-6.5.

6.2 Regenerative simulation of large models

When conducting a simulation experiment with importance sampling it is recommended to divide the experiment into independent regenerative cycles for the stability of results. A regenerative state, or subspace, must be defined. For dependability models and very small traffic models, this is normally a single state like the state “system intact” or “empty system”. However, for real-sized traffic models, choosing a single regenerative state will make the expected regenerative cycle period too long.

This chapter uses a *regenerative box* of K dimensions, denoted Ω_0. Within this box, a regenerative cycle starts and ends. The box consists of the equilibrium states of the system, i.e. the sub-state space in which the simulation process spends most of its time. Furthermore, the regenerative box must be disjoint with all target subspaces, $\Omega_0 \wedge \Omega_j = \varnothing$, for all j. This is because no observations with respect to a target are

recorded when the simulation process is within $\underset{\sim}{\Omega}_0$. A similar concept denoted *load cycles* was applied in [HH94], also similar to the *A-cycles* in [GHNS93, LC96].

Let $p_k(i)$ be the steady state probability of generator k having i entities. The accumulated probability is $P_k(i) = \sum_{l=0}^{i} p_k(l)$. The $p_k(i)$ and $P_k(i)$ are estimated by short pre-simulations. An upper and lower boundary are defined for each dimension, or generator, of the regeneration box. Let $\omega_k^{(u)}$ denote the state that corresponds to the u-quantile in the estimated steady state distribution for generator k, defined as

$$\omega_k^{(u)} = \{i | (\hat{P}_k(i) \le u \le \hat{P}_k(i+1))\}. \tag{6.1}$$

The upper quantile of the regenerative box is $u_+ = 0.50 + \varepsilon$, while $u_- = 0.50 - \varepsilon$ is the lower quantile. The ε is 0.33 for all experiments in this chapter.

Example 6.1: The regenerative box is identified from the pre-simulations of case 3.1 in section 6.5. Now, using $\varepsilon = 0.33$, the upper quantile is $u_+ = 0.83$ and the lower quantile $u_- = 0.17$. Hence, the corresponding states that give the boundaries of the regenerative box are obtained from the estimates $\hat{p}_k(i)$ and $\hat{P}_k(i)$,

- upper bound: $\underset{\sim}{\omega}^{(u_+)} = \{1, 2, 1, 3, 2, 2, 1, 2, 2, 1, 3, 2, 3, 2, 4, 1\}$,

- lower bound: $\underset{\sim}{\omega}^{(u_-)} = \{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0\}$.

The concepts of the regenerative box and the expected cycle time related to it, are not known in advance. Hence, a pre-simulation is required, consisting of the following phases:

1. *Regenerative box identification* is made by a block simulation experiment where the equilibrium states are identified.

2. *Relative distribution* of the states within the regenerative box is obtained by a second block simulation.

3. *Regenerative cycle estimation.* The cycle time is defined as the time between two departures from the regenerative box (identified after phase 1). Each cycle starts in an arbitrary state within $\underset{\sim}{\Omega}_0$, sampled from the box distribution estimated in phase 2. Otherwise, the cycles will not be independent.

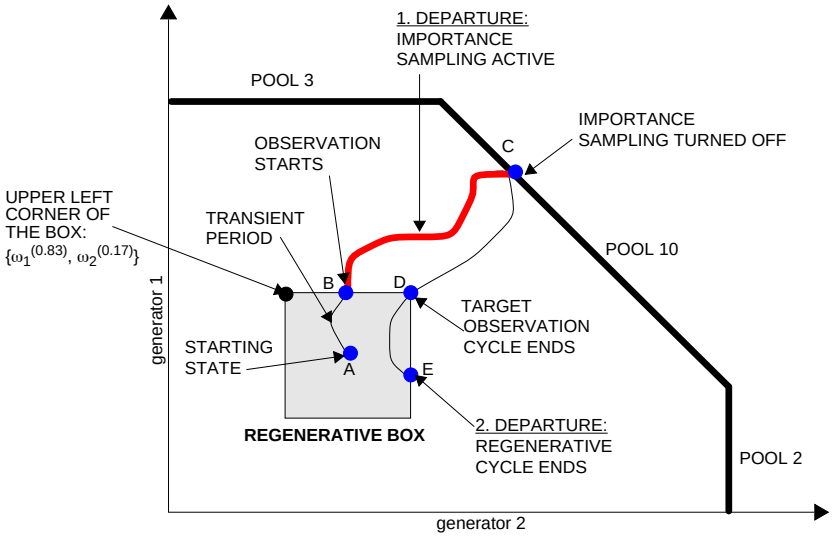


Figure 6.1 : A regenerative cycle in the state space of example 4.2 which includes a visit to the target subspace of pool 10.

The pre-simulations should be, and typically are, much less computer demanding than the main simulation. This will, of course, depend on the termination rule used in the pre-simulation. Block simulations are used for identification of the regeneration box and for estimation of the steady state distribution inside this box. The number of events that constitutes the transient period, which is removed, must be defined, in addition to the number of simulated events in the steady state behaviour. In this chapter, the number of events in the transient and steady state periods are determined experimentally. The third pre-simulation, where the regenerative cycles are estimated, continues until the relative error of the cycle estimate is approximately 1-2%. The main simulation, where importance sampling and the adaptive parameter biasing are applied, is termination when the estimates have a relative error of approximately 10-15%.

Example 6.2:

As a typical example, consider the total simulation time (1300.7 sec) of case 2 in section 6.4. The time elapsed at the different phases is distributed as follows:

i. Regenerative box identification 4.2 sec (0.3%) {50,000 events}

ii. Regenerative box distribution 17.7 sec (1.4%) {50,000 events}

iii. Regenerative cycle estimation 21.8 sec (1.7%) {10,000 cycles}

iv. Importance sampling main simulation 1257.0 sec (96.6%) {100,000 cycles}

The complete algorithm of the main simulation consists of the following steps (see figure 6.1 for an example of a cycle):

- Sample an initial state (state A) according to the relative distribution of the states within the regenerative box.

- Start a regenerative cycle at the first departure from the box (state B) and switch to the importance sampling parameters.

- Turn off importance sampling biasing if a visit to a target is observed (state C).

- End the cycle at first return to the regenerative box (state D). This will save simulation time without loss of target observations. No targets can be observed inside the regenerative box, i.e. in the remaining of the regenerative cycle (from state D to state E in the figure). Hence, the sub-path from D to E can be considered as simulation overhead that can be removed.

6.3 Case 1: No priority nor alternative routing

When all the generators have the same priority and no routing alternatives, it is feasible to obtain exact blocking probabilities for a model of moderate size, for instance by the *convolution method* [Ive87]. The simulation results in this case are compared with exact values which will demonstrate the efficiency and precision of the method. This case and the results are also presented in [Hee96, Hee97a].

6.3.1 Generators and resource pools

The network consists of 8 nodes that are interconnected by 12 links. The topology is described in figure 6.2. Each link is modelled as a resource pool, see table 6.1.

The network is offered traffic from $K = 10$ different user types, modelled as generators. A call setup generates an entity that requires c_{kj} resources of link j. In this case, $c_{kj} = c_k$ for all links in the routing set, $j \in \Phi_k$. Recall from chapter 4 that the routing set contains the sequence of resource pools that models the fixed route between the origin and desti-

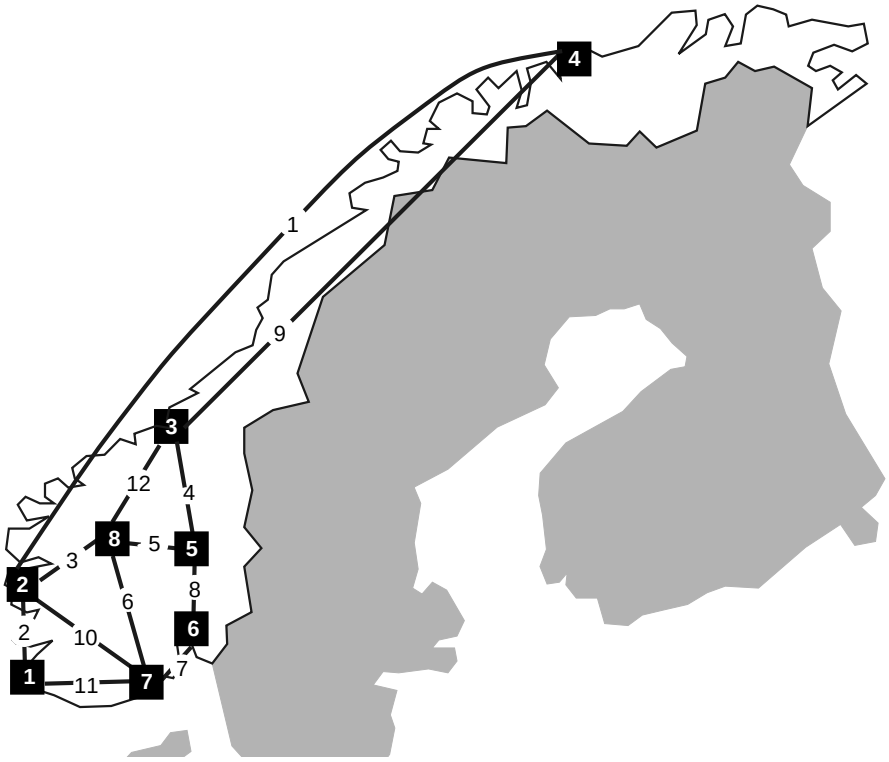


Figure 6.2 : Topology of system case 1.

nation node of a connection. The complete set of attributes of all $K = 10$ generators is given in table 6.2.

Example 6.3:

Consider generator $k = 4$. This is a model of the user type that is generating connections between node 2 and 6 via node 8 and 5, see the topology in figure 6.2. The connection requests $c_4 = 1$ channel from each of the links 3, 5 and 8, and hence the routing set becomes $\Phi_4 = \{3, 5, 8\}$.

6.3.2 Simulation setup

The regenerative box is identified by conducting two pre-simulations of 10000 events each. First, the accumulative steady state probabilities $P_k(i)$ of each generator k are estimated. Then the steady state distribution is estimated, given a state inside the regenerative box. The box is defined by using the $0.5 \pm \varepsilon$ quantile in the estimated $\hat{P}_k(i)$ distribution with $\varepsilon = 0.33$.

Table 6.1: The resource pools of case 1.

j	Γ_j	N_j
1	{1,6,7}	20
2	{2,7}	12
3	{1,2,4,8}	22
4	{2,8}	12
5	{2,3,4}	15
6	{1,3,10}	23
7	{1,5,9,10}	25
8	{4}	7
9	{6}	6
10	{5,6}	7
11	{9}	12
12	{8,10}	26

Table 6.2: The generators of case 1.

k	λ_k	M_k	μ_k	S_k	c_k	p_k	Φ_{k0}
1	0.03	∞	1.0	5	4	0	{1,3,6,7}
2	0.05	∞	1.0	12	1	0	{4,5,3,2}
3	0.03	∞	1.0	5	3	0	{5,6}
4	0.06	∞	1.0	7	1	0	{3,5,8}
5	0.07	∞	1.0	7	1	0	{10,7}
6	0.04	∞	1.0	6	1	0	{9,1,10}
7	0.04	∞	1.0	6	2	0	{1,2}
8	0.01	∞	1.0	4	3	0	{4,12,3}
9	0.05	∞	1.0	6	2	0	{11,7}
10	0.04	∞	1.0	7	3	0	{12,6,7}

The cycle time is estimated from simulation of 10000 regenerative cycles. The property of interest, i.e. the blocking probabilities, are estimated by conducting a regenerative simulation experiment with 50000 cycles. The pre-simulations are conducted only once, while the main simulations, where the properties of interest are estimated, are replicated 20 times.

6.3.3 Results

The exact blocking probabilities are provided by Prof. V. B. Iversen who has used his convolution method [Ive87]. This method is rather computer intensive for networks of the size of case 1. The computation effort increases exponentially when new links or source types are added. According to Prof. V. B. Iversen, the exact solution required more than 9 hours of CPU time on a HP735 (100 Mhz) WS, and more than 55 Mbytes of memory.

In contrast, each replication of the simulation experiment required less than 10 minutes of CPU-time on an Axil320 (120 Mhz) WS. The computational complexity of the simulation approach is $O(K \cdot J)$.

The simulation results with 95% error bars are plotted in figure 6.3 together with the exact results.

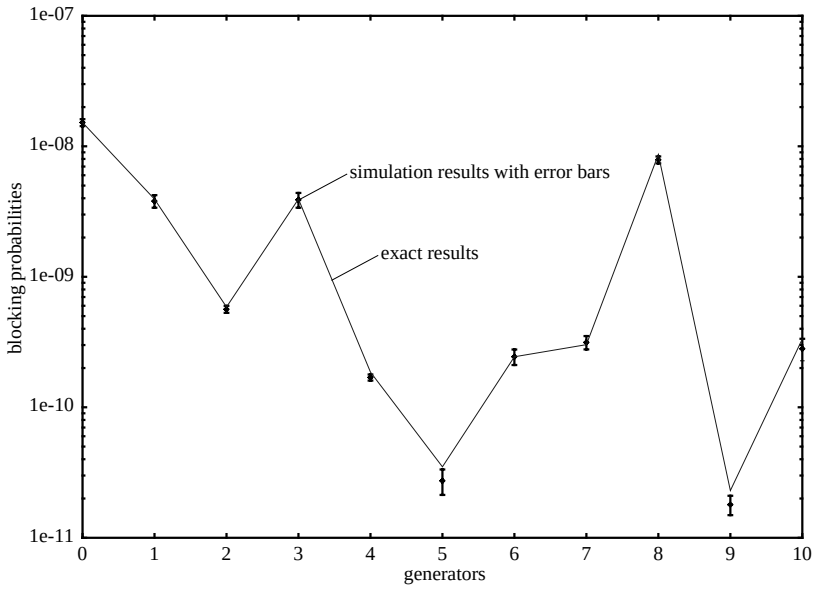


Figure 6.3 : The call blocking for all generators are close to the exact values.

6.3.4 Observations

The most important observations made in this simulation study are:

1. The estimated loss probabilities show good agreement with the exact values for *all generators.*

2. The generators with the largest blocking probabilities, (generator 1,3 and 8) have the best precision.

3. Compared to simulation, the calculation of exact values are computer intensive.

4. All estimates are within ~10% relative to the exact values. The 95% confidence interval includes the exact values for all source types, except generators 5 and 9.

5. The mean observed likelihood ratio is close to the expected value 1, $\bar{L} = 0.988$ with standard error $S_{\bar{L}} = 0.00659$. See section 3.5.1 for discussion of the observed likelihood ratio used for validation of the importance sampling estimates.

6.4 Case2: Improving the quality of service by rerouting

This case demonstrates the use of rerouting caused by traffic overload on the primary route. Two simulation series are produced, one with an alternative (secondary) route, and one with only a primary route. The improvement in the quality of service by adding an alternative route, is estimated by means of the reduced blocking probability for each generator.

The rough approach used for dimensioning of the network capacities, and to identify the primary and secondary routes, is presented in appendix F.

6.4.1 Generators and resource pools

The network consists of 6 nodes interconnected by 10 links. The topology is described in figure 6.4. The resource pools that model the links are described in table 6.3. The Γ_j sets include the generators that have resource pool j in their primary route Φ_{k0}.

The network is offered traffic from $K = 15$ generators. All attributes of the generators are listed in table 6.4. Observe that two alternative routes are defined in the table, Φ_{k0} and Φ_{k1}. The route $r = 0$, Φ_{k0}, is the primary route. As long as sufficient capacity is available on all links in Φ_{k0}, this route is chosen. When an overload situation occurs, two alternative strategies can be used:

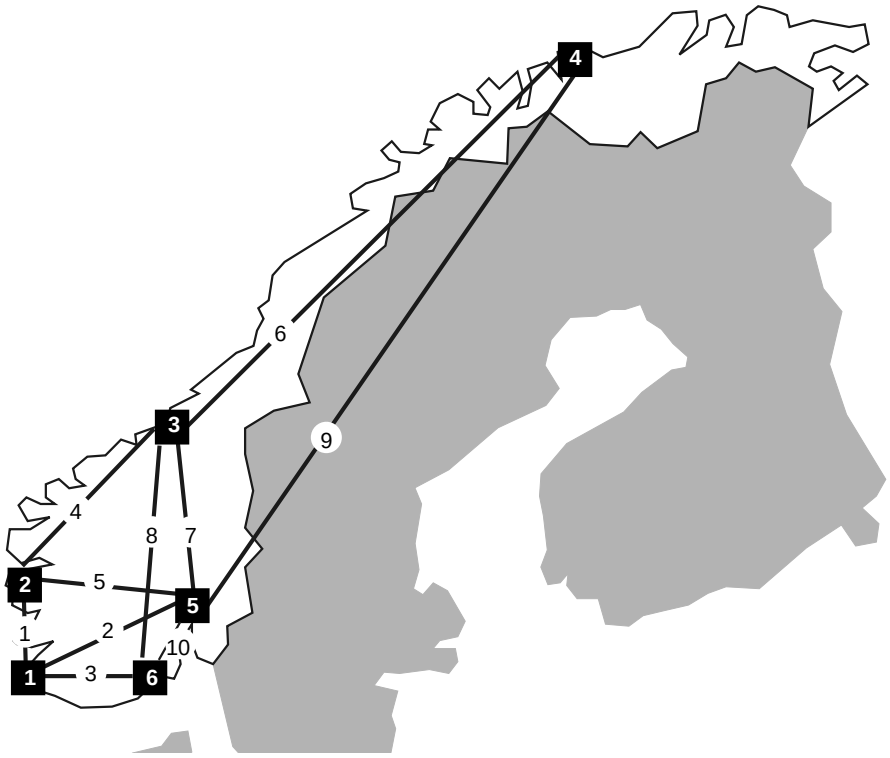


Figure 6.4 : Topology of system case 2 and 3.

- *Case 2.1:* arriving calls are lost, or

- *Case 2.2:* arriving calls are connected via the secondary route, $r = 1$, Φ_{k1}.

The simulation experiments in this section compare the blocking probability in case 2.1 and 2.2.

The capacity required by each entity is, in this case, $c_{kj} = c_k$ for all links in both routing sets, $j \in \Phi_{kr}$, $(r = 0, 1)$.

Table 6.3: The resource pools of case 2.

j	Γ_j	N_j
1	{1,2,3,9}	33
2	{4,6,18}	33
3	{5,9}	37
4	{2,3,6,7}	27
5	{8}	25
6	{3,7,10,13,14}	30
7	{11,13}	28
8	{12,14}	30
9	{-}	21
10	{15}	37

Table 6.4: The generators of case 2.

k	λ_k	M_k	μ_k	S_k	c_k	p_k	Φ_{k0}	Φ_{k1}
1	0.16	∞	1.0	16	4	1	{1}	{2,5}
2	0.19	∞	1.0	16	4	1	{1,4}	{3,8}
3	0.14	∞	1.0	16	4	1	{1,4,6}	{2,9}
4	0.39	∞	1.0	16	4	1	{2}	{3,10}
5	0.28	∞	1.0	15	4	1	{3}	{2,10}
6	0.20	∞	1.0	13	4	1	{4}	{1,3,8}
7	0.15	∞	1.0	15	4	1	{4,6}	{5,9}
8	0.41	∞	1.0	12	4	1	{5}	{1,2}
9	0.30	∞	1.0	18	4	1	{1,3}	{2,10}
10	0.18	∞	1.0	15	4	1	{6}	{7,9}
11	0.48	∞	1.0	14	4	1	{7}	{8,10}
12	0.35	∞	1.0	15	4	1	{8}	{1,3,4}
13	0.36	∞	1.0	15	4	1	{6,7}	{9}
14	0.26	∞	1.0	15	4	1	{6,8}	{9,10}
15	0.72	∞	1.0	18	4	1	{10}	{2,3}

6.4.2 Simulation setup

The regenerative box is identified by conducting two pre-simulations of 50000 events each. First, the accumulative steady state probabilities $P_k(i)$ of each generator k are estimated. Then the steady state distribution is estimated, given a state inside the regenerative box. The box is defined by using the $0.5 \pm \varepsilon$ quantile in the estimated $\hat{P}_k(i)$ distribution with $\varepsilon = 0.33$.

The cycle time is estimated from simulation of 10000 regenerative cycles. The property of interest, i.e. blocking probabilities, is estimated by conducting a regenerative simulation experiment with 100000 cycles, both of the model with primary route only, and the model with secondary routes.

6.4.3 Results

No exact results have been established for comparison with the simulation results. However, to get an indication of the correctness of the simulation results, the blocking criteria used in the network dimensioning is added to the plot in figures 6.5(a). In appendix F the details are given. In addition, recall from section 3.5.1, that the observed likelihood ratio may serve as an indication.

The simulation of the model with primary route only, was running for 21 min. 40.7 sec on a Sun Ultra 1 (167Mhz), while with secondary route for 1 hour 21 min. 2.8 sec. The simulation results are presented in figure 6.5 where the blocking of case 2.1 and 2.2 are given. The reduction in the blocking probabilities by introduction of the alternative routes is given in figure 6.6. This plot is obtained by taking the ratio between the estimated blocking in case 2.2 and case 2.1. Two experiments without importance sampling are conducted to calculate the speedup between direct and importance sampling simulation. The detailed results are given in appendix G.

6.4.4 Observations

The most important observations made in this simulation study are:

1. All simulated blocking probabilities are close to the approximation obtained in appendix F.

2. The observations are most accurate for the largest blocking probabilities.

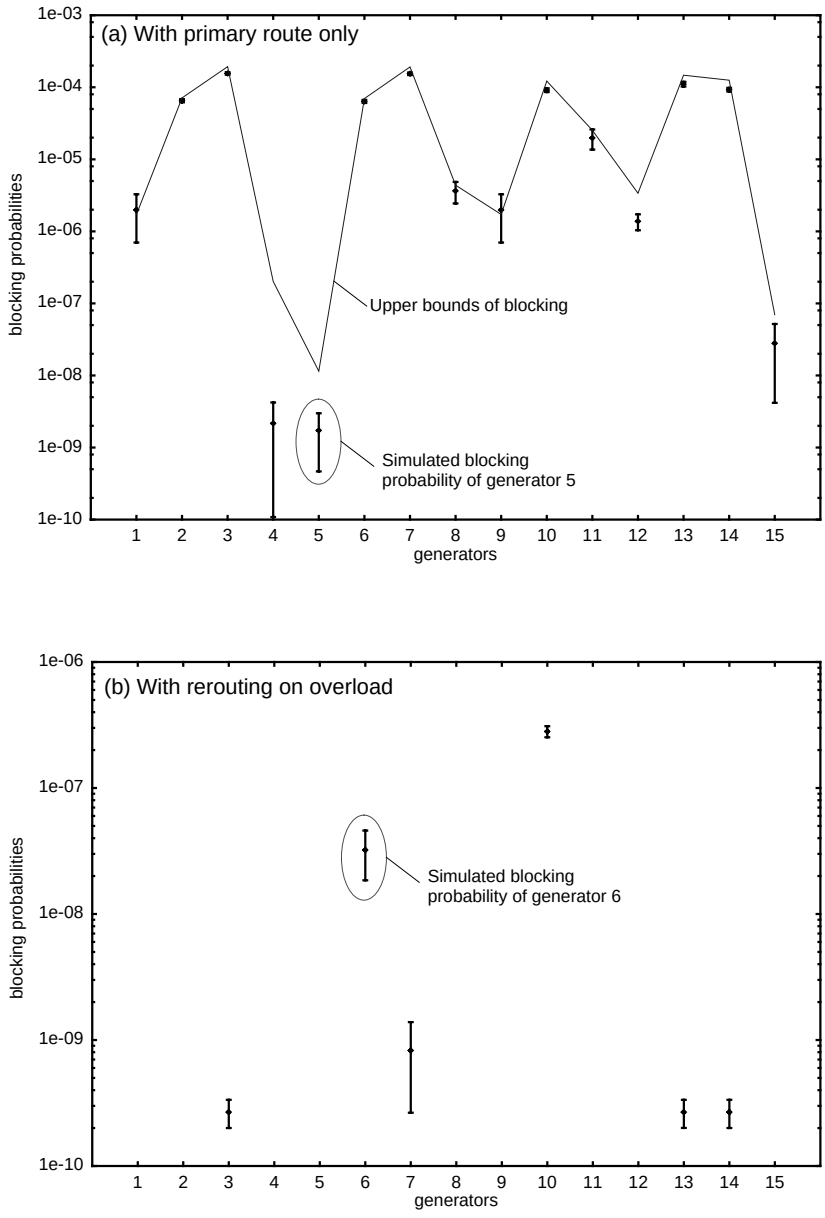


Figure 6.5 : The blocking probabilities of generator 1-15 are significantly reduced for all generators by inclusion of a secondary route, see figure 6.6 *for the reduction factor. All observations below 1e-10, if any, are removed.*

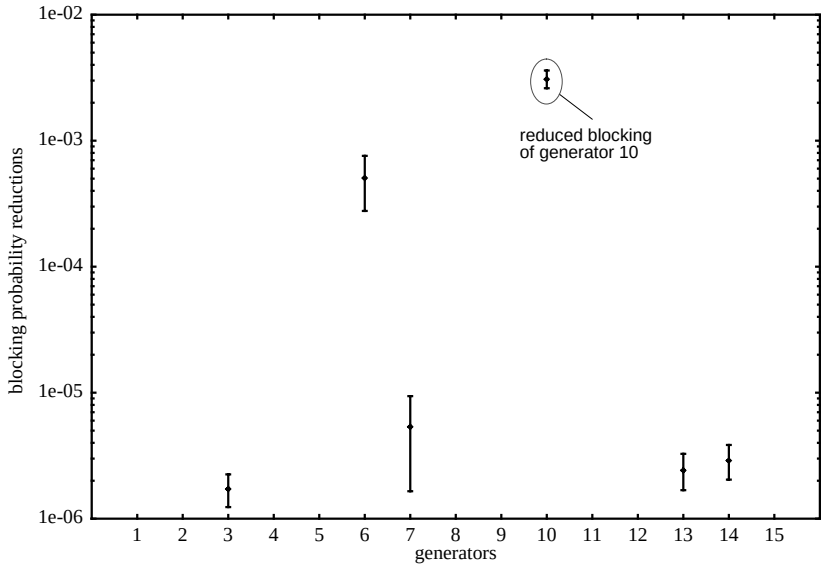


Figure 6.6 : The reduction in the blocking probabilities is significant all generators when a secondary route is introduced.
All observations below 1e-10, if any, are removed.

3. In the current example where sufficient capacity is available for rerouting purposes, switching to an alternative route on overload will significantly improve the performance as reduced blocking probabilities.

4. The mean observed likelihood ratio is not far from its expected value, $E(L) = 1$, for both experiments:

   - $\bar{L} = 0.898$ with standard error $S_{\bar{L}} = 0.00407$,

   - $\bar{L} = 0.941$ with standard error $S_{\bar{L}} = 0.00574$.

   This indicates that the simulation results are accurate.

5. A significant speedup is observed in the network with rerouting where the blocking probability is in the order of 10^{-7}. Counting the number of cycles, a speedup of approximately 650 is observed, and including the simulation overhead introduced by

the adaptive strategy, the speedup is approximately 50. In the case with no rerouting, where the blocking is in the order of 10^{-4}, no speedup is observed. Tables G.1-G.2 contain more details.

6.5 Case 3: Disturbing low priority traffic

To demonstrate the use of the preemptive priority mechanism, low priority traffic is added to the case 2. Three simulation series are produced where the blocking probabilities of low priority traffic are estimated. First the network is simulated with only low priority traffic generators, then mixed with high priority traffic, and finally mixed with both high priority traffic and affected by link failures.

The rough approach used for dimensioning of the network capacities, and to identify the primary and secondary routes, are presented in appendix F.

This case and the results are previously presented in [Hee98].

6.5.1 Generators and resource pools

The network consists of 6 nodes interconnected by 10 links. The topology is described in figure 6.4. The resource pools that model the links are described in table 6.5.

Table 6.5: The resource pools of case 3.

j	Γ_j	N_j
1	{1,2,3,9,21,23,27,31,32}	33
2	{4,6,16,18,20,23,24,30,31}	33
3	{5,9,17,19,21,27,30}	37
4	{2,3,6,7,27}	27
5	{8,16,22}	25
6	{3,7,10,13,14}	30
7	{11,13,25}	28
8	{12,14,17,21,26}	30
9	{18,22,25,28,29}	21
10	{15,19,20,24,26,29}	37

The network is offered both low and high priority traffic from 32 generators connecting all 10 nodes. The high priority end-to-end connections use the alternative route when the primary route is either congested or disconnected on arrival of new entities. The corresponding low priority end-to-end connections, have their primary (and only) route along the secondary route of the high priority connections. High priority traffic may preempt low priority connections, and the disconnected calls are lost. A link failure is considered to be a high priority “call” which is preempting all connections on a link and allocating the entire link capacity, N_j.

The 32 generators are: 15 high priority generators with bursty traffic ($c_k = 4$), 15 low priority traffic generators with smooth traffic ($c_k = 1$), 1 low priority generator with busty traffic ($c_k = 3$), and a generator of link 1 failure where $c_k = 33$, i.e. all channels on link 1, N_1. The channel capacities are, N={33, 33, 37, 27, 25, 30, 28, 30, 21, 37}. Generators 1-15 are the same as the generators used in case 2 in section 6.4. Their attributes are given in table 6.4, while the new generators 16-32 are defined in table 6.6.

6.5.2 Simulation setup

The regenerative box is identified by conducting two pre-simulations of 50000 events each. First, the accumulative steady state probabilities $P_k(i)$ of each generator k are estimated. Then the steady state distribution is estimated, given a state inside the regenerative box. The box is defined by using the $0.5 \pm \varepsilon$ quantile in the estimated $\hat{P}_k(i)$ distribution with $\varepsilon = 0.33$.

The cycle time is estimated from simulation of 10000 regenerative cycles. The property of interest, i.e. blocking probabilities, is estimated by conducting three different regenerative simulation experiments:

- *Case 3.1:* Low priority traffic only: simulated by 100000 cycles.
- *Case 3.2:* Low priority mixed with high priority traffic: simulated by 10000 cycles.
- *Case 3.3:* Low priority mixed with high priority traffic and exposed to link failures: simulated by 10000 cycles.

In all experiments the importance sampling biasing is defined to change parameters with respect to provoke the rare events associated with generator 23 and 31. This means that the importance sampling parameters are switched off when blocking is observed in either

Table 6.6: The generators 16-32 of case 3. The generators 1-15 are the same as in case 2, see table 6.4 for their attributes.

k	λ_k	M_k	μ_k	S_k	c_k	p_k	Φ_{k0}	Φ_{k1}
16	0.64	∞	1.0	33	1	2	{2,5}	-
17	0.76	∞	1.0	37	1	2	{3,8}	-
18	0.57	∞	1.0	33	1	2	{2,9}	-
19	1.57	∞	1.0	37	1	2	{3,10}	-
20	1.14	∞	1.0	37	1	2	{2,10}	-
21	0.80	∞	1.0	37	1	2	{1,3,8}	-
22	0.60	∞	1.0	25	1	2	{5,9}	-
23	1.65	∞	1.0	33	1	2	{1,2}	-
24	1.19	∞	1.0	37	1	2	{2,10}	-
25	0.71	∞	1.0	28	1	2	{7,9}	-
26	1.94	∞	1.0	37	1	2	{8,10}	-
27	1.41	∞	1.0	37	1	2	{1,3,4}	-
28	1.45	∞	1.0	21	1	2	{9}	-
29	1.05	∞	1.0	37	1	2	{9,10}	-
30	2.90	∞	1.0	37	1	2	{2,3}	-
31	0.55	∞	1.0	11	3	2	{1,2}	-
32	1e-6	1	0.1	1	33	0	{1}	-

generator 23 or 31. The generator 23 and 31 are chosen because they have the same route and priority level, but have different resource requirements, $c_{23} \neq c_{31}$.

6.5.3 Results

No exact results have been established for comparison with the simulation results. However, to get an indication of the correctness of the simulation results, the blocking criteria from appendix F are added to figures 6.8-6.9. In addition, recall from section 3.5.1, that the observed likelihood ratio may serve as an indication.

The simulations of the model with only low priority traffic was running for 51 min. 56.1 sec on a Sun Ultra 1 (167Mhz). With high priority traffic added, the simulation time was 1 hour 31 min. 57.4 sec, while with link failure it was 1 hour 31 min. 27.0 sec. For calculation of the speedup between direct and importance sampling simulation, a series of

experiments without importance sampling are conducted. The detailed results are given in appendix G.

The simulation results of case 3.1 and 3.2 are presented in the figures 6.7 and 6.8. The blocking probabilities of all low priority traffic generators are included. In figure 6.9, the blocking probabilities are given for generator 23 and 31.

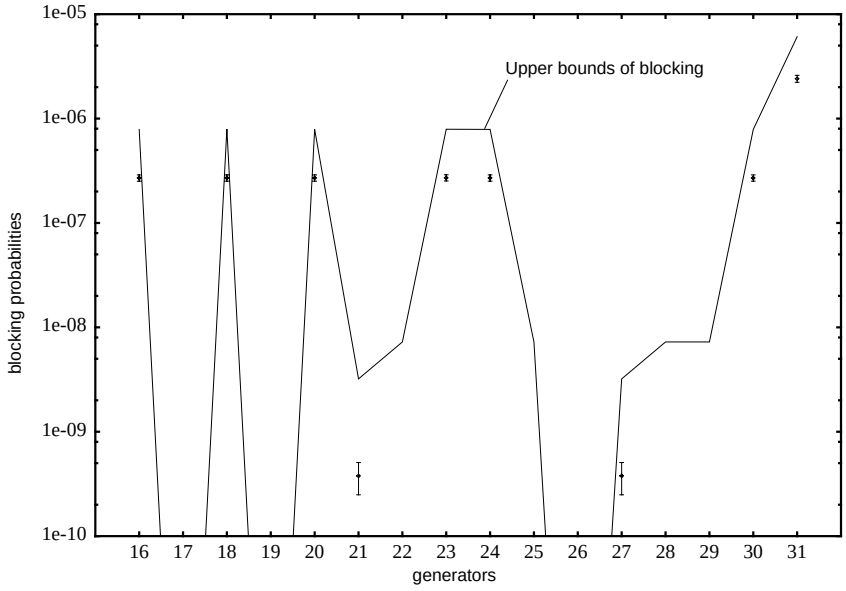


Figure 6.7 : Plot of blocking probabilities in case 3.1. All observations below 1e-10, if any, are removed.

6.5.4 Observations

The most important observations made in this simulation study are:

1. The simulated blocking of generator 23 and 31 is less than the approximated blocking obtained by neglecting correlations between the links. It is expected that the rough approach produces conservative values because this assumption is not realistic.

2. The accuracy of the estimates of generator 23 and 31 is at least as good as the other estimates. This is as expected because the importance sampling experiment was set up to provoke rare events which involve generator 23 or 31.

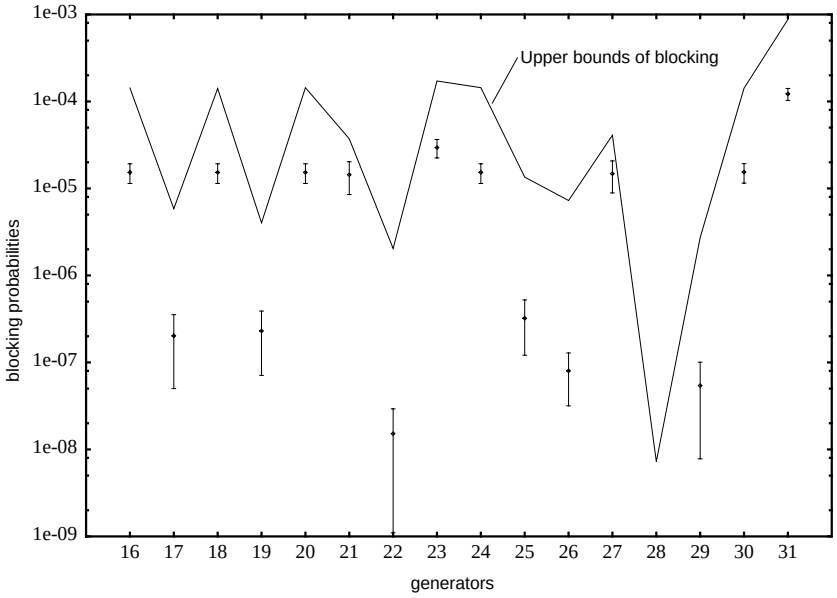


Figure 6.8 : Plot of blocking probabilities in case 3.2.

3. The mean observed likelihood ratio for the three cases are

   - $\bar{L}_{3.1} = 1.01$ with standard error $S_{\bar{L}_{3.1}} = 0.067$.

   - $\bar{L}_{3.2} = 0.885$ with standard error $S_{\bar{L}_{3.2}} = 0.098$.

   - $\bar{L}_{3.3} = 0.742$ with standard error $S_{\bar{L}_{3.3}} = 0.073$.

   This indicates that the simulation results are likely to be correct, at least for case 3.1 and maybe for 3.2. while the biasing applied in case 3.3 is perhaps slightly too large? The reason may be that the failure process gives too large contribution to the likelihood ratio?

4. A tremendous speedup is observed in case 3.1 comparing the *relative error* of the estimated blocking of generator 23 and 31. The sample mean $\hat{\gamma}$ from the direct simulation experiments were more than 1 order of magnitude less than the importance sampling estimates, $\hat{\gamma}_{\mathrm{IS}}$. The importance sampling estimates were in the same order of magnitude as the approximate blocking values given in table F.7 in appendix F. Hence, the

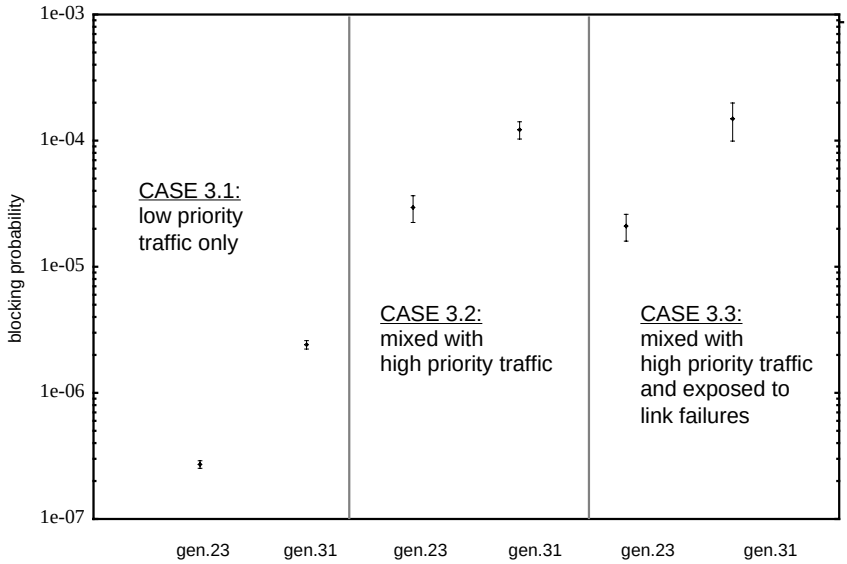


Figure 6.9 : The blocking of smooth and bursty low priority traffic exposed to influence by high priority traffic with and without link failures.

speedups given by the efficiency measures in (2.11) are misleading as an indication of the speedup given by importance sampling. In cases 3.2 and 3.3, where the blocking probabilities are in the order of 10^{-4}, no speedup are observed.

6.6 Closing comments

This chapter demonstrates the feasibility of the simulation framework with importance sampling and the adaptive parameter biasing proposed in chapter 5. The simulation results show that it is feasible to use this framework to conduct network simulations where the users have different service requirements, preemptive priorities, and may switch to an alternative route when the primary route is blocked.

Importance sampling provides good and stable estimates when it is incorporated in regenerative simulation. The definition of the regenerative state is critical for the efficiency of the regenerative simulations. Hence, the experiments in this chapter defines, by pre-simulations, a *regenerative box* where the most likely states are included. A regenerative cycle starts and ends from this set of states. Further work should be done to determine the number of events that need to be simulated in the pre-simulations of the regenerative box.

The sensitivity to the ε-factor which defines the size of the box, should also be further examined.

Although some of the simulation results gave interesting insight in specific properties of the systems used as examples, this was not the objective of this study. Hence, only the observations that demonstrate the feasibility of the importance sampling strategy are commented.

- In importance sampling simulation, the rare events of interest should occur in proportion to the probability of the targets. This means that the properties of interest with the largest value, or that are visited most frequently, will have the most precise estimate. The simulation results in this chapter show this behaviour.

- The original simulation parameters are reinstated when one of the specified rare events of interest is observed. The properties that depend on these rare events will have the most accurate estimates, while other properties are “by-products” and will generally have less accurate estimates.

Example 6.4: Consider the blocking in generator k to be the property to be estimated. All state subspaces, Ω_j where $j \in \Phi_k$ are defined as targets. The importance sampling biasing is switched off when a visit to any of the Ω_j, $j \in \Phi_k$ is observed. The properties of generator k, and other generators dependent on these Ω_j, will be precisely estimated. Other properties may also be observed, but will, because the importance sampling setup is not prepared for this, produce less precise estimates.

- When the simulations produce accurate estimates, it is observed that the sampled mean of the likelihood ratio is close to its expected value 1.

Further experiments should be conducted to gain more experience with the adaptive biasing. Of particular interest are for instance:

- Further simulation experiments on models with a known exact solution, or a good approximation. This is required to get additional insight in the behaviour of the simulation strategy, e.g. what is the observed mean likelihood ratio in the case of correct and incorrect simulation results.

- Further simulations of models that combine dependability and traffic properties, e.g. similar to case 3.3.

- Further simulations of other properties than the blocking probabilities, both steady state and transient properties.

Application of alternative rare event techniques to networks with the characteristics given in this chapter should be investigated. The RESTART technique proposed in [VAVA91] is an approach that has proven to be an efficient means for rare event simulations. However, the problem with defining RESTART states in a multidimensional model, see chapter 2, must be solved. In [GHSZ97] it is proven that, except for very specific models, it is not possible to define RESTART states in multidimensional models so that the most likely simulated path is the same as the optimal path from origin to the target. The remaining question is how robust and efficient is a sub-optimal definition of RESTART states?

In chapter 2, a combination of importance sampling and RESTART was briefly mentioned. In the network models it may be possible to define RESTART states associated with “meta events” like rerouting, disconnection, failure, etc., and to RESTART simulations with importance sampling parameters from these states.

7
Conclusions

When the network has very strict quality of service requirements, a traditional, direct simulation approach becomes too inefficient to be of practical use, and a speedup technique like importance sampling or RESTART must be applied. In this thesis, simulation with importance sampling is used for performance evaluation of network models with balanced utilisation of resources.

The main challenge with importance sampling is to determine the best way of changing the underlying simulation process. This *biasing* of the simulation parameters is very model specific, which means that previous results does not necessarily apply to new models. Optimal parameter biasing is known only for a few, simple models. The importance sampling efficiency is sensitive to the parameter biasing. Importance sampling is very efficient when optimal, or close to optimal, biasing is applied, while the variance of the estimates becomes unbounded when the biasing is too strong.

To incorporate importance sampling in simulation of a network with balanced utilisation of resources, previous strategies for biasing the simulation parameters are no longer efficient. Hence, a new adaptive parameter biasing has been developed, and its feasibility is demonstrated on several network examples. This, and other contributions, are listed in the following section. Section 7.2 discusses further work.

7.1 Main contributions

The main contribution of this thesis is the description of a new adaptive biasing of the importance sampling parameters. With this biasing, importance sampling can be applied to multidimensional models, e.g. descriptions of telecommunication networks with balanced utilisation of resources. The adaptive biasing removes the unrealistic constraint that the system performance is dominated by single bottleneck. Previous strategies are based on such an assumption which simplifies the biasing significantly.

A new, flexible framework is defined for modelling both dependability and traffic aspects of communication networks. The adaptive biasing applies to models described within this framework.

Several network examples are defined and their performance is evaluated by importance sampling simulations. These experiments serve as demonstration of the feasibility of the adaptive biasing. The network examples include users with different resource requirements, alternative routing strategies, and preemptive priorities. One of the examples also includes one link failure process. The simulation results are compared with exact results when they are available, or with approximate results used for dimensioning of the network examples. The comparison shows that the results from the importance sampling simulation are fairly good.

A heuristic that is derived from the general study of importance sampling, proposes that the observed likelihood ratio may serve as an indication of the accuracy of the simulation results. Explicit expressions of the variance of the importance sampling estimates and the likelihood ratio are developed for an M/M/1/N queue. The study of these variance expressions under different importance sampling parameter biasing, shows that the variance is (close to) 0 when the biasing is (close to) optimal. However, the variance is infinite when the biasing is outside a stable region. The exact bounds of this region have been established for M/M/1/N models. It is observed, both by theoretical analysis of a simple systems and from the network simulations, that when the importance sampling estimates are close to its true value, the mean observed likelihood ratio is close to its expected value 1 and has low variance.

Although, importance sampling has been the main focus, the thesis contains a brief overview of a broader range of speedup simulation techniques. Some of them may be combined to achieve an additional speedup. A brief comparison between the two rare event provoking techniques, RESTART and importance sampling, is included. Importance sampling with optimal parameters will always be at least as good as RESTART, typically far better. However, optimal importance sampling parameters are only known for a limited class of models. Several experiments were conducted to compare RESTART and a non-optimal importance sampling. The result was that for some parameters a small additional speedup was observed compared to using either RESTART or importance sampling separately.

Another successful application of importance sampling is demonstrated. Importance sampling is applied in a trace driven simulation of a multiplex of MPEG coded video streams. The simulation results are compared with another approach based on stratified sampling, and with direct simulation. The comparison shows that importance sampling provides a significant speedup.

7.2 Further work

Several directions for further research can be given, both regarding the adaptive biasing technique, and on other approaches to rare event network simulations.

The adaptive parameter biasing ought to be further tested on additional networks. Other measures, dependent on rare events, than the blocking probability should also be examined. For this purpose, real sized networks should be described where the quantities of interest are known, either from analytic (numerical) solutions, or from measurements.

New guidelines should be developed to set up importance sampling simulations. For instance, the use of the observed likelihood ratio as indication of correctness in simulation results should be further investigated.

The influence on the simulation efficiency of adding the adaptive technique should be further investigated. It has been observed that the target distribution estimate is less precise for some model parameters, and an increased understanding should be developed to determine when the estimate is good, and what to do when it is not. This could lead to the need for improvements of the target distribution estimates. However, it is strongly believed that the basic idea of the adaptive technique is an efficient approach that should be applied when conducting importance sampling simulations of multidimensional models.

The adaptive biasing of importance sampling is developed under the assumption of a continuous time Markov chain. It is expected that the model framework and the adaptive biasing can be generalised, for instance, to a *generalised semi-Markov process*. In non-Markovian models, importance sampling can be applied using a uniformisation technique as described in [HSN94].

To get a more complete picture of rare event simulation of network models, alternative techniques like RESTART should be investigated. This requires that the problem with defining optimal thresholds in multidimensional models is solved. The experimental series should be defined to include both a series of separate RESTART experiments for

comparison with importance sampling, and a series where RESTART and importance sampling are combined.

References

[AHH96] Ragnar Ø. Andreassen, Poul E. Heegaard, and Bjarne E. Helvik. Importance sampling for speed-up simulation of heterogeneous MPEG sources. In Emstad et al. [EHM96], pages 190–203.

[And97] Ragnar Andreassen. *Traffic performance studies of MPEG variable bitrate video over ATM*. PhD thesis, Norwegian University of Science and Technology, June 1997.

[BCN96] Jerry Banks, John S. Carson, and Barry L. Nelson. *Discrete-event system simulation*. Prentice Hall, 2nd edition, 1996.

[BFS87] Paul Bratley, Bennet L. Fox, and Linus E. Scharge. *A Guide to Simulation*. Springer-Verlag, 1987.

[Buc90] James A. Bucklew. *Large Deviation Techniques in Decision, Simulation, and Estimation*. Wiley, 1990.

[Car91] Juan A. Carrasco. Failure distance based simulation of repairable fault-tolerant computer systems. In G. Balbo and G. Serazzi, editors, *Proceedings of the Fifth International Conference on Computer Performance Evaluation. Modelling Techniques and Tools*, pages 351 – 365. North-Holland, Feb. 15-17 1991.

[CFM83] Marie Cottrell, Jean-Claude Fort, and Germard Malgouyres. Large deviation and rare events in the study of stochastic algorithms. *IEEE Transaction of Automatic Control*, AC-28(9):13–18, September 1983.

[CG87] Adrian E. Conway and Ambuj Goyal. Monte Carlo Simulation of Computer System Availability/Reliability Models. In *Digest of paper, FTCS-17 - The seventeenth international symposium on fault-tolerant computing*, pages 230 –235, July 6 - 8 1987.

[CHJS92] C. S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin. Effective bandwidth and fast simulation of ATM intree networks. Research Report RC 18586 (81346), IBM, December 1992.

[CHS95] Cheng-Shang Chang, Philip Heidelberger, and Perwez Shahabuddin. Fast simulation of packet loss rates in a shared buffer communications switch. *ACM Transaction on Modeling and Computer Simulation*, 5(4):306–325, October 1995.

[CM65] D. R. Cox and H. D. Miller. *The theory of stochastic processes*. Chapman and Hall, 1 edition, 1965.

[CMBS96] J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain, editors. *Winter Simulation Conference*, 1996.

[CTS93] Russel C. H. Cheng, Louise Traylor, and Janos Sztrik. Simulation of rare queueing events by switching arrival and service rates. In Evans et al. [EMRB93], pages 317–322.

[Dam93] Halim Damerdji. Parametric inference for generalised semi-Markov processes. In Evans et al. [EMRB93], pages 323–328.

[DT93] Michael Devetsikiotis and J. Keith Townsend. Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks. *IEEE/ACM Transactions on Networking*, 1(3):293 – 305, June 1993.

[EHM96] Peder J. Emstad, Bjarne E. Helvik, and Arne H. Myskja, editors. *The 13th Nordic Teletraffic Seminar (NTS-13)*, Trondheim, Norway, 20 - 22 August 1996. Tapir Trykk.

[EMRB93] G. W. Evans, M. Mollagasemi, E. C. Russel, and W. E. Biles, editors. *Winter Simulation Conference*, Los Angeles, California, USA, Dec. 1993.

[FA94] Michael R. Frater and Brian D. O. Anderson. Fast simulation of buffer overflows in tandem networks of GI/GI/1 queues. *Annals of Operation Research*, 49:207–220, 1994.

[Fit90] K. Fitzgerald. Vulnerability exposed in AT&T's 9-hour glitch. *The Institute - A news supplement to IEEE Spectrum*, 1990.

[FLS88] Victor S. Frost, W. W. Larue, and K. Sam Shanmugan. Efficient techniques for the simulation of computer communications networks. *IEEE Journal on selected areas in communications*, 6(1):146 – 157, January 1988.

[Fra93] Michael R. Frater. Fast simulation of buffer overflows in equally loaded networks. *Australian Telecom Research*, 27(1):13–18, 1993.

[Fuj90] Richard M. Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):31–52, Oct. 1990.

[GHNS93] Peter W. Glynn, Philip Heidelberger, Victor F. Nicola, and Perwez Shahabuddin. Efficient estimation of the mean time between failure in non-regenerative dependability models. In Evans et al. [EMRB93], pages 311–316.

[GHSZ96] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Splitting for rare event simulation: an analysis of simple cases. In Charnes et al. [CMBS96], pages 302–308.

[GHSZ97] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. A look at multilevel splitting. Research Report RC 20692, IBM, T.J. Watson Research Center Yorktown Heights, NY 10598, Jan. 1997.

[GK95] Paul Glasserman and Shing-Gang Kou. Analysis of an importance sampling estimator for tandem queues. *ACM transaction on modeling and computer simulation*, 5(1):22–42, January 1995.

[Gly89] Peter W. Glynn. A GSMP formalism for discrete event systems. *Proceedings of the IEEE*, 77(1):14–23, Jan. 1989. Invited paper.

[GSHG92] A. Goyal, P. Shahabuddin, P. Heidelberger, and P.W. Glynn. A unified framework for simulating Markovian models for highly dependable systems. *IEEE Transaction on Computers*, 41(1):36 – 51, 1992.

[Hee95a] Poul E. Heegaard. Comparison of speed-up techniques for simulation. In Ilkka Norros and Jorma Virtamo, editors, *The 12th Nordic Teletraffic Seminar (NTS-12)*, pages 407–420, Espoo, Finland, 22 - 24 August 1995. VTT Information Technology.

[Hee95b] Poul E. Heegaard. Rare event provoking simulation techniques. In *Proceeding of the International Teletraffic Seminar (ITS)*, pages 17.0–17.12, Bangkok, Thailand, 28 Nov.-1 Dec. 1995. Session III: Performance Analysis I, Regional ITC-Seminar.

[Hee95c] Poul E. Heegaard. Speed-up techniques for simulation. *Telektronikk*, 91(2):195–207, 1995. ISSN 0085-7130.

[Hee96] Poul E. Heegaard. Adaptive optimisation of importance sampling for multi-dimensional state space models with irregular resource boundaries. In Emstad et al. [EHM96], pages 176–189.

[Hee97a] Poul E. Heegaard. Efficient simulation of network performance by importance sampling. In *Teletraffic Contributions for the Information Age*, Washington D.C., USA, June 23-27 1997.

[Hee97b] Poul E. Heegaard. Speedup simulation techniques (survey). In C. Görg and C. Kelling, editors, *Workshop on Rare Event Simulation*, Aachen University, Germany, 28.-29. Aug. 1997.

[Hee98] Poul E. Heegaard. A scheme for adaptive biasing in importance sampling. *AEÜ, Special issue on Rare Event Simulation*, 1998. To appear.

[Hei95] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM transaction on modeling and computer simulation*, 5(1):43–85, January 1995.

[Hel95] Bjarne E. Helvik. Synthetic load generation for ATM traffic measurements. *Telektronikk*, 91(2):174–194, 1995. ISSN 0085-7130.

[HH94] Bjarne E. Helvik and Poul E. Heegaard. A technique for measuring rare cell losses in ATM systems. In Labatoulle and Roberts [LR94], pages 917–930.

[HMM93] Bjarne E. Helvik, Ole Melteig, and Leif Morland. The synthesized traffic generator; objectives, design and capabilities. In *Integrated Broadband Communication Networks and Services (IBCN&S)*. IFIP, Elsevier, April 20-23 1993. Copenhagen, Denmark, Also available as STF40 A93077.

[HSN94] Philip Heidelberger, Perwez Shahabuddin, and David M. Nicol. Bounded relative error in estimating transient measures of highly dependable non-Markovian systems. *ACM Transaction on Modeling and Computer Simulation*, 4(2):137 – 164, April 1994.

[Ive87] Villy B. Iversen. A simple convolution algorithm for the exact evaluation of multi-service loss system with heterogeneous traffic flows and access control. In Ulf Körner, editor, *The 7th Nordic Teletraffic Seminar (NTS-7)*, pages IX.3–1–IX.3–22, Lund tekniska högskola, Sweden, 25 - 27 August 1987. Studentlitteratur.

[Jef85] David R. Jefferson. Virtual time. *ACM Transaction on Programming Languages and Systems*, 7(3):404 – 425, July 1985.

[Kel86] F. Kelly. Blocking probabilities in large circuit switched networks. *Advances in applied probability*, 18:473–505, 1986.

[Kel96] Ch. Kelling. A framework for rare event simulation of stochastic petri nets using RESTART. In Charnes et al. [CMBS96], pages 317–324.

[KM88] James F. Kurose and Hussein T. Mouftah. Computer-aided modeling, analysis, and design of communication networks. *IEEE Journal on selected areas in communications*, 6(1):130 – 145, January 1988.

[KRWW93] N. Kalantery, A. P. Redfern, S. C. Winter, and D. R. Wilson. Fast parallel simulation of SS7 in telecommunication networks. In *Proceedings of the First International Workshop on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOT'93)*, pages 171 – 175. IEEE Computer Society Press, 1993.

[KW92] G. Kesidis and J. Walrand. Quick simulation of ATM buffers. In *Proceeding of the 31rst conference on Decision and Control Tucson Arizona*, pages 1018–1019, Arizona, Dec. 1992.

[KW93] G. Kesidis and J. Walrand. Quick simulation of ATM buffers with on-off multiclass Markov fluid sources. *ACM Transactions on Modeling and Computer Simulation*, 3(3):269 – 276, July 1993.

[Lav83] Stephen.S. Lavenberg. *Computer Performance Modeling Handbook*. Academic Press, 1983.

[LC96] Pierre L'Ecuyer and Yanick Champoux. Importance sampling for large ATM-type queueing networks. In Charnes et al. [CMBS96], pages 309–316.

[Les88] A. Lesanovsky. Multistate markov models for systems with dependent units. *IEEE Trans. On Reliability*, 37(5):505 – 511, Dec. 1988.

[Lin94] Yi-Bing Lin. Parallel independent replicated simulation on networks of workstations. In *Proceedings of the 8th Workshop on Parallel and Distributed Simulation (PADS'94)*, pages 71 – 81. IEEE Computer Society Press, July 6-8 1994.

[LO88] P. A. W. Lewis and E. J. Orav. *Simulation Methodology for Statisticians, Operations Analysts and Engineers*, volume I. Wadsworth & Brooks/Cole Advanced Books & Software, 1988.

[LR85] B. Lubachevsky and K. Ramakrishnan. Parallel time-driven simulation of a network using a shared memory MIMD computer. In D. Potier, editor, *Modelling Tools and Techniques for Performance Analysis*. North-Holland, 1985.

[LR94] J. Labatoulle and J.W. Roberts, editors. *The 14th International Teletraffic Congress (ITC'14)*, Antibes Juan-les-Pins, France, June 6-10 1994. Elsevier.

[LS95] Gunnar Lie and Helge A. Sundt. Importance sampling as speed-up technique for simulation in multi-dimensional systems. Technical report, The

Norwegian Institute of Technology, spring, 1995. Term project in Telematics.

[Maj97] Kurt Majewski. Heavy traffic approximations for large deviations of feedforward queueing networks. Submitted to QUESTA, 1997.

[Man95] Michel Mandjes. Finding the conjugate of Markov fluid processes. *Probability in the engineering and informational sciences*, 9:297–315, 1995.

[Man96a] Michel Mandjes. Overflow asymptotics for large communication systems with general Markov fluid sources. *Probability in the engineering and informational sciences*, 10:501–518, 1996.

[Man96b] Michel Mandjes. Rare event analysis of batch arrival queues. *Telecommunication systems*, 6:161–180, 1996.

[Man96c] Michel Mandjes. *Rare event analysis of communication networks*. PhD thesis, Vrije Universiteit, December 1996.

[McG92] Catherine McGeoch. Analysing algorithms by simulation: variance reduction techniques and simulation speedups. *ACM Computer surveys*, 24(2):195–212, June 1992.

[Nak94] Marvin K. Nakayama. A characterization of the simple failure biasing method for simulation of highly reliable markovian systems. *ACM Transaction on Modeling and Computer Simulation*, 4(1):52 – 88, January 1994.

[NH95] Victor F. Nicola and Gertjan A. Hagesteijn. Efficient simulation of consecutive cell loss in ATM networks. *ATM NETWORKS Performance Modelling and Analysis*, 2, 1995.

[NH97] David Nicol and Philip Heidelberger. Parallel execution for serial simulators. *ACM transaction on modeling and computer simulation*, 6(3):210–242, July 1997.

[Nic97] Victor F. Nicola. Email reply to my list of action point and conclusions after *Workshop on Rare Event Simulation* in Aachen, Aug. 1997. Includes comments from Dr. Phil Heidelberger.

[Orm91] Terje Ormhaug. Plans for a broadband laboratory at Norwegian Telecom Research. Internal document F91/u/304, -305, Norwegian Telecom Research, March 1991. In Norwegian.

[PW89] Shyam Parekh and Jean Walrand. Quick simulation of excessive backlogs in networks of queues. *IEEE Transaction of Automatic Control*, 34(1):54–66, 1989.

[RME96] James Roberts, Ugo Mocci, and Jorma Virtamo (Eds.). *Broadband Network Teletraffic*. Springer, 1996. Final Report of Action COST 242.

[Sad91] J. S. Sadowski. Large deviations and efficient simulation of excessive backlogs in a GI/G/m queue. *IEEE Transaction of Automatic Control*, AC-36:1383–1394, 1991.

[SG94] Friedrich Schreiber and Carmelita Görg. Rare event simulation: a modified RESTART-method using the LRE-algorithm. In Labatoulle and Roberts [LR94], pages 787 – 796.

[VA$^+$94] M. Villén-Altamirano et al. Enhancement of the accelerated simulation method RESTART by considering multiple thresholds. In Labatoulle and Roberts [LR94], pages 797 – 810.

[VAVA91] Manuel Villén-Altamirano and Jose Villén-Altamirano. RESTART: A method for accelerating rare event simulation. In C. D. Pack, editor, *Queueing Performance and Control in ATM*, pages 71 – 76. Elsevier Science Publishers B. V., June 1991.

[VAVA94] Manuel Villén-Altamirano and Jose Villén-Altamirano. RESTART: a straightforward method for fast simulation of rare events. In *Proceedings of the 1994 Winter Simulation Conference*, pages 282–289, December 1994.

[Wol91] Stephen Wolfram. *Mathematica, A System for doing Mathematics by Computer*. Addison-Wesley, 2 edition, 1991.

Appendix A

Importance sampling in trace-driven MPEG simulation[1]

Ragnar Andreassen, Telenor R&D

Poul E. Heegaard and Bjarne E. Helvik,
Norwegian University of Science and Technology, Department of Telematics

Abstract

The ISO/ITU standard MPEG is expected to be of extensive use for the transfer of video/moving pictures traffic in the coming ATM high capacity networks. This traffic will stem from both multimedia services like teleconferencing and video distribution. Hence, MPEG encoded video will be a salient constituent of the overall traffic. The encoding causes large and abrupt shifts in the transferred rate at the frame borders and induce strong periodic components, i.e. it generates a traffic pattern that is difficult to handle with a guaranteed QoS and a sufficiently large multiplexing gain. All methods, both analytical and simulation, proposed up to now for evaluation ATM systems under this traffic has substantial shortcomings when cell losses in the order of 10^{-9} is required. This paper introduces a new trace driven simulation technique for cell losses in ATM buffers loaded by a large number of heterogeneous sources. Statistically firm results are obtained, within a reasonable computation effort/time, by applying a special importance sampling approach. The properties of the technique is examined and compared to a previously suggested stratified sampling technique. The capabilities of the technique is demonstrated by simulation of 76 sources of nineteen different MPEG VBR video source types with cell losses in the 10^{-9} - 10^{-12} domain.

1. This is a joint work with Dr. R. Andreassen and Prof. B. E. Helvik, conducted during the winter 95-96. This chapter is a *re-formatted reprint* of [AHH96]. Some of the results are also published in [And97]. Observe that the notation and concepts are *not consistent* in every aspect with the rest of the thesis. However, the appendix is self-contained and at the beginning of this paper a complete list of notation is given.

A.1 Introduction

A substantial part of the information that will be transferred in the coming ATM based high capacity networks will be related to moving pictures/video. This will stem from a wide range of applications for entertainment, e.g. video on demand, and professional use, e.g. computer supported cooperative work (CSCW). With its ability for low cost, real time, variable bitrate transfer, ATM is the enabling technology for this medium. MPEG is the broadly accepted standard for transfer of moving pictures/video [LG91, Tud95]. Due to how spatial and temporal redundancy in the pictures are removed in the MPEG encoding and how interpolation is carried out, see section A.2, the resulting information flow is highly variable, changes abruptly at frame borders and has a certain periodicity. Furthermore, it has, as all VBR encoded video streams, large medium and long term autocorrelation. All of these properties makes it difficult to perform a stochastic multiplexing with a calculated trade-off between a (high) channel utilization and information/ cell loss.

With traffic from MPEG encoded sources as a substantial constituent of the overall traffic in the network, it is of major importance to handle this trade off. Analytical models exist, which will give reasonable results in the special cases of homogeneous frame-synchronised sources [AER95, And96]. However, homogeneity is an unrealistic simplifying assumption, and as the number of sources increases, the assumption of frame synchronisation will give far too pessimistic results. Considering simulation, several publications describe parametric statistical simulation models, which in certain respects and to various degrees reproduce properties of real sources [HDMI95, JLS96, KSH95, RMR94]. While such models offer insight into the statistical nature of video sources, they have a limited ability to encompass all the aspects of a real MPEG source. Hence, our work is based on trace driven simulations, i.e. the information transfer requirements from real MPEG sources are used as input, obtaining more precise results.

Anyhow, at the quality of service (QoS) levels aimed at in ATM networks, direct simulation will fail to produce valuable results, irrespective of whether it is model based or trace driven. This is due to the scarcity of ATM cell losses. Direct simulation of load scenarios yielding a cell loss rate in the order of 10^{-9} and less, requires years of simulation time before a firm statistical basis is obtained. Simulation speed-up techniques are mandatory.

For trace driven simulations, this line of attack was started by using the method of stratified sampling as presented in [AER95, And96]. That approach, however, is restricted to

homogeneous sources and suffers from ‘state explosion’ as the number of sources increases. Another general speed-up approach for source model based ATM simulation and measurements is presented in [HH94]. This approach utilizes a combination of importance sampling and control variables. As pointed out above, model based simulation fails to reflect the properties of MPEG sources sufficiently accurate. Furthermore, for stable results, also this approach requires rather homogeneous sources.

In the current work, a new trace driven simulation model based on importance sampling (IS) is presented. This approach enables assessments of low cell loss probabilities, it does not require frame alignments and is well suited for load scenarios with heterogeneous MPEG video sources. A presentation is given in section A.3. A comparison of this and the approach based on stratified sampling is found in section A.4. In section A.5 use of importance sampling based, trace driven simulation is demonstrated before section A.6 concludes the paper. First, however, a brief introduction to MPEG coding necessary for the rest of the paper, is given in section A.2.

A.2 MPEG coding

For a general introduction of MPEG coding principles, it is referred to the available literature, e.g. [LG91, Tud95]. However, mainly to introduce the terms used, a very brief introduction to some MPEG coding concepts is given here. In MPEG compression, both spatial and temporal redundancy is removed. The spatial redundancy is reduced by using transforms and entropy coding, and the temporal redundancy is reduced by motion compensation and differential information encoding. The latter mechanism is enabled by the definition of three different types of frames:

I-Frames: Intraframes, only intra-frame encoding is used, i.e. only spatial redundancy is removed. I-frames are typically the largest.

P-Frames: Predicted frames; temporal redundancy is removed by reference to the previous I or P frame. P-frames are typically the second largest.

B-Frames: Interpolated or bidirectionally predicted frames; temporal redundancy is removed with reference to the previous and/or subsequent I or P frame. B-frames are typically the smallest.

The frame types are arranged in a systematic and periodic manner to form groups of pictures (GOPs).

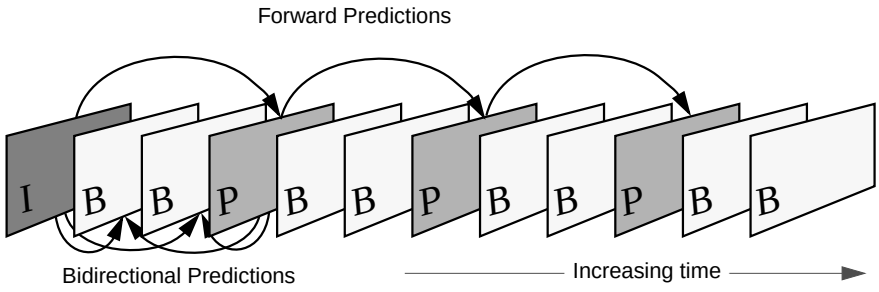


Figure A.1: Example of the basic structure of an MPEG encoded video sequence.

As illustrated in figure A.1, a GOP is headed by an I-frame and the alterations of B and P frames form sub-cycles in the periodic structure. A fairly usual GOP-pattern (which is also employed by the sources used in the current work) is ". . . **I**BB**P**BB**P**BB**P**BB . . .". It is seen that MPEG traffic has an activity at the burst level which is governed by frames of a constant duration. This is common to most video sources. The major difference lies in that the frames which follow each other in the periodic GOP are of a different kind with a basically different information content, and a smooth transition over the frames cannot be assumed.

An important coding parameter of the MPEG coding algorithm is the quantization parameter (q), which basically regulates the degree of information loss, and hence the coarseness of the decoded pictures. Active use of the q-parameter during encoding may be employed as a flow regulation mechanism. In the sequences used here, fixed quality video is investigated, and the q-parameter is set at a constant value for each frame type. To exploit the nature of frame referencing, the I-frames that start the GOP is given the finest encoding, the P-frames a medium degree encoding, and the B-frames that are not further referenced are given the coarsest encoding.

A.3 Importance sampling for heterogeneous MPEG sources

A.3.1 General idea

Direct simulation of the cell loss ratio of a multiplex of MPEG sources, as illustrated in figure A.2, becomes very inefficient when the ratio becomes small (typically 10^{-7} and less). *Importance sampling* is introduced to increase this efficiency by increasing the number of cell losses, i.e. to make multiplex-patterns where overloads are more likely. A straight forward heuristics is to sample a starting frame position for each source *according to the relative load* of each (I, B or P) frame of the film sequence. In addition, a *load selection* of the allowed starting frame positions is made in order to ensure that the total number of cells generated at the starting frame position exceeds the server capacity.

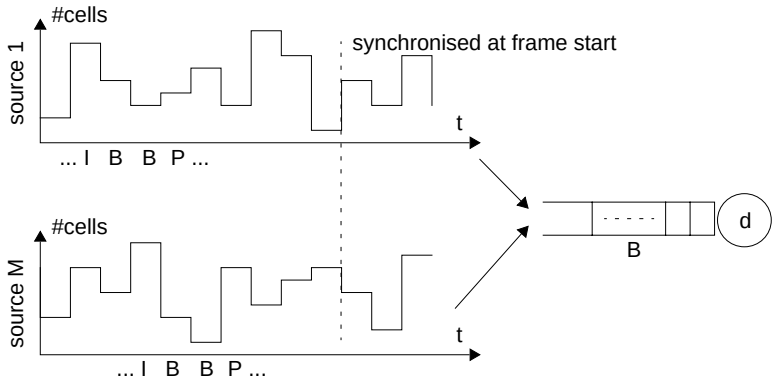


Figure A.2: MPEG multiplexing.

This section will describe these ideas in more details. First, some basic concepts and notations are introduced and importance sampling described. For simplicity, the speed-up strategy is described for homogenous sources with synchronised film-sequences. Finally, generalisations to heterogeneous and synchronised MPEG sources are given.

A.3.2 Basics and notation

Before the importance sampling strategy is introduced the basic MPEG multiplexing simulation set-up is described, see figure A.2. Necessary notation and basic concepts are also included. Note that for notation simplicity, only the special case of homogenous sources

and synchronised film sequences are described here. This means that the frames arrive simultaneously to the multiplexers of figure A.2, and that they all stem from the same film-sequence. In section A.3.4.2 and A.3.7, generalisations are described.

A.3.2.1 General notations

N Number of frames in a sequence.

M Number of sources.

$a \oplus b = (a + b - 1) \bmod(N) + 1$ (the modulo N addition).

$\underset{\sim}{1} = \{1, 1, \ldots, 1\}$ (an identity-vector, implicitly of size M).

$$I(x) = \begin{cases} 1 & \text{if } x>0 \\ 0 & \text{otherwise} \end{cases} \quad \text{(indicator function)}$$

A.3.2.2 Trace driven simulation

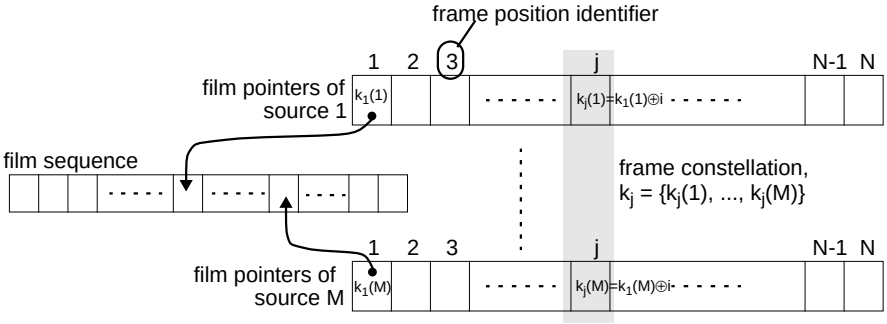


Figure A.3: Basic concepts related to trace driven simulation of multiplexed homogeneous MPEG sources.

The simulation of multiplexed MPEG sources is done by sampling a starting position in the film sequence for each source i, in sequence from 1 to M. This ordered set of starting positions, and generally a synchronised or aligned frame-positions, are denoted a *frame constellation,* $\underset{\sim}{k}$. The system response Y is the number of cell losses which are determin-

istically given by tracing through the N frames in the multiplex of film sequences starting each source at the position given by $\underset{\sim}{k}$.

$\underset{\sim}{k} = \{k(1), \ldots, k(M)\}$ Frame constellation (ordered set)

$\underset{\sim}{k}_j = \underset{\sim}{k}_1 \oplus j \cdot \underset{\sim}{1}$ The frame constellation at frame position identifier j (see figure A.3).

$Y(A)$ System response of A.

$\underset{\sim}{X}_{\underset{\sim}{k}} = \{X_{k(1)}, \ldots, X_{k(M)}\}$ Number of cells in frames given by constellation $\underset{\sim}{k}$.

$X_{\max} = \max_{k \in \{1, N\}} \{X_k\}$ Maximum number of cells in a frame.

$J(x) = \{i | (X_i > x)\}$ Set of frame positions having more than x cells.

d Number of cells served during a frame period (excl. buffer-length).

B Buffer capacity.

A.3.2.3 Concept of alignments

The sampling of a starting frame constellation can be viewed as a sampling of a film sequence *alignment*, because the relative position between the M sources is constant throughout the entire sequence. This means, that the same alignment can be sampled as a result of sampling any of the frame constellations this alignment consists of. However, the system response will, due to an initial transient caused by buffers, for high loads be slightly dependent on the starting frame constellation.

An alignment constituted by the specific frame constellation $\underset{\sim}{k}$ is then:

$$A_l^{(\underset{\sim}{k})} = \{\underset{\sim}{k}, \underset{\sim}{k} \oplus \underset{\sim}{1}, \ldots \underset{\sim}{k} \oplus (N-1)\underset{\sim}{1}\} \tag{A.1}$$

and $A_l = A_l^{(\underset{\sim}{k}_i)}$, for all $i = 1$ to N, is the alignment number l with either (non-specified) frame constellation in A_l as the starting constellation. Observe that N different rotations of the alignment depicted in figure A.3 will result in the same alignment. Whenever a reference to the starting frame constellation is needed, the index refers to the first source, called *frame position identifier* as indicated in figure A.3.

Note that even if there exists several *permutations* of the order of MPEG sources (a source corresponds to a vector of film-sequence pointers in figure A.3), and that each of them will give identical response, it is, however, essential that each permutation constitutes a unique alignment. This assumption is necessary to make a simple sampling algorithm, see section A.3.6.

A.3.2.4 Alignment probabilities

The probability of sampling the alignment A_l with an offset position j relative to the *frame position identifier* is denoted $P(l, j)$. Hence, the probability of an alignment A_l is $P(l) = \sum_{j=1}^{N} P(l, j)$. The original sampling distribution is uniform, i.e.

$$P(l) = \sum_{j=1}^{N} P(l, j) = \sum_{j=1}^{N} \frac{1}{N^M} = 1/N^{M-1} \tag{A.2}$$

System response is obtained under the assumption of a deterministic intra-frame cell arrival process. With this assumption, and noting that a large number of cells will arrive during a frame epoch, response is determined by multiplexer states at frame arrival instants. If service before cell arrival is assumed, the following recursion relation apply:

$$n_j = \mathrm{Max}(\mathrm{Min}(n_{j-1} + (\underset{\sim}{X}_{k_j} \cdot \underset{\sim}{1} - d), B-1), 0) \tag{A.3}$$

Hence, the system response is ($n_0 = 0$):

$$Y(l) = \sum_{j=1}^{N} \mathrm{Max}(n_{j-1} + (\underset{\sim}{X}_{k_j} \cdot \underset{\sim}{1} - d) - (B-1), 0) \tag{A.4}$$

with expectations $E(Y) = \sum_{\forall l} Y(l)P(l)$. Hence, an unbiased estimator for n direct simulation experiments is:

$$\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y(l_i) \tag{A.5}$$

where the alignment A_{l_i} is sampled according to the uniform distribution of (A.2).

A.3.3 Importance sampling fundamentals

Importance sampling have been used with success to yield speed-up in rare event simulation, see [Hei95] for an excellent overview. In MPEG simulation cell losses are *rare events* which will require extremely long simulation periods to obtain stable estimates.

The theoretical fundamentals of importance sampling can shortly be described by the following. Consider Y as an observation of the quantity of interest to be a function $g(X)$ where X is sampled from $f(X)$. Non-zero values of Y are rarely observed in a direct simulation. The basic idea is simply to change the underlying sampling distribution to $f^*(X)$ to make Y more likely to occur and where the following relation hold:

$$E_f(Y) = E_f(g(X)) = E_{f^*}(Y \cdot f(X) / f^*(X)) = E_{f^*}(Y \cdot \Lambda(X)) \tag{A.6}$$

where $\Lambda(X)$ is the likelihood ratio between $f(X)$ and $f^*(X)$. Thus, the property of interest Y can be estimated by taking R samples from $f^*(X)$, accumulate $\Lambda(X)$, and use the following unbiased estimator:

$$\bar{Y} = \frac{1}{R} \cdot \sum_{r=1}^{R} Y_r \cdot \Lambda(X_r) \tag{A.7}$$

The main challenge is to choose a new distribution, f^*, that minimizes the variance to this estimator, $\bar{Y}$. If an unsuited distribution is used, it is observed that simulation is inefficient and is producing inaccurate results, see e.g. [DT93].

In MPEG simulation let X be an alignment A_l of film sequences originally sampled from a uniform distribution, $f(x)$, and let $Y = Y(l)$ be the number of cells lost, see (A.4). The following section will discuss heuristics specific for the MPEG sources which determine the new sample distribution $f^*(X)$.

A.3.4 Changing the sampling distribution

A.3.4.1 Heuristics

In a trace driven simulator for MPEG sources, the alignments are sampled according to a uniform distribution. When a large number of frames in the sequence have few cells, and/

or the overall mean load are low, this will result in an enormous number of $Y_l = 0$ observations.

Importance sampling seeks to reduce this problem by increasing the frequency of alignments having $Y > 0$. This objective is achieved by changing the $f^*(X)$ by the following heuristics:

1. *Load distribution*: Sample the starting (frame) position of each of the multiplexed sources proportional to its load, instead of the uniform distribution, see figure A.4. This will provoke the heavy load frames to coincide in an alignment.

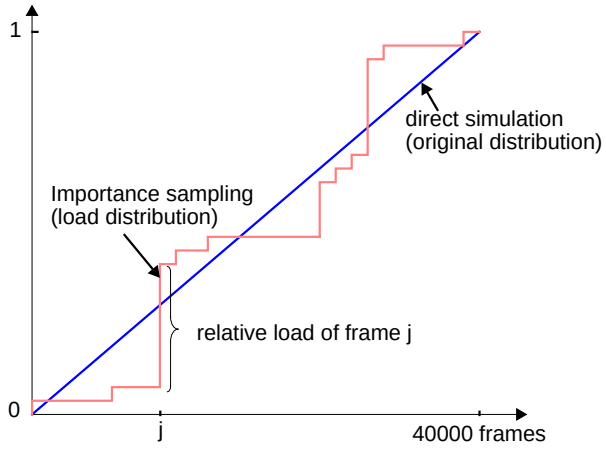


Figure A.4: Cumulative load distribution.

2. *Load selection*: The conditional starting position of source i are restricted to those which makes an overload (and hence cell loss) feasible.

A.3.4.2 Alignment probabilities

Because an alignment is selected through sampling of one of its frame constellations, the alignment probability $P(l)$ is as the sum of frame constellation probabilities, $P(l, j)$.

Let $J(x) = \{i | (X_i > x)\}$ be the set of frame positions having more than x cells. Then, the change of frame constellation probabilities to $P^*(l, j)$, according to the heuristics from section A.3.4.1, can be expressed as:

In sequence i=1 to M do[1]:

$$\begin{aligned} &\text{Let } x_i = d - \sum_{m=1}^{i-1} X_{k_j(m)} - (M-i)X_{\max} \\ &P_i^*(l, j) = (X_{k_j(i)} \cdot I(x_i)) / [\sum_{m \in J(x_i)} X_{k_j(m)}] \end{aligned} \tag{A.8}$$

and finally

$$P^*(l, j) = \prod_{i=1}^{M} P_i^*(l, j) \tag{A.9}$$

This operation is $O(N)$-complex and is the most critical operation in the algorithm with respect to computer efficiency. However, if the film-sequence is sorted by frame size in decreasing order, and the frame references to the original unsorted sequence are known, the evaluation can be reduced to a $O(1)$-complex operation, see [And97].

Eq. (A.8) can easily be generalised to heterogeneous sources. The number of cells at frame position i, X_i, must be generalised to $X_i^{(f)}$ where the new index (f) refer to the film-sequence type. This is simply substituted into (A.8). However, to avoid the frame constellation sampling to be biased, the sequence of which sources are sampled, must be randomly ordered[2].

A.3.5 The loss likelihood

The likelihood ratio is the ratio between the alignment probabilities in (A.2) and (A.9):

$$\Lambda(l) = \frac{P(l)}{P^*(l)} = \frac{\sum_{j=1}^{N} P(l, j)}{\sum_{j=1}^{N} P^*(l, j)} = \frac{1/N^{M-1}}{\sum_{j=1}^{N} P^*(l, j)} \tag{A.10}$$

A.3.6 Algorithm

The complete algorithm can be described as follows:

Requirement: $MX_{\max} \geq B$

1. The frame space is reduced for every source, dependent on the accumulated load of the frames sampled by the previous sources, and the maximum load which is possible to obtain by the remaining sources.
2. In the homogeneous case, this sequence is fixed, i.e. the frame position of source 1 is always sampled first, then 2, and finally source M.

```
/* repeat the following to obtain frame constellation probabilities */
For i = 1, …, M {
        k(i) ← P_i*(l, j) from (A.8)
}
P*(l, j) = ∏_{i=1}^{M} P_i*(l, j)
/* sample an offset from k to reduce the effect of the buffer transients */
U ← U(1, N)
/* add this offset to all sources */
k = k ⊕ (U · 1)
/* tract all N frame positions in the sampled alignment */
For j = 1, …, N {
        if loss at k_j > 0 {
                /* update system response, see (A.4) */
                Y = Y + loss at k_j
                /* update alignment probability according to (A.8) */
                P*(l) = P*(l) + P*(l, j)
        }
}
/* calculate the loss likelihood according to (A.10) */
Λ(l) = P(l)/P*(l)
/* update statistics of the observation Y · Λ(l) */
```

This algorithm is repeated R times, and $\bar{Y}$ is estimated by (A.7).

A.3.7 Non-synchronized sources

In establishing equation (A.3), the assumption was made that frames from different sources arrive simultaneously at the multiplexer. When this assumption is removed, each source will in addition to the frame starting point also be associated with a specific frame phase. Such a generalisation will influence both calculations of system response and likelihood ratio.

A.3.7.1 Revised multiplexing model

System response may still be determined by system states at frame arrival instants, but now frame arrivals are spread throughout a frame epoch according to the frame phases of sources. Hence, the following generalisation of the recursion in (A.3) applies:

$$n_j = \mathrm{Max}(\mathrm{Min}(n_{j-1} + \frac{d_j}{d} \cdot \underset{\sim}{X}_{\underset{\sim}{k}_j} \cdot \underset{\sim}{1} - d, B - 1), 0) \qquad \text{(A.11)}$$

where d_j is the number of cells served during the j'th fixed rate interval.

A.3.7.2 Revised likelihood ratio

Consider that when multiplexing video sources, the smoothest compound source is obtained when frame phases are evenly spaced in the frame period. In this scenario, each of the M sources may occupy one of M distinct and different frame phases, so in a sequence of length N, there will then be $N \cdot M$ discrete starting points which can be chosen in $M! \cdot N^M$ ways. The uniformly distributed probability of choosing any alignment is then given by

$$P(l) = \frac{N \cdot M}{M! \cdot N^M} = 1/[(M-1)! \cdot N^{M-1}] \tag{A.12}$$

The probability of choosing a specific phase-constellation is $1/M!$, so the biased alignment probability may be expressed as

$$P^*(l) = \frac{1}{M!} \sum_{j=1}^{N \cdot M} P^*(l, j) = \frac{1}{M!} \sum_{j=1}^{N \cdot M} \prod_{i=1}^{M} P_i^*(l, \left\lfloor \frac{j + f_i}{M} \right\rfloor) \tag{A.13}$$

Here f_i denotes the phase of source i. The likelihood ratio of the l'th alignment will then be:

$$\Lambda(l) = \frac{M/N^{M-1}}{\sum_{j=1}^{N \cdot M} P^*(l, j)} \tag{A.14}$$

Letting the number of discrete phase values increase, a Riemann sum can be formed such that the likelihood ratio in the limit of continuously varying frame phases can be expressed as

$$\Lambda(l) = \left(N^{M-1} \frac{1}{T_f} \int_0^{NT_f} P^*(l, t) dt \right)^{-1} = \left(N^{M-1} \sum_i P^*(l, t_i) \frac{d_i}{d} \right)^{-1} \tag{A.15}$$

Here T_f denotes the frame duration, the sum in the last term is performed over all fixed-rate intervals in the simulation sequence and d_i/d is the length of the i'th fixed-rate interval relative to the frame duration. It should be noted that the complexity of calculations for unsynchronised sources is a factor M higher than for unsynchronised sources.

A.4 Speed-up techniques for homogenous MPEG sources

In previous work [AER95], an alternative speed-up technique for MPEG simulation was applied, which was based on the use of *stratified sampling*, see e.g. [LO88] for an introduction.

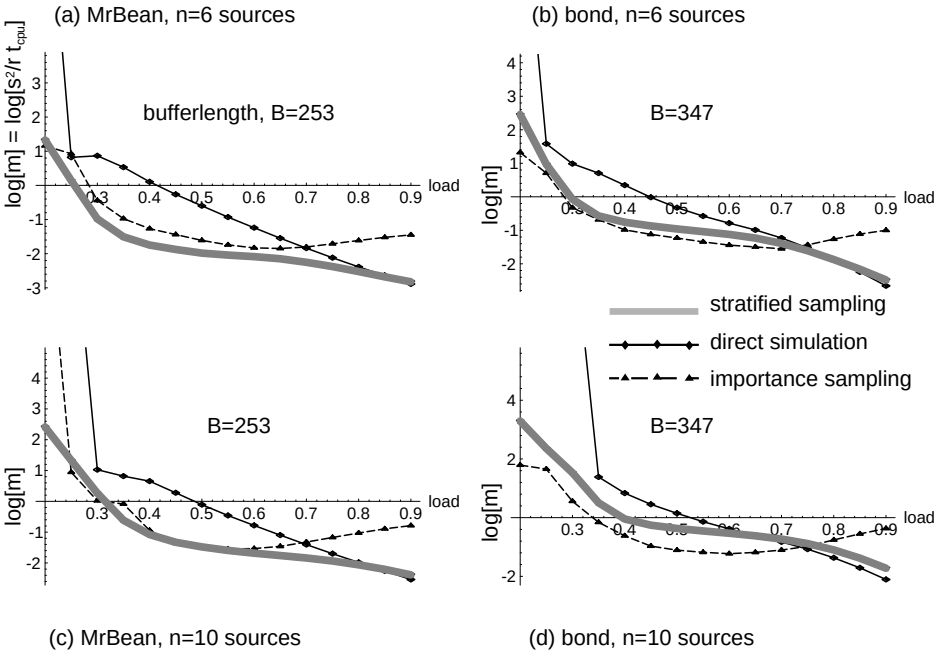


Figure A.5: Comparison of direct simulation, importance sampling and stratified sampling with respect to the efficiency measure. Note that all values are in the logarithmic scale.

This section compares the importance sampling with stratified sampling and the direct simulation approach for application to a multiplex of MPEG video sources. The comparisons are made based on an efficiency measure m, considering both the variability σ^2 and the CPU-time consumption t_{cpu}, over r experiments, see [Hee95]. Note that the most efficient technique will have the lowest measure[1].

$$m = \sigma^2 / r \cdot t_{\text{cpu}} \tag{A.16}$$

1. Observe that this efficiency measure is the reciprocal of the measure in (2.11).

The comparison was carried out for two film sequences, MrBean and Bond, having different characteristics, see table A.1. A representative sample of the results are presented in figure A.5 showing the efficiency measure in a logarithmic scale for the three approaches. The main observations are:

1. For high load values, direct simulation is always better than importance sampling and at least as good as stratified sampling. For high loads the are no rare events associated with the estimates, and the additional computer cost introduced by a speed-up technique is wasted.

2. For a small number of sources, stratified sampling is at least as good as importance sampling.

3. Importance sampling is better that stratified sampling even for a small number of sources of film-sequences having rather low maximum to mean ratio ($S_{\max}/S$) like the Bond sequence.

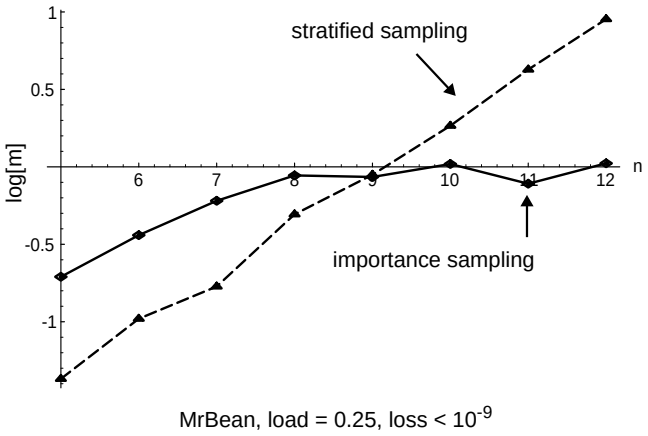


Figure A.6: Efficiency comparison as the number of sources increases.

The relative efficiency of importance sampling to stratified sampling are illustrated in figure A.6, using load of 0.25 on the MrBean film-sequence and varying the number of sources from n=5 to 12.

4. Importance sampling is better than stratified sampling when the number of sources become large.

As presented in [AER95], the stratified sampling model assumes synchronized and homogenous sources, while importance sampling does not have these restrictions. Hence, in the following section, importance sampling will be used to speed-up the simulation of heterogeneous sources

A.5 Multiplexing of heterogeneous MPEG sources

A collection of 19 sequences of diverse content will be used in the calculations to come. The statistics was produced at the University of Würzburg. Tematic content and some traffic characteristics of the traces are described in [Ros95].

A.5.1 Source characteristics

The material was captured via analog VHS video-format, and can be characterized as fairly low quality / low bitrate video. The sequences are 40,000 frames long, the picture format 288 lines by 384 pels, and the quantization parameter triplet is given by q_i, q_p, q_b = 10, 14, 18. Table A.1 summarizes some static traffic characteristics of sources.

In the table, S_{max}, S, v, refers to sample maximum, mean and coefficient of variation respectively. Indices I, P, B refers to the sequences formed by partitioning the original sequence after frame types. In the second last column is shown the relative sizes of I-frames versus P and B frames, which may be taken as a measure of the success of temporal redundancy removal in the different sequences, see section A.2. The unit of the numbers is a cell, where a cell payload of 44 bytes is assumed.

What is immediately obvious from the above table is the great diversity of statistical properties of the sources. The sources are similar in that they use the same coding parameters and picture format, and were retrieved in the same way from analog video tape. So, even as the sequences contained the same amount of data before the coding took place, the outcome varies widely both with respect to mean bitrate, burstiness (peak/mean-rate) and variability. It may also be seen that the relative sizes of I-frames versus PB-frames vary from a factor three to about a factor eight, reflecting the various outcomes of temporal redundancy removal. Analyses of temporal properties of selected sources [And96,Ros95] show source diversities also in the temporal domain. No single source or source model can reasonably reflect the above diversity of properties, so the MPEG multiplexing performance analyses based on single source types will lack in statistical significance.

Table A.1: Non-temporal statistical characteristics of sources.

Source	S	v	S_{I}	v_{I}	S_P	v_{P}	S_{B}	v_{B}	S_{PB}	$S_{\mathrm{I}}/S_{\mathrm{PB}}$	S_{max}/S
MrBean	50.1	1.17	213.5	0.26	51.9	0.78	29.0	0.66	35.3	6.1	13.0
Asterix	63.5	0.90	202.0	0.27	77.1	0.60	41.1	0.67	50.9	4.0	6.6
Atp	62.2	0.93	215.1	0.29	75.8	0.55	38.0	0.48	48.3	4.5	8.7
Bond	69.1	1.06	236.6	0.31	117.7	0.49	29.9	0.40	53.8	4.4	10.1
Dino	37.2	1.13	156.5	0.21	41.1	0.67	20.8	0.61	26.3	5.9	9.2
Fuss	77.1	0.96	224.9	0.32	135.0	0.48	36.9	0.48	63.6	3.5	6.9
Lambs	20.8	1.53	108.0	0.34	21.1	1.08	9.7	0.93	12.8	8.4	18.4
Movie2	40.6	1.32	163.8	0.37	66.8	0.74	15.4	0.84	29.4	5.6	12.1
Mtv	69.9	0.94	198.5	0.35	111.7	0.55	38.2	0.69	58.2	3.4	9.3
Mtv_2	56.2	1.08	174.4	0.41	71.1	0.86	35.8	0.99	45.4	3.8	12.7
News	43.6	1.27	200.6	0.30	43.9	0.82	23.9	0.60	29.4	6.8	12.4
Race	87.4	0.69	225.1	0.26	108.5	0.48	62.2	0.46	74.8	3.0	6.6
Sbowl	66.8	0.80	193.0	0.28	88.9	0.47	42.7	0.51	55.3	3.5	6.0
Simpsons	52.8	1.11	210.4	0.26	61.2	0.73	29.9	0.67	38.4	5.5	12.9
Soccerwm	71.3	0.85	201.5	0.36	93.8	0.58	46.6	0.61	59.5	3.4	7.6
Star2	26.5	1.39	125.0	0.32	28.8	0.94	13.3	0.91	17.5	7.1	13.4
Talk_2	50.9	1.02	209.6	0.18	50.8	0.46	31.1	0.32	36.5	5.7	7.4
Talk	41.3	1.14	183.9	0.16	42.1	0.59	23.2	0.43	28.3	6.5	7.3
Term	31.0	0.93	106.2	0.22	40.1	0.51	18.1	0.60	24.1	4.4	7.3

A.5.2 Simulation experiments

In the simulation experiments, the number of simultaneous sources was regulated by multiplying the number of each source type, giving multiplexing scenarios with multiples of 19 sources. Each calculation is based on 5000 independent simulations, and in figures, error-bars indicate the obtained 95% confidence intervals. Simulation runs lasted from some hours for the lowest loss probabilities / least number of sources, up to several days per point for the largest number of sources and the highest loss probabilities. We have concentrated on multiplexers with moderately sized buffers.

Results from the simulation experiments are presented in figure A.7.

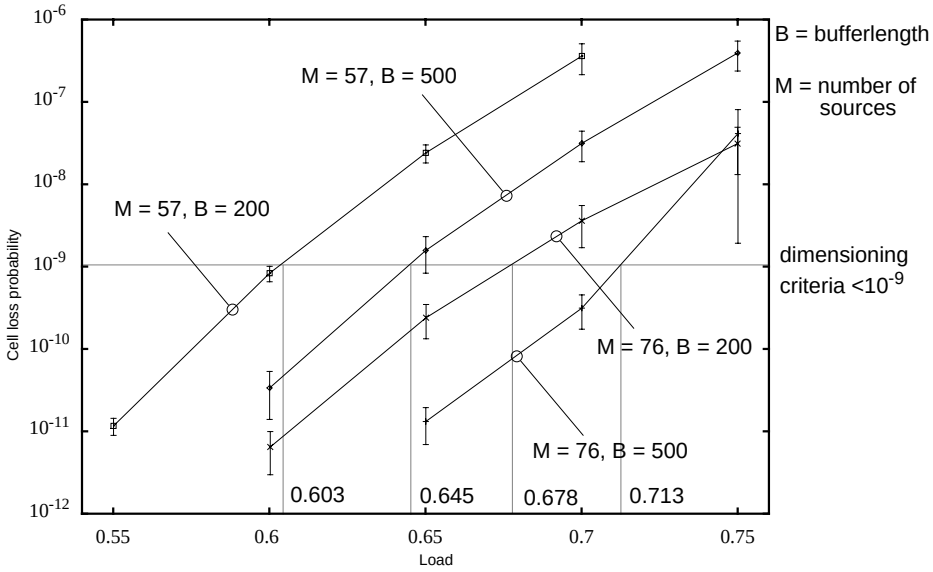


Figure A.7: Loss probabilities of multiplexed sources, 95% confidence intervals.

A frequently cited QoS objective for ATM-networks is a cell loss probability of 10^{-9}. Hence, results in figure A.7 are presented for the region between 10^{-12} and 10^{-6}. Even though these probabilities are low from a networking perspective, it should be noted that these numbers are very high compared to the absolute minimum loss probability of the multiplexed sources, occurring if all sources send their maximum sized frame simultaneously, i.e. giving a loss probability of less than $1/40.000^{M}$. Our calculations are based on several thousand non-zero observations, and have a high level of confidence.

In [KSH95] and [RMR94] considerable statistic multiplexing gains for MPEG sources were reported. These analyses were based on of single MPEG source types in high loss regions. From the rightmost column of table A.1, it can be seen that burstiness of sources vary between 6 and 18, so by the methods presented here, we have been able to confirm that even for the low loss probability regions of heterogeneous multiplexed sources, there is a good potential for statistical multiplexing.

As load increases to certain levels, the sampling bias will tend to destabilize the system, giving poor confidence as a result. This may be observed for the highest loss value of M=76, B=500. For high loss probabilities, direct simulation will, however, be possible.

As number of sources increases, obtaining good result will be correspondingly more difficult, and the multiplexing 76 sources represents what we estimated as the maximum reasonable complexity with the current choice of parameters. The method is sensitive to buffer lengths, because the *load selection* algorithm does not include buffer when irrelevant samples are excluded. Hence, analysing a larger number of sources should be feasible for shorter buffer lengths.

Considering the multiplexing scenarios, the link rates necessary to accommodate the different number of sources at a loss probability objective of 10^{-9} is calculated by linear interpolation and presented in table A.2. In the calculations, a video frame rate of 25 frames per second is assumed.

Table A.2: Multiplexer dimensioning

Number of sources	Compound source rate	Link rate, B = 200	Link rate, B = 500
M = 57	32.4 Mbit/s	53.7 Mbit/s	50.2 Mbit/s
M = 76	43.2 Mbit/s	63.7 Mbit/s	60.6 Mbit/s

The low bitrate / low quality of the sources allows a fairly large number of sources to be multiplexed even at moderate link rates, thus giving reasonable levels of utilisation. Acquiring similar data for other video qualities will require further investigations.

A.6 Conclusions

Assessing ATM multiplexing performance at the lower cell loss objectives have for a long time been a general problem. There are several factors contributing to the difficulty of such calculations. Many data sources, and MPEG video sources specifically, are of a complex statistical nature, and will not easily yield to analytical analysis. Using parametric simulation models, there is the problem of capturing essential source properties and the added problem of obtaining results of significance for the rarely occurring events. Both in simulation modelling and analytical analysis, there is the problem of modelling the diversity of video sources.

In the current work, we have described a method that will solve the above problems. We have based our method on using the statistical material in a direct manner, and in obtaining good results, the availability of sufficient amount of statistical material is crucial. As MPEG coders become more readily available, the availability of statistical material will

be abundant, and the method may be applied to any desired mix of sources. Although the current focus has been on MPEG sources, the method is applicable for all sources with a constant duration framing property.

As no known methods exist for obtaining optimal biasing strategies for the importance sampling calculations, the employed sampling heuristics uses the knowledge of the characteristics of MPEG source to provoke overloads (and cell losses). The heuristics are based on a combination of likely contributions to cell losses from the individual frames, and a systematic exclusion of irrelevant samples.

The usefulness of the method is demonstrated by giving, to the authors knowledge, the most reliable MPEG multiplexing performance data published to date.

Acknowledgements

The authors would like to thank Dr. Oliver Rose at the University of Würzburg for making the video traces available. The traces used here were obtained by anonymous ftp from: ftp-info3.informatik.uni-wuerzburg.de:/pub/MPEG/

References

[AER95] Ragnar Ø. Andreassen, Peder J. Emstad, and Tore Riksaasen. Cell losses of multiplexed VBR MPEG sources in an ATM-multiplexer. In Norros and Virtamo [NV95], pages 83–95.

[And96] R. Ø. Andreassen. Correlation properties of MPEG VBR video sources: Relative importance of short and medium term correlation on ATM multiplexer performance. In *IFIP/IEEE Broadband Communications'96*, pages 40–52, 1996.

[And97] Ragnar Andreassen. *Traffic performance studies of MPEG variable bitrate video over ATM*. PhD thesis, Norwegian University of Science and Technology, June 1997.

[DT93] Michael Devetsikiotis and J. Keith Townsend. Statistical optimization of dynamic importance sampling parameters for efficient simulation of communication networks. *IEEE/ACM Transactions on Networking*, 1(3):293 – 305, June 1993.

[HDMI95] C. Huang, Devetsikiotis M., Lambadaris M., and A. R. I., Kaye. Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. In *ACM Sigcomm 95*, pages 114–125, 1995.

[Hee95] Poul E. Heegaard. Comparison of speed-up techniques for simulation. In Norros and Virtamo [NV95], pages 407–420.

[Hei95] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM transaction on modeling and computer simulation*, 5(1):43–85, January 1995.

[HH94] Bjarne E. Helvik and Poul E. Heegaard. A technique for measuring rare cell losses in ATM systems. In Labatoulle and Roberts [LR94], pages 917–930.

[JLS96] P R. Jelencovic, A A. Lazar, and N. Semret. Multiple time scale sand subexponentiality in MPEG video streams. In *IFIP/IEEE Broadband Communications'96*, pages 64–75, 1996.

[KSH95] M. Krunz, R. Sass, and H. Hughes. Statistical characteristics and multiplexing of MPEG streams. In *IEEE Infocom 95*, pages 455–462, 1995.

[LG91] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34:47–58, 1991.

[LO88] P. A. W. Lewis and E. J. Orav. *Simulation Methodology for Statisticians, Operations Analysts and Engineers*, volume I. Wadsworth & Brooks/Cole Advanced Books & Software, 1988.

[LR94] J. Labatoulle and J.W. Roberts, editors. *The 14th International Teletraffic Congress (ITC'14)*, Antibes Juan-les-Pins, France, June 6-10 1994. Elsevier.

[NV95] Ilkka Norros and Jorma Virtamo, editors. *The 12th Nordic Teletraffic Seminar (NTS-12)*, Espoo, Finland, 22 - 24 August 1995. VTT Information Technology.

[RMR94] D. Reininger, B. Melamed, and D. Raychaudhuri. Variable Bit Rate MPEG Video: Characteristics, Modeling and Multiplexing. In Labatoulle and Roberts [LR94], pages 295–306.

[Ros95] Oliver Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. Technical Report 101, University of Würzburg, Institute of Computer Science, February 1995. The traces obtained from: ftp-info.informatik.uni-wuerzburg.de /pub/MPEG.

[Tud95] P. N. Tudor. Mpeg-2 video compression. *IEE Electronics & communication engineering journal*, pages 257–264, Dec. 1995.

Appendix B

List of symbols, estimators and concepts

This appendix includes a list of symbols used in this thesis, and an overview over the modelling concepts described in chapter 4.

B.1 List of symbols

B.1.1 Property

γ property of interest,

$\hat{\gamma}$ original estimate of γ,

$\hat{\gamma}_{IS}$ importance sampling estimate of γ,

$\hat{\gamma}_{R}$ RESTART estimate of γ.

B.1.2 Distributions

$\underline{\omega}$ sample, e.g. sample path $\underline{\omega} = \{\omega_i\}_{i=0}^{n}$,

$g(\underline{\omega})$ observed property of interest,

$f(\underline{\omega})$ original sampling distribution,

$f^*(\underline{\omega})$ importance sampling distribution.

B.1.3 System state

$\underset{\sim}{\Omega}$ global state space, implicit of K dimensions, $\underset{\sim}{\Omega}^K$,

$\underset{\sim}{\omega} = \{\omega_k\}_{k=1}^{K}$ system state $\omega_k = \#e_k$, $\underset{\sim}{\omega} \in \underset{\sim}{\Omega}$,

$\underset{\sim}{\omega}_i$ system state $\underset{\sim}{\omega}$ after event i,

$\underset{\sim}{\Omega}_j$ target subspace j, ($\underset{\sim}{\Omega}_j \subset \underset{\sim}{\Omega}$, $j = 1, \ldots, J$),

$\underset{\sim}{\Omega}_0$ regenerative subspace, $(\underset{\sim}{\Omega}_0 \subset \underset{\sim}{\Omega}) \wedge (\underset{\sim}{\Omega}_0 \cap \underset{\sim}{\Omega}_j = \varnothing)$.

B.1.4 Building blocks

Resource pools

J number of resource pools,

N_j number of resources in pool j, (resource pool capacity),

Γ_j set of generators with pool j as a constraint.

Generators

K number of generators,

e_k entity from generator k,

Φ_k routing set, (a fixed set of resource pools),

$\lambda_k(\underset{\sim}{\omega})$ (state dependent) arrival rate,

M_k population size,

$\mu_k(\underset{\sim}{\omega})$ (state dependent) departure rate,

S_k number of servers for generator k entities,

c_{kj} capacity that entities from generator k requires from pool j,

p_k priority level of generator k, $p = 0$ is the highest priority level,

R_k number of rerouting alternatives for generator k.

B.1.5 Target subspace

$\underset{\sim}{\Omega}_j$ target subspace j expressed in terms of the system states,

B_j target subspace j expressed in terms of blocking positions in pool j,

$p_j(\underset{\sim}{\omega})$ target distribution ($j = 1, \ldots, J$), at specific state $\underset{\sim}{\omega}$,

$H_j(\underset{\sim}{\omega})$ importance of target j ($j = 1, \ldots, J$), at specific state $\underset{\sim}{\omega}$.

B.1.6 Simulation process

$\underset{\sim}{X}(t)$ state of continuous time Markov process at time t,

$\underset{\sim}{X}_i$ state of discrete time Markov process at embedded point i at time t_i,

$\underset{\sim}{Z}_i$ event variable describing possible next states of the process $\underset{\sim}{X}_i$.

B.1.7 Most likely path to a target

$\sigma_j(\underset{\sim}{\omega})$ the most likely subpath from current state $\underset{\sim}{\omega}$ up to $\underset{\sim}{\Omega}_j$.

n_j number of events in $\sigma_j(\underset{\sim}{\omega})$,

$\underset{\sim}{k} = \{k_1, k_2, \ldots, k_n\}$
sequence of generators constituting the subpath $\sigma_j(\underset{\sim}{\omega})$,

k_i generator of the i'th event in $\sigma_j(\underset{\sim}{\omega})$, $k \in \Gamma_j$,

x number of resource allocated,

x_0 initial number of resource allocated at current state $\underset{\sim}{\omega}$,

$\underset{\sim}{\omega}^{(x)}$ system state $\underset{\sim}{\omega}$ where the sum of allocated resources, $\sum_{k \in \Gamma_j} \omega_k c_{kj} = x$,

α_{xk} probability contribution from generator k to state x,

G_{xk} normalisation constant from generator k at state x,

G_x normalisation constant at state x,

π_x probability of allocation of x resources (from pool j).

B.2 Indexation

$\underset{\sim}{1}_k = \{0, \ldots, 1, \ldots, 0\}$
index vector of size K which is 1 at position k and 0 elsewhere.

X some property (of interest),

$X*$ property X under importance sampling distribution,

X_r rth sample of X,

X_k property X of generator k,

X_j property X of resource pool j,

$X^{(p)}$ property X at priority level p,

X_i property X after event i,

X_x or $X^{(x)}$
property X with x resources allocated.

B.3 Estimators

R number of replicas in a simulation experiment.

Sample mean:

$$\bar{X} = \frac{1}{R}\sum_{r=1}^{R} X_r$$

Sample variance:

$$S_X^2 = \frac{1}{R-1}\sum_{r=1}^{R}(X_r - \bar{X})^2$$

Standard error (of sample mean):

$$S_{\bar{X}} = \sqrt{S_X / R}$$

Relative error:

$$\text{r.e.}(\bar{X}) = S_{\bar{X}} / \bar{X}$$

B.4 Modelling concepts

Resource pool: a finite set of identical and interchangeable resources. The capacity of the pools is N_j, $j = 1, \ldots, J$.

Entity, e_k, $k = 1, \ldots, K$: an item that holds c_{kj} resources from pool j

Event: an occurrence that trigger a *request* or *release* of c resources.

Generator (of entities): a component which explicitly models processes generating the events that operates on the entities.

System state, $\underset{\sim}{\omega} \in \underset{\sim}{\Omega}$ ($\underset{\sim}{\Omega}$ is the global state space): a representation of the current number of entities at any time, i.e. $\underset{\sim}{\omega} = \{\omega_1, \ldots, \omega_K\}$ where $\omega_k = \#e_k$.

Path: any sequence of events $\underset{\sim}{s} = \{\underset{\sim}{\omega}_0, \underset{\sim}{\omega}_1, \ldots, \underset{\sim}{\omega}_{n-1}, \underset{\sim}{\omega}_n\}$, where $\underset{\sim}{\omega}_i$ is the system state after event i and n is the total number of events in path $\underset{\sim}{s}$.

Target, $\underset{\sim}{\Omega}_j$: a subset of $\underset{\sim}{\Omega}$ where the capacity N_j of resource pool j is exceeded.

Rare event: a visit to the target when $P(\underset{\sim}{\Omega}_j) \ll 1$, i.e. a visit is unlikely to happen.

Single target model: a model with only one (dominating) resource pool.

Multiple target model: a model with several pools with significant contributions to the quantity of interest.

Appendix C

Probability of the maximum state in a sample path

This appendix presents the details for the probabilities P_i that are used in section 2.7.4.

C.1 Settings

Let $\underset{\sim}{s}_r$ denote the rth regenerative simulation *cycle*, or a *sample path*, defined as $\underset{\sim}{s}_r = \{\underset{\sim}{\omega}_0, \underset{\sim}{\omega}_1, \ldots, \underset{\sim}{\omega}_{n_r}\}$. The probability of state i being the maximum state visited in $\underset{\sim}{s}_r$ is in (2.34) defined as:

$$P_i = Pr\{\max_{\forall r}(\underset{\sim}{s}_r) = i\} = Pr\{(\underset{\sim}{\omega}_x = i) \wedge (\underset{\sim}{\omega}_x \leq i), (x = 1, \ldots, n_r)\} \tag{C.1}$$

To derive P_i, the probability of visiting state i in $\underset{\sim}{s}$ is required:

$$Q_i = Pr\{\underset{\sim}{\omega}_x = i, (\exists x = 1, \ldots, n_r)\} \tag{C.2}$$

Explicit expressions for P_i and Q_i are presented in this appendix for an M/M/1/N queue. The Markov process $\underset{\sim}{X}(t)$ is described in section 3.2, and in a more general form in section 4.2. For an M/M/1/N queue, the event variable $\underset{\sim}{Z}_x$ at event x in the recursive expression in (4.2) is:

$$\underset{\sim}{Z}_x = \begin{cases} -1 & \text{with probability } \mu/(\mu + \lambda) \\ 1 & \text{with probability } \lambda/(\mu + \lambda) \end{cases} \tag{C.3}$$

for $x = 1, \ldots, n_r$. Initially, $\underset{\sim}{Z}_0 = 1$ with probability 1.

To derive Q_i, the *first step analysis* [CM65] is applied. The following equations are established for every state $i = 1, \ldots, N-1$ in the queuing model:

$$Q_i = \mu Q_{i+1} + \lambda Q_{i-1} \tag{C.4}$$

where $Q_0 = 1$ and $Q_i = 0$ for $i > N$. After some manipulations,

$$Q_i = \frac{\mu/\lambda - 1}{(\mu/\lambda)^i - 1} \qquad (i = 1, \ldots, N) \tag{C.5}$$

This probability is essential for the derivation of the probabilities P_i.

C.2 Direct simulation

The original distribution P_i is derived directly from (C.5). Recall from (C.2) that Q_i is the probability of visiting state i in $\underset{\sim}{s}$. P_i is the probability of state i is the *maximum state* in $\underset{\sim}{s}$, i.e. state i is visited in $\underset{\sim}{s}$ but not state $i + 1$:

$$P_i = \begin{cases} Q_i - Q_{i+1} & 0 \le i < N \\ Q_i & i = N \end{cases}. \tag{C.6}$$

C.3 RESTART

$P_i^{(R)}$ are the probabilities P_i in the *RESTART* distribution. To explain how $P_i^{(R)}$ are determined, recall the *basic regenerative cycle* (or sample path) from section 2.7.1. This is the path that consists of a sequence of events that starts from, and ends in, the renewal state Ω_0 and contains no splitting of the simulation process. This is the simulation process under the original distribution. The maximum probabilities in the basic path is the P_i from (C.6). This means that for all maximum states i below the first threshold state the maximum probabilities for optimal and RESTART distributions are equal, except for a normalisation constant that is introduced later. For maximum states i above the first threshold, the number of replications will influence the $P_i^{(R)}$. In figure C.1 an example is plotted including two intermediate thresholds states. At threshold 1 at state 4, R_1 replications of the simulation process are made from this threshold state. This implies that the probability of a maximum state above the threshold state are increased by a factor of R_1. The same is repeated at every threshold state.

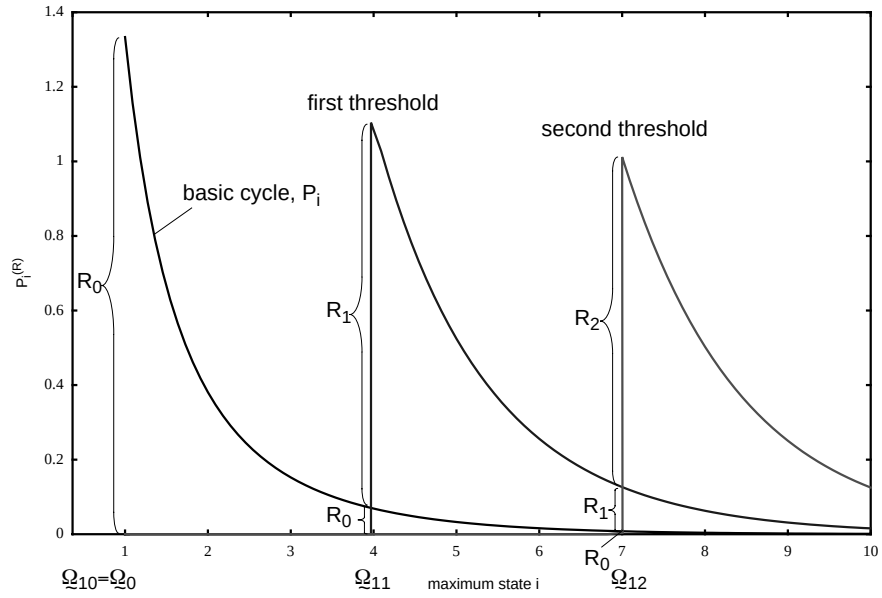


Figure C.1: The maximum state probabilities in the RESTART distribution is a product of the probabilities in the original distribution, P_i, and the number of replications made at every threshold.

Let $\underset{\sim}{\Omega}_{1j}$ be the j th threshold state with respect to target subspace $\underset{\sim}{\Omega}_1$. The number of replications at every threshold $m = 0, \ldots, M$ is denoted R_m. The probabilities can then be expressed as:

$$P_i^{(\mathrm{R})} = P_i / G \cdot \begin{cases} R_0 & \Omega_{10} \le i < \Omega_{11} \\ R_0 \cdot R_1 & \Omega_{11} \le i < \Omega_{12} \\ R_0 \cdot R_1 \cdot R_2 & \Omega_{12} \le i < \Omega_{13} \\ \ldots & \ldots \end{cases} \tag{C.7}$$

or in a more compact notation

$$P_i^{(\mathrm{R})} = P_i / G \cdot \prod_{l=0}^{m} R_l \qquad (\Omega_{1m-1} \le i < \Omega_{1m}) \wedge (1 < m < M) \tag{C.8}$$

where the normalisation constant G satisfies $\sum_{i=0}^{N} P_i^{(R)} = 1$.

C.4 Importance sampling

$P_i^{(\mathrm{IS})}$ are the probabilities P_i in the *importance sampling distribution*. The maximum state distribution for importance sampling is derived simply by scaling the arrival and departure rates as described in section 3.4.2. The new rates are substituted in

$$Q^*_i = \frac{\mu^*/\lambda^* - 1}{(\mu^*/\lambda^*)^i - 1} = \frac{\mu/(\lambda \mathrm{BIAS}^2) - 1}{(\mu/(\lambda \mathrm{BIAS}^2))^i - 1}. \tag{C.9}$$

With optimal scaling $\mathrm{BIAS} = \mu/\lambda$, then $Q^*_i = (\lambda/\mu - 1)/((\lambda/\mu)^i - 1)$.

Thus, substitute Q_i by Q^*_i in (C.6) then:

$$P_i^{(\mathrm{IS})} = \begin{cases} Q^* - Q^*_{i+1} & 0 \le i < N \\ Q^* & i = N \end{cases}. \tag{C.10}$$

C.5 Importance sampling combined with RESTART

$P_i^{(\mathrm{IS+R})}$ are the probabilities P_i in the *importance sampling combined with RESTART distribution.* These are derived by substituting P_i by $P_i^{(\mathrm{IS})}$ from (C.10) in (C.8). Then, the following expression applies:

$$P_i^{(\mathrm{IS+R})} = P_i^{(\mathrm{IS})}/G^* \cdot \prod_{l=0}^{m} R_l \qquad (\Omega_{1\,m-1} \le i < \Omega_{1m}) \wedge (1 < m < M) \tag{C.11}$$

where the normalisation constant G^* satisfies $\sum_{i=0}^{N} P_i^{(\mathrm{IS+R})} = 1$.

C.6 Constant ratio between direct and optimal importance sampling

The ratio between the maximum state probabilities in original and *optimal* importance sampling distributions, $P_i / P_i^{(\mathrm{IS})}$, are observed from the figures in section 2.7.4 to be constant for all $1 \le i < N$. This constant can be determined analytically:

$$
\begin{aligned}
\frac{P_i}{P_i^{(\mathrm{IS})}} &= \frac{Q_i - Q_{i+1}}{Q^*{}_i - Q^*{}_{i+1}} \\
&= \frac{(\mu/\lambda - 1)/((\mu/\lambda)^i - 1) - (\mu/\lambda - 1)/((\mu/\lambda)^{i+1} - 1)}{(\lambda/\mu - 1)/((\lambda/\mu)^i - 1) - (\lambda/\mu - 1)/((\lambda/\mu)^{i+1} - 1)} \\
&= \frac{(\mu/\lambda - 1) \cdot [1/((\mu/\lambda)^i - 1) - 1/((\mu/\lambda)^{i+1} - 1)]}{(\lambda/\mu - 1) \cdot [1/((\lambda/\mu)^i - 1) - 1/((\lambda/\mu)^{i+1} - 1)]} \\
&= -\frac{\mu}{\lambda} \cdot \frac{((\mu/\lambda - 1) \cdot (\mu/\lambda)^i \cdot ((\lambda/\mu)^i - 1) \cdot ((\lambda/\mu)^{i+1} - 1))}{((\lambda/\mu - 1) \cdot (\lambda/\mu)^i \cdot ((\mu/\lambda)^i - 1) \cdot ((\mu/\lambda)^{i+1} - 1))} \\
&= -\frac{\mu}{\lambda} \cdot \frac{(\mu/\lambda - 1) \cdot (1 - (\mu/\lambda)^i) \cdot ((\lambda/\mu)^{i+1} - 1)}{(\lambda/\mu - 1) \cdot ((\mu/\lambda)^i - 1) \cdot (1 - (\lambda/\mu)^{i+1}) \cdot \mu/\lambda} \\
&= -\frac{\mu}{\lambda} \cdot \frac{(\mu/\lambda - 1)}{(\lambda/\mu - 1) \cdot \mu/\lambda} \\
&= \frac{\mu}{\lambda}
\end{aligned}
\tag{C.12}
$$

Hence, the constant ratio is equal to the optimal BIAS -factor, see section 3.3.3.1.

For other biasing than optimal, the ratio $P_i / P_i^{(\mathrm{IS})}$ is not constant.

Appendix D

Analytic variance of the importance sampling estimates and the likelihood ratio

This appendix provides the formulas necessary to produce the plots in chapter 3.5. These plots show the variance of the conditional and unconditional likelihood ratio, and of the importance sampling estimate, $\hat{\gamma}_{\mathrm{IS}}$.

D.1 Model assumptions

The formulas in this appendix are developed under assumption of simple random walks. The K-dimensional simulation process $\underset{\sim}{X}(t)$ is described in section 3.2, and in a more general form in section 4.2. The recursive expression for the embedded process $\underset{\sim}{X}_n$ given in (4.2) is.

$$\underset{\sim}{X}_n = \underset{\sim}{X}_{n-1} + \underset{\sim}{Z}_n \tag{D.1}$$

where $\underset{\sim}{Z}_n$ is the random event variable. With absorption barriers at origin and at resource capacity N, the expression in (D.1) is changed to:

$$\underset{\sim}{X}_n = \begin{cases} \underset{\sim}{X}_{n-1} + \underset{\sim}{Z}_n & (0 < |\underset{\sim}{X}_{n-1}| < N) \\ \underset{\sim}{X}_{n-1} & |\underset{\sim}{X}_{n-1}| \le 0 \wedge |\underset{\sim}{X}_{n-1}| \ge N \end{cases} \tag{D.2}$$

where $|\underset{\sim}{X}_n| = \sum_{k=1}^{K} \omega_{k;n}$, i.e. the sum of the state variables at embedded point n.

Example D.1: For $K = 1$, the event variable becomes (where $p + q = 1$):

$$Z_n = \begin{cases} -1 & \text{with probability } q \\ 1 & \text{with probability } p \end{cases} \tag{D.3}$$

and the recursion from (D.2) is:

$$X_n = \begin{cases} X_{n-1} + Z_n & (0 < X_{n-1} < N) \\ X_{n-1} & X_{n-1} \le 0 \wedge X_{n-1} \ge N \end{cases}. \tag{D.4}$$

Figure D.1 illustrates how a one dimensional random walk on a $N + 1$ states Markov chain looks like.

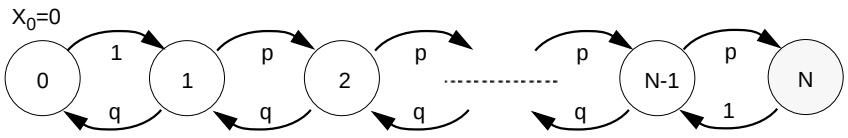


Figure D.1: Simple N+1 state Markov chain model of an M/M/1/N queue.

The likelihood ratio, $L(s)$, and the property of interest γ, are both associated with a *sample path* $\underset{\sim}{s}$ *(or cycle)*, see (3.2). A sample path consists of n observations of the process X_i between two arrivals to empty system, i.e. where $X_0 = 0$ and $X_n = 0$ and $X_i > 0$ for all $i = 1, \ldots, n-1$. A sample path $\underset{\sim}{s}$ is also referred to as a *cycle* because it represents a regenerative cycle in the Markov chain model.

D.2 Contents of appendix

Expression for the following properties are given in this appendix:

$p(i, N)$ the probability of absorption in state N after i events in a sample path starting from state 0, see definition in (D.5) and expressions in (D.8) and (D.12).

$p(i, 0)$ the probability of absorption in state 0 after i events in a sample path starting from state 1, see expressions in (D.9) and (D.13).

$E(L|N)$ the first order expectation of the likelihood ratio, given absorption in state N, see (D.30).

$E(L^2|N)$ the second order expectation of the likelihood ratio, given absorption in state N, see (D.32).

$Var(L|N)$ the variance of the likelihood ratio, given absorption in state N, see (D.35).

$E(L|0)$ the first order expectation of the likelihood ratio, given absorption in state 0, see (D.33).

$E(L^2|0)$ the second order expectation of the likelihood ratio, given absorption in state 0, see (D.34).

$Var(L|0)$ the variance of the likelihood ratio, given absorption in state 0, see (D.36).

$E(L)$ the first order expectation of the likelihood ratio, see (D.37).

$Var(L)$ the variance of the likelihood ratio, see (D.38).

$Var(\hat{\gamma}_{IS})$ the variance of the importance sampling estimate, see (D.41).

Section D.3.1 presents a recursive formula for $p(i, N)$, and D.3.2 includes a closed form expression from [CM65]. Observe that the original expressions from the book [CM65] has been corrected for a missing factor.

In section D.4, the expectation and variance of the conditional likelihood ratio, given a visit to state N, are described in an explicit form by simplifications of the $p(n, N)$. The variance of $\hat{\gamma}_{IS}$ easily follows from $E(L|N)$ and $Var(L|N)$.

As a part of the derivation of the expression $Var(L|N)$, a convergence condition to ensure finite variance was established for single dimensional model in section D.4.3. An explicit expression for the upper bounds of the BIAS factor is derived.

By application of Kroenecker algebra, see [Les88] for a brief overview, the probability matrix can be extended to a multidimensional model. This makes the recursive formulas from section D.3.1 and D.4.2 applicable to multidimensional models. In section D.6 the extension to a two dimensional model is described.

In section D.7, the formulas are used on a few examples of one and two dimensional models.

D.3 The probability of absorption in state N

An expression of $p(i, N)$ is established in this section. A recursive equation which is valid for both single and multidimensional models is described. The calculation of $p(i, N)$ are computer demanding, and hence for single dimensional models an alternative and *explicit* expression for efficient calculations is derived.

D.3.1 The probability matrix approach

With reference to the model in figure D.1, the probability of visiting state N for the first time after i events in a sample path, $\underset{\sim}{s}$, given start in state 0, is:

$$p(i, N) = Pr\{X_i = N \,|\, (0 < X_j < N, (j = 1, \ldots, i-1)) \wedge (X_0 = 0)\}. \tag{D.5}$$

This is the same as the probability of reaching the state N after $i-1$ events in a Markov chain with absorbing barriers at state 0 and N, given start in state 1. Figure D.2 introduces absorbing barriers at state 0 and N in the model from figure D.1

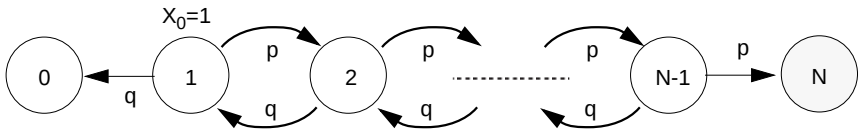


Figure D.2: Absorbing barriers at state 0 and N in the model of figure D.1.

Let $\underset{\sim}{P}$ be the transition probability matrix for the model in figure D.2:

$$\underset{\sim}{P} = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ q & 0 & p & \ldots & 0 \\ & \ldots & \ldots & \ldots & \\ 0 & \ldots & q & 0 & p \\ 0 & \ldots & 0 & 0 & 1 \end{bmatrix}. \tag{D.6}$$

By Kroenecker algebra, described in section D.6, a transition probability matrix $\underset{\sim}{P}$ can be generated for multidimensional models, which makes the results in the following applicable also to these models. The probability distribution on $\underset{\sim}{\Omega}$ after event i is given by:

$$\underset{\sim}{\Pi}^{(i)} = \underset{\sim}{\Pi}^{(i-1)} \times \underset{\sim}{P} = \underset{\sim}{\Pi}^{(0)} \times \underset{\sim}{P}^{i}. \tag{D.7}$$

The initial condition for one dimensional model is $X_0 = 1$, i.e. $\underset{\sim}{\Pi}^{(0)} = \{0, 1, 0, \ldots, 0\}$.

Now:

$$p(i, N) = \Pi_N^{(i)} - \Pi_N^{(i-1)} \text{ for } i = N, \ldots, \tag{D.8}$$

$$p(i, 0) = \Pi_0^{(i)} - \Pi_0^{(i-1)}, \text{ for } i = 1, \ldots, \tag{D.9}$$

where the latter is the corresponding probability of reaching the absorbing state 0 in i events, given start in state 1.

The asymptotic probabilities after a large number of events i are ($\Pi_0^{(0)} = 0$ and $\Pi_N^{(0)} = 0$):

$$p(N) = \sum_{i=1}^{\infty} \Pi_N^{(i)} - \Pi_N^{(i-1)} = \Pi_N^{(\infty)}, \tag{D.10}$$

$$p(0) = \sum_{i=1}^{\infty} \Pi_0^{(i)} - \Pi_0^{(i-1)} = \Pi_0^{(\infty)}. \tag{D.11}$$

These recursive formulas are computer demanding, and hence an explicit expression is required as the size of the model grows, i.e. N become larger.

D.3.2 The Cox and Miller formula

In [CM65], explicit expressions for (D.8)-(D.11) are found, valid for an one dimensional Markov chain. The absorbing barriers are defined at state $-b$ and a. In the model from figure D.2 this corresponds to $-b = -1$ and $a = N - 1$, i.e. a renumbering of the entire state space $\underset{\sim}{\Omega}$ by -1. Section 2.2 in [CM65] contains an error in (22) on page 32. A factor $4\sqrt{pq}$ is missing. Hence, the *corrected* formulas are included in this section for the sake of completeness, using $-b = -1$ and $a = N - 1$ as absorbing barriers, and assuming that $p + q = 1$:

$$p(i, N) = \frac{2\sqrt{pq}}{N} \cdot \left(\frac{p}{q}\right)^{\frac{N-1}{2}} \cdot \sum_{\nu=1}^{N-1} \frac{\alpha_\nu(1)}{s_\nu^{i-1}} \text{ for } i = N-1, \ldots, \tag{D.12}$$

$$p(i, 0) = \frac{2\sqrt{pq}}{N} \cdot \left(\frac{q}{p}\right)^{1/2} \cdot \sum_{\nu=1}^{N-1} \frac{\alpha_\nu(N-1)}{s_\nu^{i-1}}, \text{ for } i = 1, \ldots, \tag{D.13}$$

where $\alpha_\nu(x) = (-1)^{\nu+1} \cdot \sin\left(\frac{x\nu\pi}{N}\right) \sin\left(\frac{\nu\pi}{N}\right)$ and $s_\nu = 1/(2\sqrt{pq} \cdot \cos(\nu\pi/N))$.

Furthermore, if $p \neq q$:

$$p(N) = p^{N-1} \frac{p-q}{p^N - q^N} \text{ (for } p = q\text{: } p(N) = 1/N\text{)}, \tag{D.14}$$

$$p(0) = q \frac{p^{N-1} - q^{N-1}}{p^N - q^N} \text{ (for } p = q\text{: } p(0) = (N-1)/N\text{)}. \tag{D.15}$$

D.3.3 The conditional probabilities

Normalization of $p(i, N)$ and $p(i, 0)$ by $p(N)$ and $p(0)$, respectively, gives the following conditional probabilities:

$$p(i|N) = \frac{p(i, N)}{p(N)}, \tag{D.16}$$

$$p(i|0) = \frac{p(i, 0)}{p(0)}. \tag{D.17}$$

These are the probabilities of absorption after i events, given start in state 1, and absorption in state N and 0, respectively.

D.3.4 Importance sampling probabilities

The notation $p(i, \omega)$ is applied in the following to represent the probability of reaching the absorbing state ω in i events, where $\omega = 0$ or N. Under the importance sampling distribution the arrival and departure probabilities, p and q, are substituted by p^* and q^*, and $p(i, \omega)$ is denoted $p^*(i, \omega)$.

D.4 The variance of the likelihood ratio

Through a series of simulation experiments it has been observed that the variance of the likelihood ratio changes significantly as the scaling of the importance sampling parameters changes, i.e. as the BIAS factor from section 3.4.2 changes. Expression for the variance of the likelihood ratio is therefore developed to investigate how this variance changes and how it relates to the change in the variance of the $\hat{\gamma}_{IS}$, see section D.5.

D.4.1 The likelihood ratio after *i* events and absorption in state ω for a simple chain

Let the model still be a one dimensional Markov chain with constant rates and with two absorbing barriers. The observed likelihood ratio in a sample path is given by the number of events $i-1$ from the initial state 1 and to the absorptions state. The importance sampling strategy turns off the biasing when the target state N is reached, and hence the remaining sequence from state N and back to 0 will not affect the accumulated likelihood ratio. The likelihood ratio consists of two factors, one contribution associated with the direct path from initial state 1 to the given absorbing state, either 0 or N. The second contribution is from an arbitrary number j of arrival and departure events between the transient states of the model, i.e. the states $1, \ldots, N-1$. These arrival and departure events are denoted *loops*. Observe that each loop consists of two events.

Let $L|(j, \omega)$ denote the likelihood ratio given j loops and absorption in state ω:

$$L|(j, N) = \left(\frac{p}{p^*}\right)^{N-1} \cdot \left(\frac{pq}{p^*q^*}\right)^j \text{ for } j = 0, \ldots, \infty, \tag{D.18}$$

$$L|(j, 0) = \left(\frac{q}{q^*}\right) \cdot \left(\frac{pq}{p^*q^*}\right)^j \text{ for } j = 0, \ldots, \infty. \tag{D.19}$$

The factor $(p/p^*)^{N-1}$ in (D.18) is the contribution to the likelihood ratio from a direct path from the initial state 1 up to the absorption state N. The factor $((pq)/(p^*q^*))^j$ is the corresponding contribution where j *loops* are observed before reaching state N. For $L|(j, N)$ the j loops corresponds to $i = N + 2j$ events from initial state 1. This implies that (D.18) is defined only for $i = N-1, N+1, N+3, \ldots$. Correspondingly, from (D.19), the factor (q/q^*) is the contribution of the direct (a single event) path from state 1 to absorption in state 0, and $((pq)/(p^*q^*))^j$ is, as in (D.18), the contribution

from j loops. The number of events are $i = 1 + 2j$, and hence (D.19) is defined for $i = 1, 3, 5, \ldots$.

D.4.2 First and second order expectations; the probability matrix approach

In section D.3.1, the original model was described by the transition probability matrix $\underset{\sim}{P}$ from (D.6). The corresponding matrix under importance sampling parameters is:

$$\underset{\sim}{P}^* = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ q^* & 0 & p^* & \ldots & 0 \\ & \ldots & \ldots & \ldots & \\ 0 & \ldots & q^* & 0 & p^* \\ 0 & \ldots & 0 & 0 & 1 \end{bmatrix}. \tag{D.20}$$

For multidimensional Markov models, a corresponding matrix $\underset{\sim}{P}^*$ can be generated by Kroenecker algebra.

Let $\underset{\sim}{\Lambda}^{(0)} = \{0, 1, 0, \ldots, 0\}$ be the initial likelihood vector. The likelihood vector after i events in the Markov chain is given by the following recursive matrix operations:

$$\underset{\sim}{\Lambda}^{(i)} = \underset{\sim}{\Lambda}^{(i-1)} \times \frac{\underset{\sim}{P}}{\underset{\sim}{P}^*}\underset{\sim}{P}^* = \underset{\sim}{\Lambda}^{(i-1)} \times \underset{\sim}{P} = \underset{\sim}{\Lambda}^{(0)} \times \underset{\sim}{P}^i \text{ (similar to (D.7))}. \tag{D.21}$$

The square likelihood vector is:

$$(\underset{\sim}{\Lambda}^2)^{(i)} = (\underset{\sim}{\Lambda}^2)^{(i-1)} \times \left(\frac{\underset{\sim}{P}}{\underset{\sim}{P}^*}\right)^2 \underset{\sim}{P}^* = (\underset{\sim}{\Lambda}^2)^{(i-1)} \times \frac{\underset{\sim}{P}}{\underset{\sim}{P}^*}\underset{\sim}{P} = (\underset{\sim}{\Lambda}^{(0)})^2 \times \left(\frac{\underset{\sim}{P}}{\underset{\sim}{P}^*}\underset{\sim}{P}\right)^i. \tag{D.22}$$

The *first order expectation* of $L|N$ is obtained from $\underset{\sim}{\Lambda}^{(i)}$ after infinite number of events:

$$E(L|N) = \Lambda_N^{(\infty)} / \Pi_N^{(\infty)} \tag{D.23}$$

and the *second order expectation* of $L|N$ from $(\underset{\sim}{\Lambda}^2)^{(i)}$ is:

$$E(L^2|N) = \Lambda_N^{2\,(\infty)} / \Pi_N^{(\infty)}. \tag{D.24}$$

Correspondingly, the first and second order expectations for the likelihood ratio given return to state 0 before reaching state N, are:

$$E(L|0) = \Lambda_0^{(\infty)}/\Pi_0^{(\infty)}, \tag{D.25}$$

$$E(L^2|0) = \Lambda_0^{2(\infty)}/\Pi_0^{(\infty)}. \tag{D.26}$$

For practical numerical calculations of (D.23) to (D.25), only a finite number of events i is included. This means that the results are approximations. For BIAS factors close to 1, they converge after a few events, and the approximation is good. But, for higher BIAS the convergence is very slow, if converging at all, and a very large number of events have to be included to get a good approximation. The calculation become very computer demanding.

Instead, an explicit expression should be applied whenever available. For one dimensional Markov model, such an expression can be obtained using the equations in section D.3.2. This makes it possible to determine the upper bounds of the BIAS factor where the variance is finite, i.e. $E(L^2|N)$ converges.

D.4.3 First and second order expectations; one dimensional model

The expectation of the likelihood ratio is given by combining the expression obtained in the previous sections.

D.4.3.1 First order expectation; absorption in state N

The *first order expectation* of $L|N$ is:

$$E(L|N) = \sum_{j=0}^{\infty} L(j|N)(p^*(N-1+2j|N)) = \sum_{j=0}^{\infty} L(j|N)\frac{p^*(N-1+2j,N)}{p^*(N)}. \tag{D.27}$$

Recall form section D.3.2 that:

$$\alpha_\nu(x) = (-1)^{\nu+1}\cdot \sin\left(\frac{x\nu\pi}{N}\right)\sin\left(\frac{\nu\pi}{N}\right) \tag{D.28}$$

and observe that:

$$s^*_\nu = 1/(2\sqrt{p^*q^*}\cdot\cos(\nu\pi/N)). \tag{D.29}$$

Then, substituting (D.12) and (D.18) in (D.27), and rearrange the order of summation:

$$\begin{aligned}
E(L|N) &= \frac{1}{p^*(N)}\sum_{j=0}^{\infty}\frac{2\sqrt{p^*q^*}}{N}\left(\frac{p}{p^*}\right)^{N-1}\left(\frac{pq}{p^*q^*}\right)^j\left(\frac{p^*}{q^*}\right)^{\frac{N-1}{2}}\sum_{\nu=1}^{N-1}\frac{\alpha_\nu(1)}{(s^*_\nu)^{N-1+2j-1}}\\
&= \frac{2\sqrt{p^*q^*}}{Np^*(N)}\left(\frac{p}{p^*}\right)^{N-1}\left(\frac{p^*}{q^*}\right)^{\frac{N-1}{2}}\sum_{j=0}^{\infty}\left(\frac{pq}{p^*q^*}\right)^j\sum_{\nu=1}^{N-1}\frac{\alpha_\nu(1)}{(s^*_\nu)^{N-1+2j-1}}\\
&= \frac{(2p)^{N-1}}{Np^*(N)}\sum_{j=0}^{\infty}\left(\frac{pq}{p^*q^*}\right)^j\sum_{\nu=1}^{N-1}\alpha_\nu(1)(4p^*q^*)^i\cos^{N-2+2i}(\nu\pi/N)\\
&= \frac{(2p)^{N-1}}{Np^*(N)}\sum_{\nu=1}^{N-1}\alpha_\nu(1)\cos^{N-2}(\nu\pi/N)\sum_{j=0}^{\infty}\phi_1(\nu)^i\\
&= \frac{(2p)^{N-1}}{Np^*(N)}\sum_{\nu=1}^{N-1}\alpha_\nu(1)\cos^{N-2}(\nu\pi/N)/(1-\phi_1(\nu))
\end{aligned} \tag{D.30}$$

where the factor $\phi_1(\nu) = 4pq\cos^2(\nu\pi/N)$ is independent of the change of measure.

D.4.3.2 Convergence condition 1; absorption in state N

To ensure a finite expectation, $E(L|N) < \infty$, the factor $\phi_1(\nu) < 1$ for all ν. Observe that the maximum is $\max(\phi_1(\nu)) = \max(4pq)\cdot\max(\cos^2(\nu\pi/N))$, where

- $\max(\cos^2(\nu\pi/N)) = \cos^2(\pi/N) = \cos^2((N-1)\pi/N) < 1$, and
- $\max(4pq) = 4\max(pq) = 4\cdot 1/4 = 1$ because $0 \le p = 1-q \le 1$.

Then, $\max(\phi_1(\nu)) = \phi_1(1) = 1\cdot\cos^2(\pi/N) < 1$. This means that the expectation is finite $E(L|N) < \infty$ for all ν.

D.4.3.3 Second order expectation; absorption in state N

Correspondingly, the *second order expectation* of $L|N$ is:

$$E(L^2|N) = \sum_{j=0}^{\infty} L(j|N)^2 p^*(N-1+2j|N) = \sum_{j=0}^{\infty} L(j|N)^2 \frac{p^*(N-1+2j, N)}{p^*(N)} \quad \text{(D.31)}$$

Substituting (D.12) and (D.19) in (D.31), and rearranging the order of summation, after some manipulations the following expression is obtained:

$$E(L^2|N) = \frac{(2p)^{N-1}}{N p^*(N)} \left(\frac{p}{p^*}\right)^{N-1} \sum_{\nu=1}^{N-1} \alpha_\nu(1) \cos^{N-2}(\nu\pi/N)/(1-\phi_2(\nu)) . \quad \text{(D.32)}$$

The factor $\phi_2(\nu) = \phi_1(\nu) \cdot (pq)/p^*q^* = 4((pq)^2/p^*q^*)\cos^2(\nu\pi/N)$ is dependent on the change of measure, i.e. the convergence of the second order expectation is dependent on the BIAS factor chosen.

D.4.3.4 Convergence condition 2; absorption in state N

To ensure the second order expectation to be finite, $E(L^2|N) < \infty$, the factor $\phi_2(\nu) < 1$, or $\max(\phi_2(\nu)) = \phi_2(1) < 1$. Then, the convergence requirement, $E(L^2|N) < \infty$, is $4(pq)^2 \le p^*q^*$, i.e. for some change of measure the second order expectation (and the corresponding variance) is infinite! This implies that the variance of the likelihood ratio, and the importance sampling estimate, is infinite for some change of measure.

Figure D.3 shows a plot of the $\phi_2(1)$ for different BIAS factors in 3 different one dimensional models. The shaded areas are the stable regions of the BIAS factor where the second order expectation is finite. It can been observed from the figure that the stable region become narrower as the model size, N, increases. This indicates that, as the model increases, it is increasingly important to have good methods to obtain an change of importance sampling parameters that is close to the optimal. Furthermore, it can be observed from figure D.3 that the optimal BIAS factor, (BIAS=μ/λ, obtained by use of the large deviation results, see section 3.4.2.1), is closer to the upper than the lower bound of this stable region.

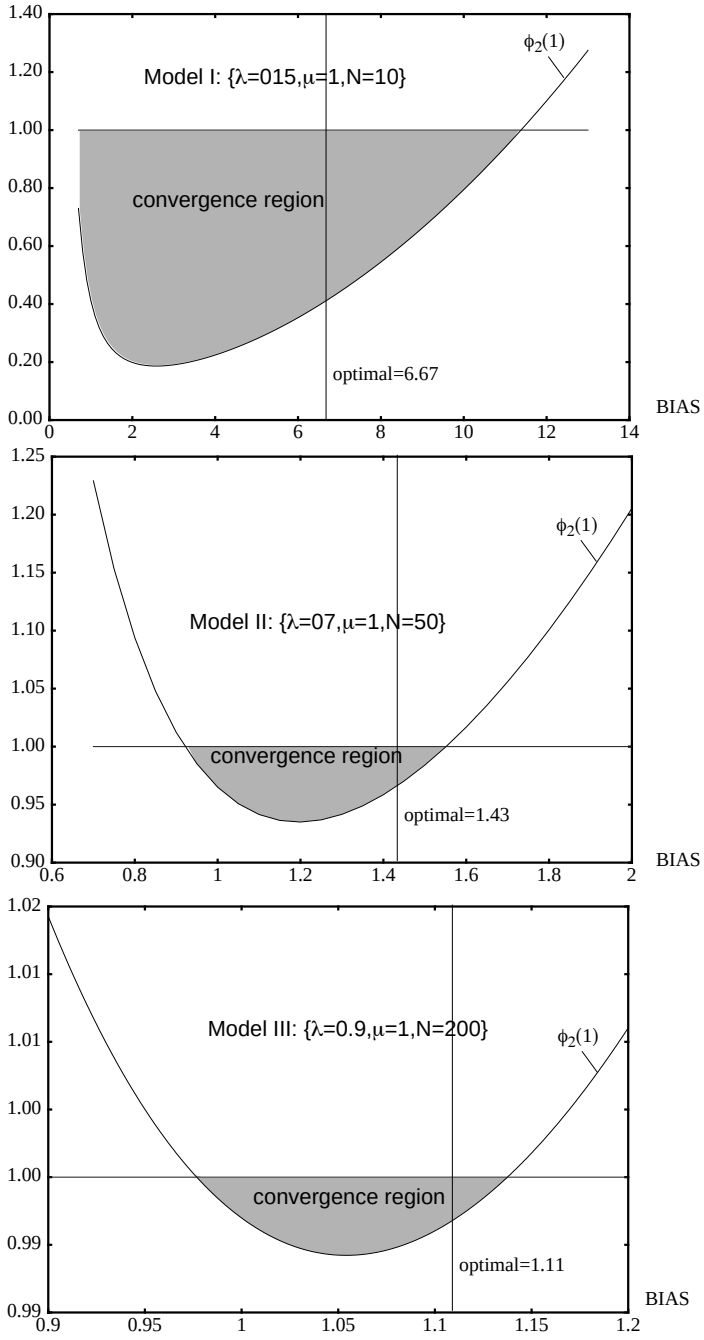


Figure D.3: The factor $\phi_2(1)$ must be less than 1 to obtain $E(L^2|N) < \infty$, i.e the shaded area indicates the stable region of the BIAS factor.

D.4.3.5 Absorption in state 0

The *first* and *second order expectations* of $L|0$ are derived similar to the expressions of $L|N$:

$$E(L|0) = \frac{2q}{N\,p^*(N)} \sum_{\nu=1}^{N-1} \alpha_\nu (N-1)/(1-\phi_1(\nu)), \tag{D.33}$$

$$E(L^2|0) = \frac{2q}{N\,p^*(N)} \cdot \frac{q}{q^*} \cdot \sum_{\nu=1}^{N-1} \alpha_\nu (N-1)/(1-\phi_2(\nu)). \tag{D.34}$$

The convergence conditions for these two sums are the same as conditions 1 and 2 from sections D.4.3.2 and D.4.3.4, respectively.

D.4.4 The variance of the conditional likelihood ratio

The variance of the conditional likelihood ratio, given absorption in state 0 or N are:

$$Var(L|N) = E(L^2|N) - E(L|N)^2, \tag{D.35}$$

$$Var(L|0) = E(L^2|0) - E(L|0)^2. \tag{D.36}$$

D.4.5 The variance of the unconditional likelihood ratio

The expectation and variance of the unconditional likelihood ratio are:

$$E(L) = E(L|N)p^*(N) + E(L|0)p^*(0) = p(N) + p(0) = 1, \tag{D.37}$$

$$\begin{aligned} Var(L) &= (E(L^2|N)p^*(N) + E(L^2|0)p^*(0)) - E(L)^2 \\ &= E(L^2|N)p^*(N) + E(L^2|0)p^*(0) - 1 \end{aligned}. \tag{D.38}$$

D.5 The variance of the importance sampling estimate

Assume that the property of interest is:

$$g(\underline{s}) = \begin{cases} 1 & \text{state } N \text{ is visited in a sample path } \underline{s} \\ 0 & \text{otherwise} \end{cases}$$

The observations $L_r Z_r$ in the estimator of γ from (3.2) are considered to be a product of two *dependent* stochastic variables, L_r, and Z_r. The properties of L_r are known from the inference in section D.4, while Z_r are variates taken from a binomial distribution with $p^* = \Pr^*(g(\underline{s}) > 0) = p^*(Z = 1) = p^*(N)$.

The nth order expectations of the observation $L_r Z_r$ are obtained by conditioning on the Z_r variates:

$$\begin{aligned} E((L_r Z_r)^n) &= E((L_r Z_r)^n \,|\, Z = 1) p^*(Z = 1) + E((L_r Z_r)^n \,|\, (Z = 0)) p^*(Z = 0) \\ &= E(L_r^n \,|\, N) p^*(N) \end{aligned} \tag{D.39}$$

The expectation of $\hat{\gamma}_{\mathrm{IS}} = (1/R)\sum_{\forall r} L_r Z_r$ is then:

$$E(\hat{\gamma}_{\mathrm{IS}}) = E(\frac{1}{R}\sum_{\forall r} L_r Z_r) = E(L_r Z_r) = E(L_r \,|\, N) p^*(N) = p(N) = \gamma \tag{D.40}$$

and the variance:

$$Var(\hat{\gamma}_{\mathrm{IS}}) = E(L_r^2 \,|\, N) p^*(N) - (E(L_r \,|\, N) p^*(N))^2 = E(L_r^2 \,|\, N) p^*(N) - \gamma^2 . \tag{D.41}$$

The same expressions were derived and presented in [CTS93].

D.6 The transition probability matrix for 2 dimensional model

As mentioned in the previous sections, the transition probability matrix of the one dimensional model can be extended to a matrix for multidimensional model by Kroenecker algebra[1]. In this section the details of an extension to a 2 dimensional matrix are described.

1. For a short introduction see [Les88].

First, recall the following elements of Kroenecker algebra. Let $\underset{\sim}{A}$ and $\underset{\sim}{B}$ be two matrices of order $n \times n$. The Kroenecker *product* is a matrix of order $n^2 \times n^2$ of the form:

$$\underset{\sim}{A} \otimes \underset{\sim}{B} = \begin{bmatrix} a_{11}\underset{\sim}{B} & a_{12}\underset{\sim}{B} & \dots & a_{1n}\underset{\sim}{B} \\ a_{21}\underset{\sim}{B} & a_{22}\underset{\sim}{B} & \dots & a_{2n}\underset{\sim}{B} \\ & \dots & \dots & \\ a_{n1}\underset{\sim}{B} & a_{n2}\underset{\sim}{B} & \dots & a_{nn}\underset{\sim}{B} \end{bmatrix}. \tag{D.42}$$

The Kroenecker *sum* is, where $\underset{\sim}{I}_n$ is an $n \times n$ identity matrix:

$$\underset{\sim}{A} \oplus \underset{\sim}{B} = \underset{\sim}{A} \otimes \underset{\sim}{I}_n + \underset{\sim}{I}_n \otimes \underset{\sim}{B}. \tag{D.43}$$

Now, define the transition rate matrix of order $n \times n$ for the one dimensional model without absorption in figure D.1:

$$\underset{\sim}{Q}_{n \times n} = \begin{bmatrix} 0 & \lambda & 0 & \dots & 0 \\ \mu & 0 & \lambda & \dots & 0 \\ & \dots & \dots & \dots & \\ 0 & \dots & \mu & 0 & \lambda \\ 0 & \dots & 0 & \mu & 0 \end{bmatrix}. \tag{D.44}$$

By Kroenecker sum, two one dimensional models can be combined into a two dimensional model, generating a transition rate matrix of order:

$$\underset{\sim}{Q}_{n^2 \times n^2} = \underset{\sim}{Q}_{n \times n} \oplus \underset{\sim}{Q}_{n \times n} = \underset{\sim}{Q}_{n \times n} \otimes \underset{\sim}{I}_n + \underset{\sim}{I}_n \otimes \underset{\sim}{Q}_{n \times n}. \tag{D.45}$$

The $\underset{\sim}{Q}_{n^2 \times n^2}$ contains the transition rates between all states in the state space given in figure D.4 below. The matrix must be modified to introduce absorption states at the origin where the sample path starts and end, and at the target subspace. The target subspace is defined by all states where the sum exceeds the resource capacity N, $\omega_1 + \omega_2 \geq N$. The resulting state space after modification is given in figure D.5. By repeating the operations in (D.45), and the modifications afterwards, a probability matrix can be generated for models with dimensionality higher than 2.

The transition probability matrix $\underset{\sim}{P}$ is assigned to the $mod(\underset{\sim}{Q}_{n \times n})$ after normalising each row of this matrix. This $\underset{\sim}{P}$ is substituted in (D.9) to derive the probability distributions.

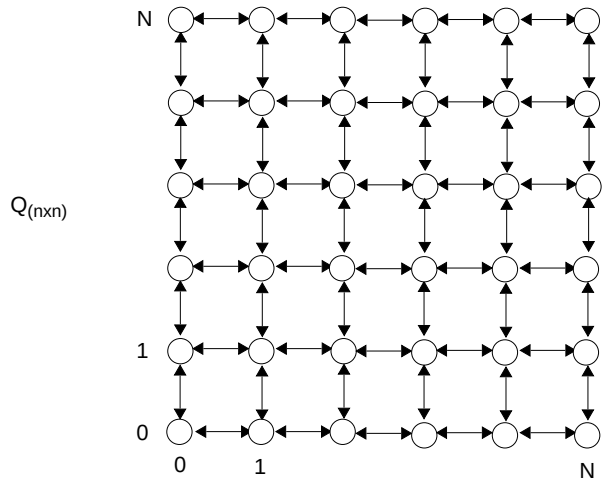


Figure D.4: The state space after full expansion of a single dimensional random walk into a 2 dimensional random walk by Kroenecker sum.

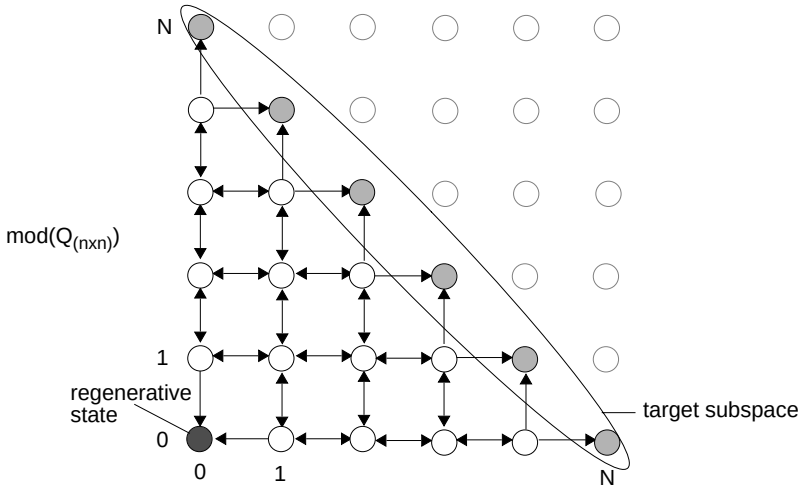


Figure D.5: The modifications of the state space of figure D.4 considering absorption and common resource restriction at N.

D.7 M/M/1/N model examples

D.7.1 Model description

This section contains numerical values from 3 one dimensional and 1 two dimensional models. Several plots are generated to illustrate how the variance of the likelihood ratio and the importance sampling estimate changes as the BIAS factor changes.

Three one dimensional models are defined:

Model I: Single M/M/1/N with λ=0.15, μ=1, N=10,

Model II: Single M/M/1/N with λ=0.7, μ=1, N=50,

Model III: Single M/M/1/N with λ=0.9, μ=1, N=200.

The BIASing in these analytical one dimensional model are the same as proposed in section 3.4.2.1. In addition, a two dimensional model is defined:

Model VI: Shared buffer N=10 (see figure D.5), two user types with λ=0.15, μ=1.

The analytic model for this two dimensional model assumes a constant BIAS factor for all states, ignoring boundary conditions. In the simulations, the BIASing that was proposed in section 3.4.2.2 is used. The BIAS factor alternates between two values, one at the *border*, $\text{BIAS} = \mu/(2\lambda) = 3.333$, and another at the *interior* of the state space (where both user types have at least one entity), $\text{BIAS} = \mu/\lambda = 6.667$. These BIAS factors are a result of reversion of the drift in their respective states.

D.7.2 Observations

Several of observations are made from the study of the variance plots in figure D.6 to D.11. The observations are valid for the model described in section D.1. However, from network simulation results, it is expected that some of the following observations are typical, and valid also for more complex systems.

- The variance of the *unconditional* likelihood ratio, $Var(L)$, in figure D.6 shows a rapid increase as the BIAS factor approaches the upper bound of the *stable region*, see section D.4.3. Observe that the optimal BIAS is closer to the upper than the lower bound. Furthermore, the optimal BIAS is *not equal to* the BIAS factor that gives the minimum variance of the $Var(L)$.

- The variance of the *conditional* likelihood ratio, $Var(L|N)$, given that a visit to state N is observed, is plotted for *Model I-III* in figure D.7. The details in the region close to the minimum variance are plotted in figure D.8 (logarithmic scale) and figure D.9 (linear scale). The $Var(L|N)$ increases rapidly on both sides of the minimum variance. The variance is 0 at the optimal BIAS factor because the likelihood ratio is constant, independent of the sample path, given absorption in state N.

- The $Var(L|N)$ for the 2 dimensional model given in figure D.10, has no significant minimum similar to what was observed for the one dimensional model. The reason is that no BIAS factor results in a likelihood ratio that is constant and independent of the sample path.

- Figures D.7 to D.9 compare the $Var(L|N)$ and $Var(\hat{\gamma}_{IS})$. In a region close to the optimal BIAS, the variance $Var(L|N) < Var(\hat{\gamma}_{IS})$. This region becomes narrower as the model size increases. Outside the region, $Var(L|N) \geq Var(\hat{\gamma}_{IS})$ for all models.

- Let $BIAS_{minL}$ be the BIAS factor for which the $Var(L|N)$ is at its minimum value, and $BIAS_{min\gamma}$ the corresponding for minimum $Var(\hat{\gamma}_{IS})$. In figures D.7 to D.9 it is observed that $BIAS_{minL} < BIAS_{min\gamma}$ for all models. However, the difference between these two BIAS factors becomes smaller when the model size increases.

- In figure D.11, the recursive formula is plotted for a *finite* number of events. This approximation of $E(L|N)$ and $Var(L|N)$ is fairly good in the stable region. But, in the region where the $Var(L|N)$ is unbounded, the approximation fails. This is similar to what will be observed in a simulation experiment where only a finite number of samples are taken. This means that the sample variance is a poor approximation of the $Var(L|N)$ when too strong biasing is applied. See comments in section 3.5.1 where sampling from a heavy tailed distribution is discussed.

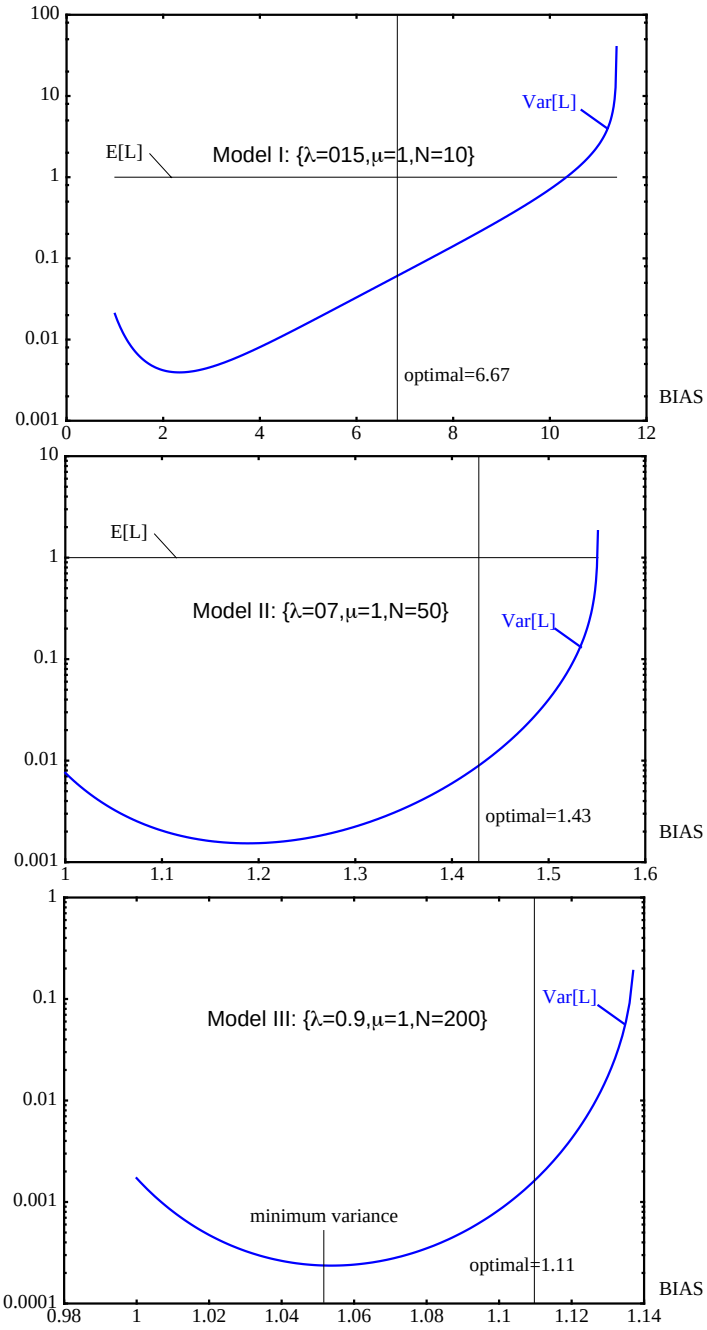


Figure D.6: The mean value and variance of the unconditional likelihood ratio.

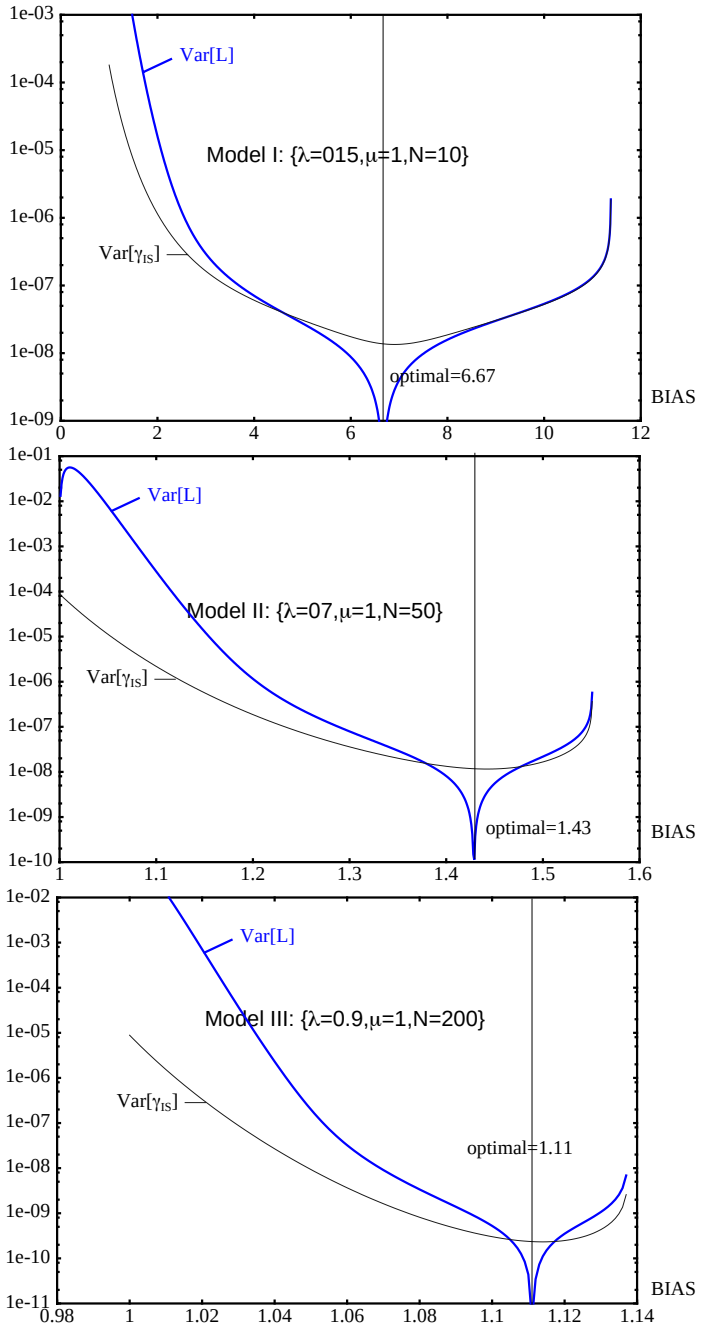


Figure D.7: The comparison of the variance for the IS estimate and the likelihood ratio, given in a logarithmic scale.

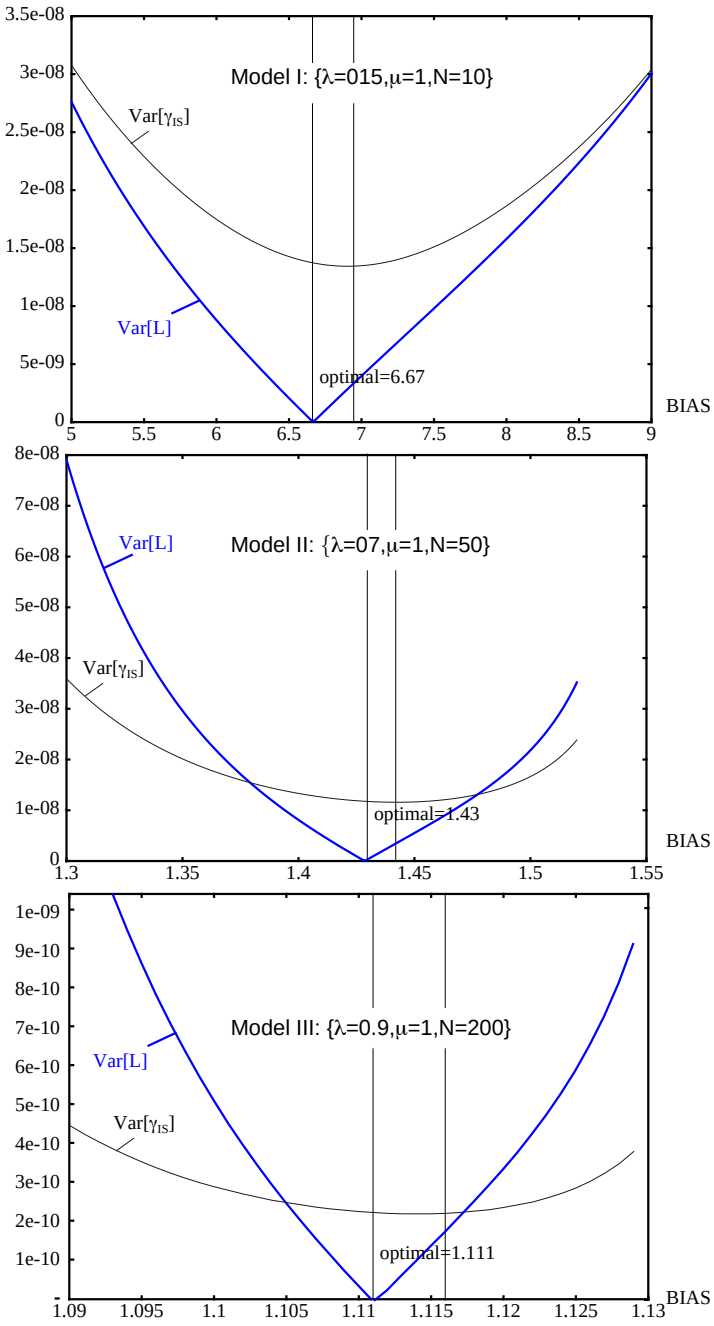


Figure D.8: Details close to the infimum of the comparison of the variance for the IS estimate and the likelihood ratio, given in a linear scale.

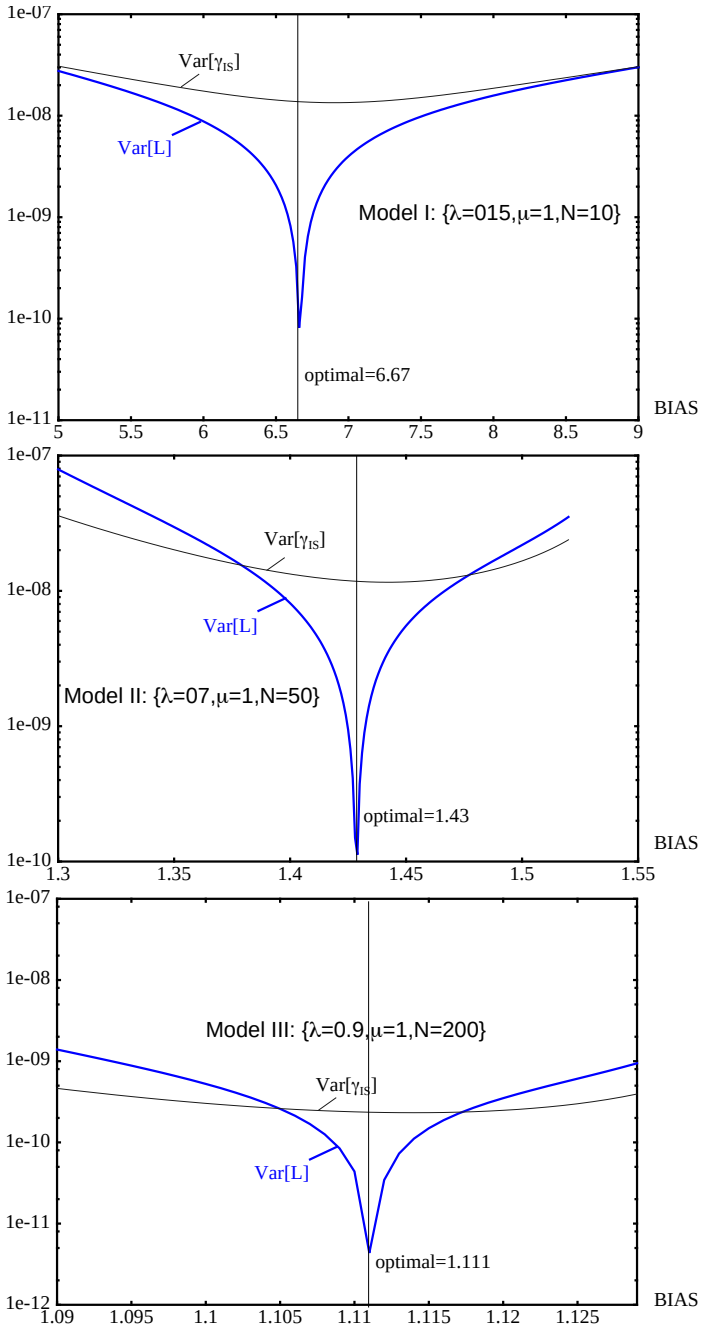


Figure D.9: Details close to the infimum of the comparison of the variance for the IS estimate and the likelihood ratio, given in a logarithmic scale.

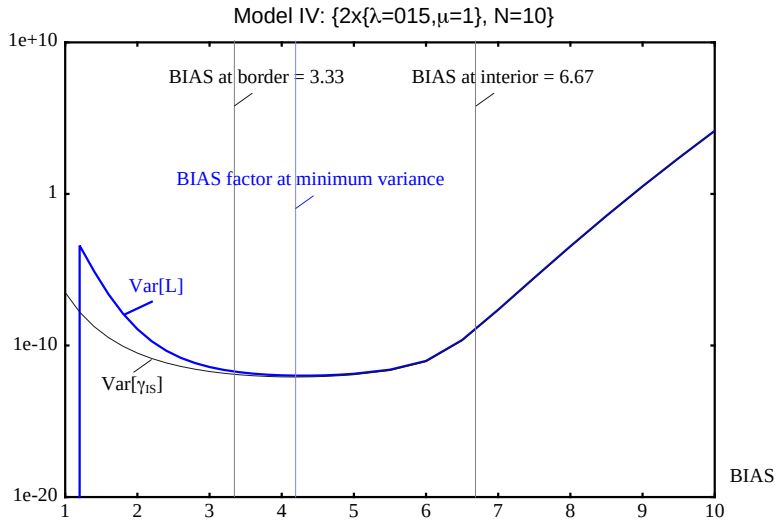


Figure D.10: The comparison of the variance for the IS estimate and the likelihood ratio for the 2 dimensional shared buffer model. The BIAS factors used in simulations of the model, at the interior and at the border of the state space, are added.

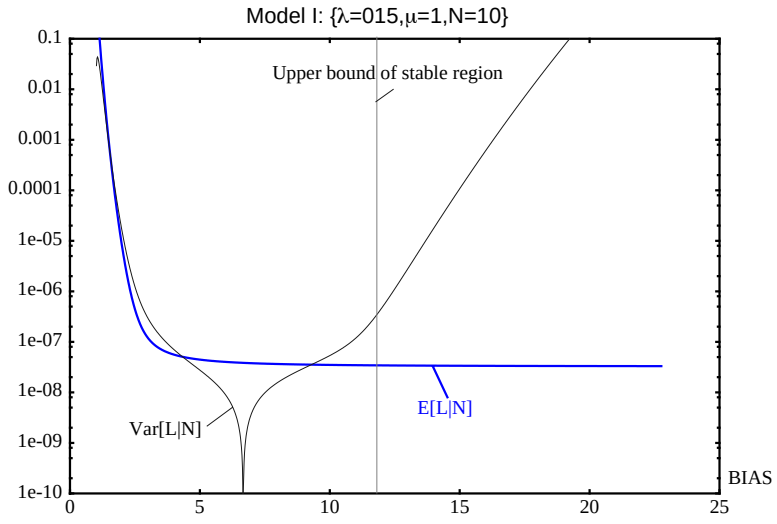


Figure D.11: The mean value and variance of the conditional likelihood ratio obtained by the recursive formula with a finite number of events (i.e. an approximation, observe what happens outside the stable region).

Appendix E

Search for the most likely subpath

This appendix describes the details of an efficient algorithm used for obtaining the most likely subpath from current state up to the first entrance to a given target subspace. The basic idea of estimating the target likelihood by this subpath is given in chapter 5.3.

The search algorithm returns the likelihood π_x, ($x = x_0, \ldots, N_j$) of a subpath as a function of

- current state, $\underset{\sim}{\omega}$,
- generators k in Γ_j,
- target subspace, $\underset{\sim}{\Omega}_j$ and B_j.

The notation used is listed in section B.1.7. The details of the search algorithm are presented in section E.1, while numerical examples are given in section E.2.

E.1 The search algorithm

To explain how the search algorithm works, it is important to realise that σ is a subpath in the multidimensional Markov chain, $\underset{\sim}{\Omega}$, spanned by the generators in Γ_j. This subpath may be mapped to an one dimensional Markov chain, see figure E.1 for an example. It is convenient to define the subpath in terms of the number of resource allocations, x, in addition to the system states. This means that the target subspace definition $\underset{\sim}{\Omega}_j$, has an analogous definition in terms of x, see section 4.3.4:

$$B_j = \left\{ x \,|\, N_j - \left(\min_{k \in \Gamma_j} c_{kj} < x \leq N_j \right) \right\}. \tag{E.1}$$

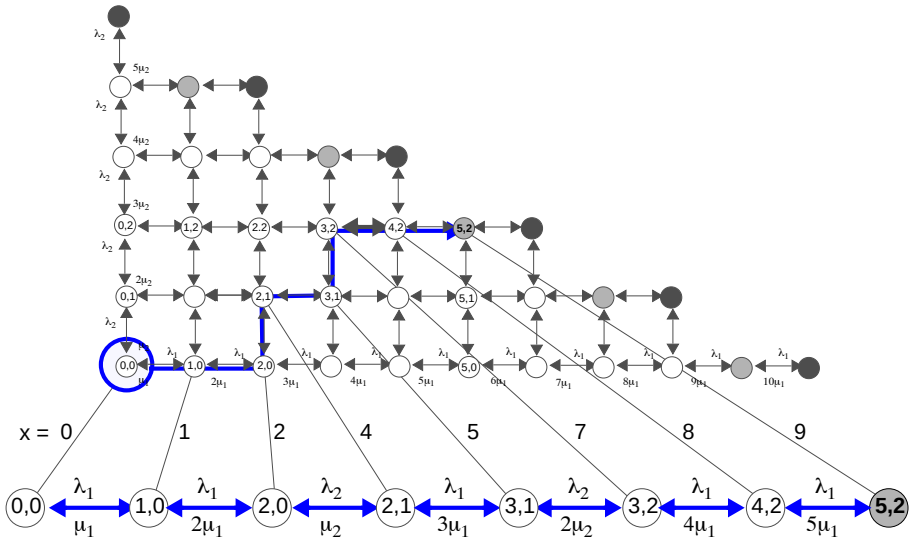


Figure E.1: The sequence $\underset{\sim}{k}$ can be considered as a one dimensional Markov chain in the 2 dimensional Markov chain spanned by the generators in Γ_j.

The search algorithm inspects the subpaths that includes *only arrival events* from current state $\underset{\sim}{\omega}$, because departure events will reduce the accumulated likelihood of a subpath with a given end state. The objective is to identify the most likely subpath from $\underset{\sim}{\omega}$ up to a system state where the number of resource allocations are x, $x = x_0, \ldots, N_j$.

The Markov properties of the underlying simulation process make it possible to derive this subpath from the previously derived subpaths. The states where each generator $k \in \Gamma_j$ have one entity less, and therefore holds c_{kj} resources less, must be considered. For each generator, the contribution to the probability π_x of the state having x resources, is derived from $\pi_{x-c_{kj}}$, i.e. the state with $x - c_{kj}$ resources. The probabilities are determined from the *balance equations* between the states with x and $x - c_{kj}$ resource. In figure E.2 an example is given that illustrates the balance between a state with x resources and the states with $x - c_{k_1 j}$ and $x - c_{k_2 j}$ resources.

Let $\underset{\sim}{\omega}^{(x)}$ be defined as the system state $\underset{\sim}{\omega}$ where x resources are allocated, then the balance equation is defined as follows:

$$\pi_{x-c_{kj}} \cdot \lambda_k(\underset{\sim}{\omega}^{(x-c_{kj})}) = \alpha_{xk} \cdot \mu_k(\underset{\sim}{\omega}^{(x-c_{kj})} + \underset{\sim}{1}_k) \tag{E.2}$$

where α_{xk} is a *probability measure* for each generator k in the state with x resource. This is derived by solving the equations in (E.2). The measure α_{xk} is interpreted as the contribution to the probability of having x resources when the latest arrival of an entity was from generator k. The probability of x resources is denoted π_x and it is the maximum of the α_{xk}s over the k in Γ_j. With this approach it is not necessary to follow every subpath from $\underset{\sim}{\omega}$ up to $\underset{\sim}{\Omega}_j$, only to determine the most likely path up to a state with x resources. This subpath is then repeatedly extended, starting from $x = x_0$ and ending at N_j in unit steps[1].

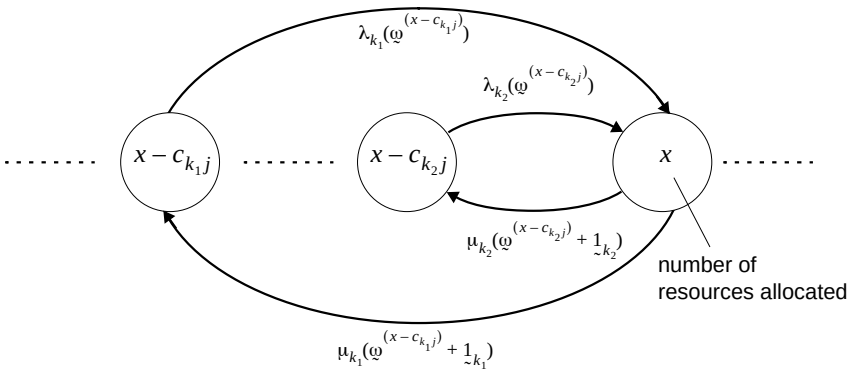


Figure E.2: The steady state balance between a state with x resources allocated, and the states with one entity less of generator k_1 and k_2, respectively.

The following algorithm is proposed to obtain the π_x ($x = x_0, x_0 + 1, \ldots, N_j$):

1. Initialise; $x = x_0$.

   The search starts from current state $\underset{\sim}{\omega}$ with $x_0 = \sum_{k \in \Gamma_j} \omega_k c_{kj}$ resources allocated. The unnormalised probability measures and the corresponding normalisation constant, are $\alpha_{xk} = 1$ and $G_{xk} = 1$; $\forall (k \in \Gamma_j)$, respectively, and $\pi_{x_0} = 1$.

2. Continue; $x = x + 1$.

   For every step, the probability measure for each generator is obtained by solution of the balance equations in (E.2).

1. If the *greatest common devisor* (gcd) of the c_{kj} for the generators in Γ_j is greater than 1, all resource counters (C_{j0}, N_j, and the c_{kj}) are scale by the gcd(c_{kj}) before invoking the search algorithm.

$\alpha_{xk} = \pi_{x-c_{kj}} \cdot \lambda_k(\underset{\sim}{\omega}^{(x-c_{kj})}) / \mu(\underset{\sim}{\omega}^{(x-c_{kj})} + \underset{\sim}{1}_k)$ and

$G_{xk} = G_{x-c_{kj}} + \alpha_{xk}$; $\forall(k \in \Gamma_j)$.

The generator that has the largest contribution trigger the event leading to $\underset{\sim}{\omega}^{(x)}$. The generator k is denoted k_x and is identified by comparison of the normalised contributions, α_{xk}/G_{xk}, where k_x is the k with the maximum contribution:

$k_x = \{k | (\alpha_{xk}/G_{xk}) = \max(\alpha_{xk}/G_{xk}) \wedge k \in \Gamma_j\}$.

The unnormalised probability of x is assigned to the probability measure of the k_x, $\pi_x = \alpha_{xk_x}$ with the corresponding constant $G_x = G_{xk_x}$. The system state associated with x is obtained by adding 1 entity from generator k_x to the system state associated with $x - c_{k_x j}$:

$\underset{\sim}{\omega}^{(x)} = \underset{\sim}{\omega}^{(x-c_{k_x j})} + \underset{\sim}{1}_{k_x}$.

3. Terminate; $x = N_j$.

When the search is completed, a π_x exists for every $x = x_0, \ldots, N_j$. The most *likely* and the most *important* subpath is assigned as described in section 5.3.

E.2 Numerical examples

To demonstrate the use of this algorithm, a model with $K = 2$ generators and $J = 1$ resource pool (target) is included. The state space description is given in figure E.3, while the model parameters are given in table E.1. The figure shows the resulting subpaths from 3 different starting states, $\underset{\sim}{\omega}$.

Table E.1: Parameters for the generators in the model with $N_1 = 10$.

Generator, k	λ_k	M_k	μ_k	S_k	c_{k1}
1	1	∞	1	10	1
2	0.1	∞	1	5	2

The results of the search algorithm are given in tables E.2 to E.4. To identify the most likely, or important, subpaths, the following target subspace is applied:

$$B_1 = \{x | (8 < x \leq 10)\} = \{9, 10\} \tag{E.3}$$

Example E.1: The search starts from the current state $\underset{\sim}{\omega} = \{0, 0\}$, with the $x_0 = 0$. The probabilities π_x, ($x = 0, \ldots, 10$), are given in table E.2. The most likely subpath is the maximum of π_9 and π_{10}. The corresponding subpath, σ, is given by the system states $\underset{\sim}{\omega}^{(x)}$ (the entries in table E.2 that are not marked with an asterisk). Figure E.4 includes the subpath that is entering the target subspace at $x = 9$ ($\underset{\sim}{\omega}^{(9)} = \{5, 2\}$).

Table E.2: Search for a subpath from {0,0} for the example in figure E.3. The most likely target is $\max(\pi_9, \pi_{10}) = \pi_9$. *(the entries marked with an asterisk are not a part of the most likely sub-path).*

x	$\underset{\sim}{\omega}^{(x)}$	k_x	π_x	G_x	π_x / G_x
0	{0, 0}	-	1	1	1
1	{1,0}	1	1	2	0.5
2	{2,0}	1	0.5	2.5	0.2
3*	{3,0}	1	0.167	2.67	0.0625
4	{2,1}	2	0.05	2.55	1.96e-2
5	{3,1}	1	1.67e-2	2.57	6.49e-3
6*	{4,1}	1	4.17e-3	2.57	1.62e-3
7	{3,2}	2	8.33e-4	2.57	3.25e-4
8	{4,2}	1	2.08e-4	2.57	8.11e-5
9	{5,2}	1	4.17e-5	2.57	1.62e-5
10*	{4, 3}	2	6.94e-6	2.57	2.70e-6

Example E.2: The search starts from the current state $\underset{\sim}{\omega} = \{1, 2\}$, with $x_0 = 5$. The probabilities π_x, ($x = 5, \ldots, 10$), are given in table E.3. Figure E.4 includes the subpath that is entering the target subspace at $x = 9$, $\underset{\sim}{\omega}^{(9)} = \{5, 2\}$.

Example E.3: The search starts from the current state $\underset{\sim}{\omega} = \{5, 0\}$, with $x_0 = 5$. The probabilities π_x, ($x = 5, \ldots, 10$), are given in table E.4. Figure E.4 includes the subpath that is entering the target subspace at $x = 9$, $\underset{\sim}{\omega}^{(9)} = \{5, 2\}$.

Example E.4: In this example the two generators have different priority levels. Generator 2 can preempt entities from generator 1. Figure E.4 shows the new state

Table E.3: Search for a subpath from {1,2} for the example in figure E.3. The most likely target is $\max(\pi_9, \pi_{10}) = \pi_9$. *(the entries marked with an asterisk are not a part of the most likely sub-path).*

x	$\underset{\sim}{\omega}^{(x)}$	k_x	π_x	G_x	π_x / G_x
5	{1,2}	-	1	1	1
6	{2,2}	1	0.5	1.5	3.33e-1
7	{3,2}	1	0.167	1.67	1.00e-1
8	{4,2}	1	4.17e-2	1.71	2.44e-2
9	{5, 2}	1	8.33e-3	1.71	4.85e-3
10*	{4,3}	2	1.39e-3	1.72	8.12e-4

Table E.4: Search for a subpath from {5,0} for the example in figure E.3. The most likely target is $\max(\pi_9, \pi_{10}) = \pi_9$. *(the entries marked with an asterisk are not a part of the most likely sub-path).*

x	$\underset{\sim}{\omega}^{(x)}$	k_x	π_x	G_x	π_x / G_x
5	{5,0}	-	1	1	1
6*	{6,0}	1	0.167	1.167	0.143
7	{5,1}	2	0.1	1.1	9.09e-2
8*	{6,1}	1	1.67e-2	1.12	1.50e-2
9	{5,2}	2	5.00e-3	1.11	4.52e-3
10*	{6,2}	1	8.33e-4	1.11	7.53e-4

transitions that are added to the model in figure E.3. The search algorithm is invoked twice, once for each priority level:

i. *High priority* ($p = 0$): Find the most likely subpath with $\Gamma_1^{(0)} = \{2\}$ and $B_1^{(0)} = \{10\}$ ($\underset{\sim}{\Omega}_1^{(0)} = \{0, 5\}$). Table E.5 includes the $\pi_x^{(0)}$ when current state is $\underset{\sim}{\omega} = \{3, 1\}$. This corresponds to starting from $\underset{\sim}{\omega} = \{0, 1\}$ since only generator 2 is included in $\Gamma_1^{(0)}$ and the generator 1 entities will be preempted.

ii. *Low priority* ($p = 1$): Find the most likely subpath with $\Gamma_1^{(1)} = \{1, 2\}$ and $B_1^{(1)} = \{10\}$ ($\underset{\sim}{\Omega}_1^{(1)} = \{\{0, 5\}, \{2, 4\}, \{4, 3\}, \{6, 2\}, \{8, 1\}, \{10, 0\}\}$). Table E.6 includes the $\pi_x^{(1)}$ when current state is $\underset{\sim}{\omega} = \{3, 1\}$.

The most likely subpath from state {3,1} is found by comparison of the $\pi_{10}^{(0)}$ and $\pi_{10}^{(1)}$.

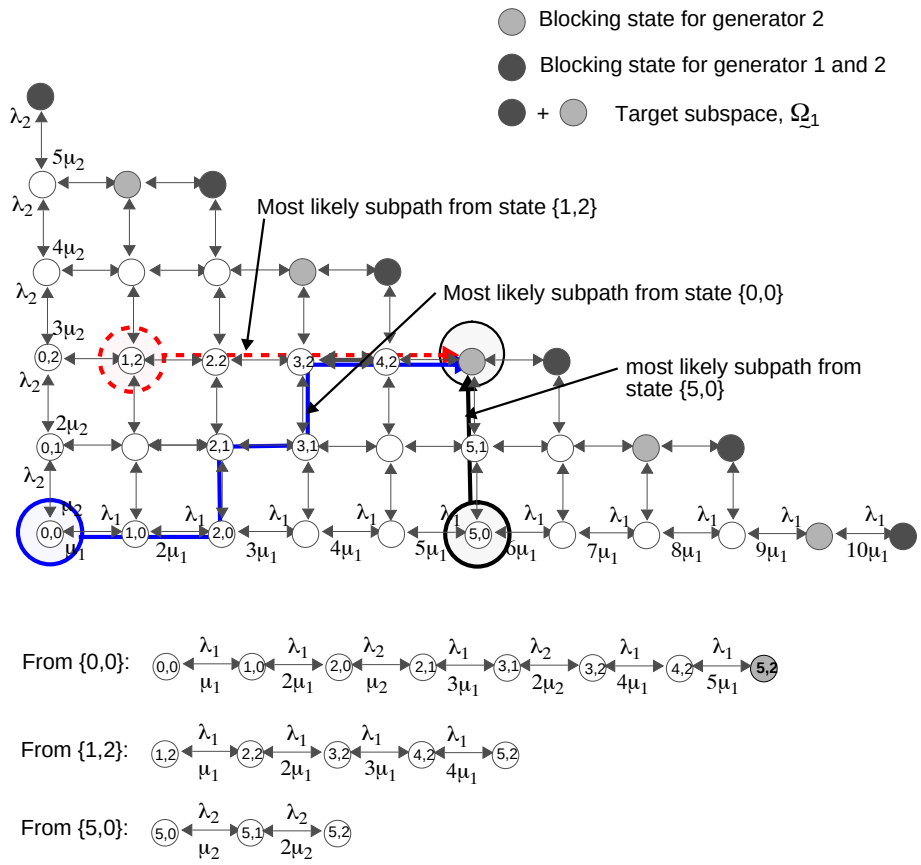


Figure E.3: The most likely path in a 2 dimensional state space, given 3 different starting states. See tables E.2-E.4 for parameters and details.

Table E.5: Search for a subpath from {3,1} for priority level $p = 0$.
The most likely target is $\pi_{10}^{(0)}$.
(the entries marked with an asterisk are not a part of the most likely sub-path).

x	$\underset{\sim}{\omega}^{(x)}$	k_x	π_x	G_x	π_x / G_x
2	{0,1}	-	1	1	1
4	{0,2}	2	0.05	1.05	0.0476
6	{0,3}	2	1.67e-3	1.0517	1.58e-3
8	{0,4}	2	4.17e-5	1.0517	3.96e-5
10	{0,5}	2	8.33e-7	1.0517	7.92e-7

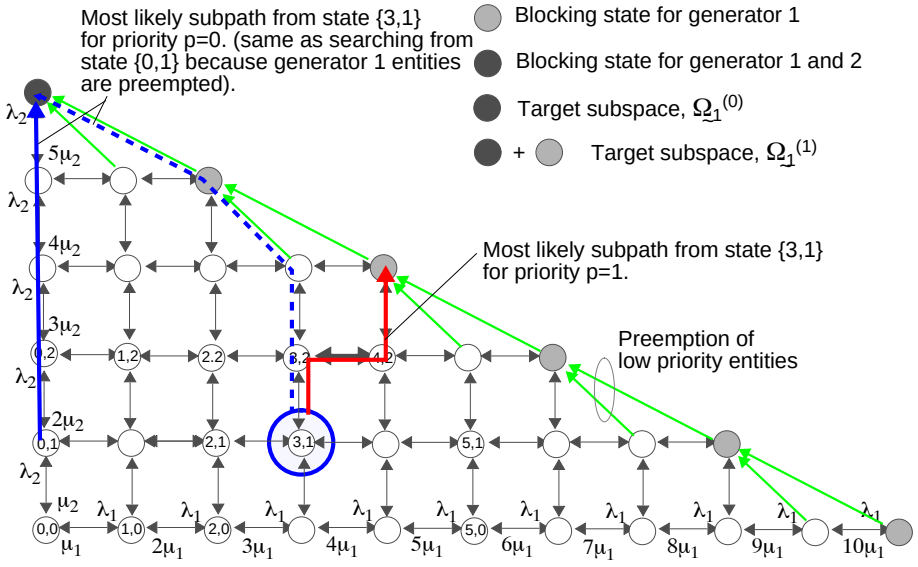


Figure E.4: Most likely subpaths from state {3,1} in a model with two priority levels. One path for each priority level.

Table E.6: Search for subpath from {3,1} for priority level $p = 0$. The most likely target is $\pi_{10}^{(1)}$. (the entries marked with an asterisk are not a part of the most likely sub-path).

x	$\underset{\sim}{\omega}^{(x)}$	k_x	π_x	G_x	π_x / G_x
5	{3,1}	-	1	1	1
6*	{4,1}	1	0.25	1.25	0.2
7	{3,2}	2	0.05	1.05	4.76e-2
8	{4,2}	1	1.25e-2	1.0625	1.18e-2
9*	{5,2}	1	2.50e-3	1.065	2.35e-3
10	{4,3}	2	4.17e-4	1.063	3.92e-4

Appendix F

Rough dimensioning of network resources

In chapter 6, two simulation experiments were conducted on a system originated from the network example depicted in figure F.1 below. In this appendix, the routing and dimensioning of this network are described.

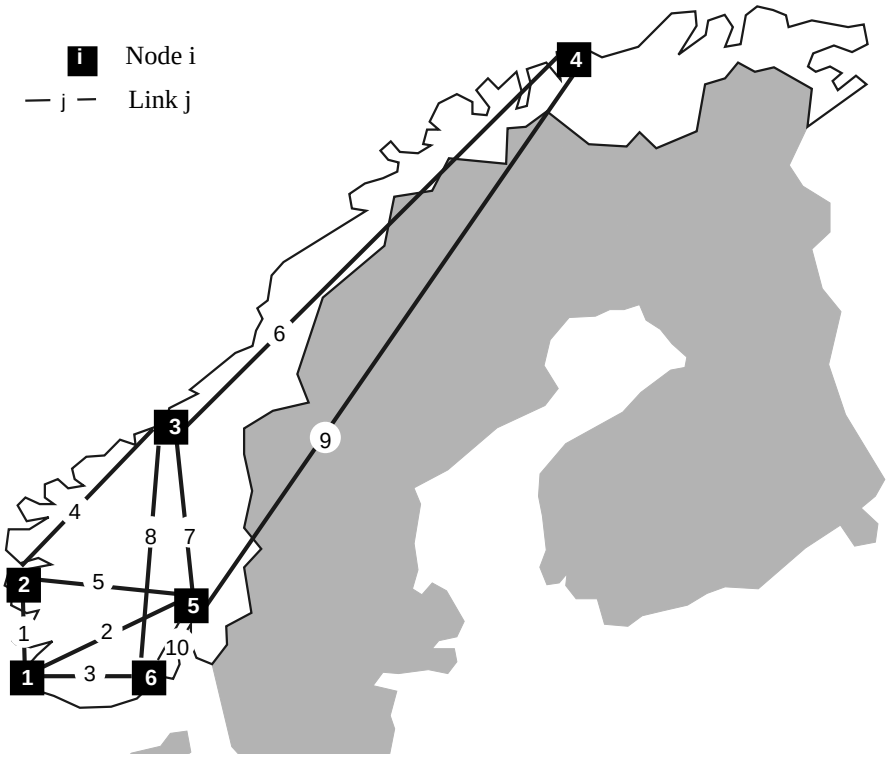


Figure F.1: Map of nodes and link connection.

It is emphasised that the example is a fictitious network although the map and the population assumptions made later are related to a specific country, namely Norway.

F.1 Topology

The network comprises 6 nodes and 10 links, the topology is given in figure F.1. Table F.1 contains the naming of the 6 nodes and the provinces (no: fylke) served by these.The traffic load on each node is assumed to be dependent on the total population served by the nodes. A population vector is placed in column to the right in table F.1.

Table F.1: Node names and placements, and the population vector.

Node	Name	Provinces served	Population vector [1000]
1	Stavanger	Rogaland, Vest-Agder	504
2	Bergen	Hordaland, Sogn- og Fjordane	530
3	Trondheim	Møre- og Romsdal, Sør-Trøndelag, Nord-Trøndelag	624
4	Tromsø	Nordland, Troms, Finnmark	468
5	Oslo	Hedemark, Oppland, Akershus,Oslo	1287
6	Larvik	Østfold, Vestfold, Buskerud, Telemark, Aust-Agder	933

F.2 Routing

Figure F.1 shows the numbering of the 6 nodes and the 10 links interconnecting these nodes. At least 2 link disjoint routes between all nodes exists.

Primary and secondary routes are established after first assigning cost factors to each link, and then finding the minimum cost between every pair of nodes. The cost consists of the following factors:

- *Topology factor* - extra costs are added for fjords and mountains (high enterprise costs).
- *Distance factor* - measured (air route) distance between the end nodes of a link.
- *Population factor* - extra costs are added in proportion to the population served by the end node of a link (heuristics: avoid a link adjacent to a heavy loaded node).

Hence, *cost factor = topology*distance*population factor.* The cost factors used in the reference example are given in table F.2.

The primary routes between each pair of nodes in the network are the routes having the minimum cost. The routes are the same in both directions. The secondary routes are link

Table F.2: Link cost factors.

Link	Topology factor	Distance factor	Population factor	Total cost
1	2.3	1.5	61.46	212.05
2	4.4	1	149.25	656.71
3	3.3	1	108.20	357.06
4	6.2	1.5	76.10	707.71
5	4.4	1	156.95	690.59
6	11.7	1	67.20	786.19
7	5.8	1	184.79	1071.77
8	6.9	1	133.96	924.33
9	16.8	1	138.59	2328.33
10	1.5	1	276.29	414.44

disjoint from the primary routes, and they are found by removing the primary route and finding the minimum route among the remaining. Such a procedure will not necessary find the optimal pair of primary and secondary routes with respect to the minimum cost. However, optimal routing strategies are not the topic in this thesis, so for the purpose of establishing a reference example this procedure is sufficient. The primary and secondary routes that are included in tables F.3 and F.4, respectively, are obtained by use of bulid-in functions in *Mathematica* [Wol91]. For example, the primary route between node 2 and 5 is via link 5, while the secondary route is via links 1 and 2.

Table F.3: The links of the primary routes between node i and j.

from i to j	2	3	4	5	6
1	1	1, 4	1, 4, 6	2	3
2		4	4, 6	5	1, 3
3			6	7	8
4				6, 7	6, 8
5					10

F.3 Traffic matrix

The traffic matrix below contains the traffic $\rho = \lambda/\mu$ between all end nodes in the network. The traffic in table F.5 is assumed to be proportional to the sum of the population associated with the end nodes, see the population vector in table F.1.

Table F.4: The links of the secondary routes between node i and j.

from i to j	2	3	4	5	6
1	2, 5	3, 8	2, 9	3, 10	2, 10
2		1, 3, 8	5, 9	1, 2	5, 10
3			7, 9	8, 10	4, 1, 3
4				9	9, 10
5					2, 3

Table F.5: End to end traffic between node i and j.

from i to j	2	3	4	5	6
1	0.64	0.76	0.57	1.57	1.14
2		0.80	0.60	1.65	1.19
3			0.71	1.94	1.41
4				1.45	1.05
5					2.90

The traffic offered is scaled in accordance to the resource capacity, c_{kj}. For instance, the high priority traffic has $c_{kj} = 4$ for all k and j. This means that the traffic ρ_{ij} in table F.1 must be reduced by a factor of 4, e.g. the traffic between node 2 and 5 is 1.65/4 = 0.41.

The high and low priority traffic offered to each of the links are given in table F.6. The traffic values are derived from table F.5 and the routing sets.

Table F.6: The high and low priority traffic on each link.

link, j	high priority traffic	low priority traffic	proposed link capacity, N_j
1	0.79	3.85	33
2	0.39	6.90	33
3	0.58	7.43	37
4	0.68	1.41	27
5	0.41	2.44	25
6	1.10	0	30
7	0.85	0.70	28
8	0.62	3.50	30
9	0	4.38	21
10	0.72	6.89	37

F.4 Rough link dimensioning

Based on the traffic in table F.5, and the primary routes of table F.3, the traffic on each link can be summarised, see table F.6. The table also includes the proposed capacity of each link.

In tables F.9-F.11, rough estimates of the blocking probabilities for each generator are included, using the resource capacities from table F.6. The estimates are established by ignoring the correlation between the blocking on different links. Hence, it is assumed that each link can be studied separately. Each entry in tables F.9-F.11 is obtained under this assumption and by application of Iversen's convolution method [Ive87]. The final estimates of the blocking of generator k are assigned to the sum of the blocking at each link in along its route. This is a rough, but fair approximation when the blocking probabilities are very low (<1e-06). Some of the values in tables F.10-F.11 are larger than 1e-06, and it is therefore assumed that these estimates are larger than the true value. Hence, these blocking probabilities may serve as an upper bound for the blocking estimates obtained by simulations in chapter 6.

In the tables F.7 and F.8, estimates of the blocking in the case 3.1 and 3.2 in chapter 6 are given.

Table F.7: An estimate of the blocking probabilities for case 3.1 in chapter 6: low priority traffic only.

k	pool 1	pool 2	Σ = pool 1 + pool 2
23	3.21e-09	7.87e-07	7.90e-07
31	3.02e-08	6.05e-06	6.08e-06

Table F.8: An estimate of the blocking probabilities for case 3.2 in chapter 6: Mixed with high priority traffic.

k	pool 1	pool 2	Σ = pool 1 + pool 2
23	3.13e-05	1.41e-04	1.72e-04
31	1.62e-04	7.13e-04	8.76e-04

F.4.1 Approximate blocking in the high priority traffic generators

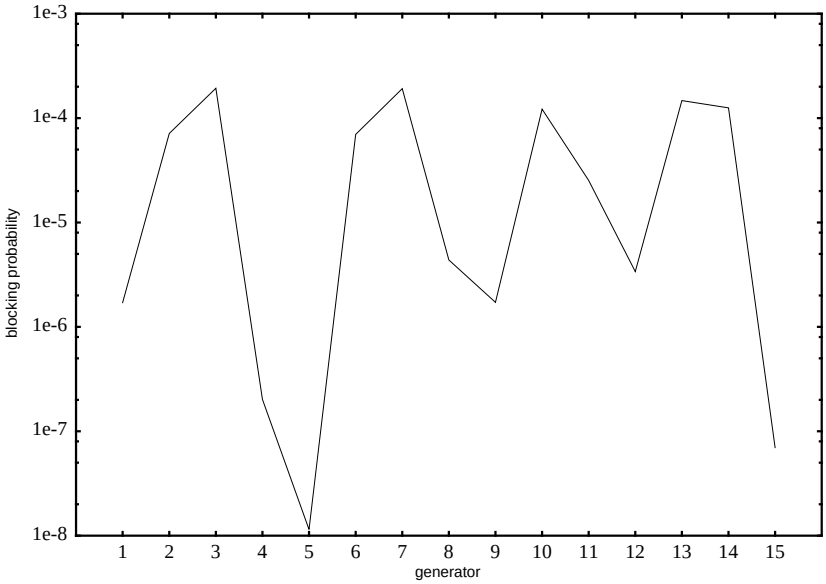


Figure F.2: Plot of the summarised blocking for generator 1-15 from table F.9.

Table F.9: A rough estimate of the blocking probabilities.

k	pool 1	pool 2	pool 3	pool 4	pool 5	pool 6	pool 7	pool 8	pool 9	pool 10	Σ
1	1.71e-6	0	0	0	0	0	0	0	0	0	1.71e-6
2	1.71e-6	0	0	6.96e-5	0	0	0	0	0	0	7.13e-5
3	1.71e-6	0	0	6.96e-5	0	1.22e-4	0	0	0	0	1.93e-4
4	0	2.02e-7	0	0	0	0	0	0	0	0	2.02e-7
5	0	0	1.15e-8	0	0	0	0	0	0	0	1.15e-8
6	0	0	0	6.96e-5	0	0	0	0	0	0	6.96e-5
7	0	0	0	6.96e-5	0	1.22e-4	0	0	0	0	1.92e-4
8	0	0	0	0	4.38e-6	0	0	0	0	0	4.38e-6
9	1.71e-6	0	1.15e-8	0	0	0	0	0	0	0	1.72e-6
10	0	0	0	0	0	1.22e-4	0	0	0	0	1.22e-4
11	0	0	0	0	0	0	2.53e-5	0	0	0	2.53e-5
12	0	0	0	0	0	0	0	3.39e-6	0	0	3.39e-6
13	0	0	0	0	0	1.22e-4	2.53e-5	0	0	0	1.47e-4
14	0	0	0	0	0	1.22e-4	0	3.3e-6	0	0	1.25e-4
15	0	0	0	0	0	0	0	0	0	6.97e-8	6.97e-8

F.4.2 Approximate blocking in the low priority traffic generators

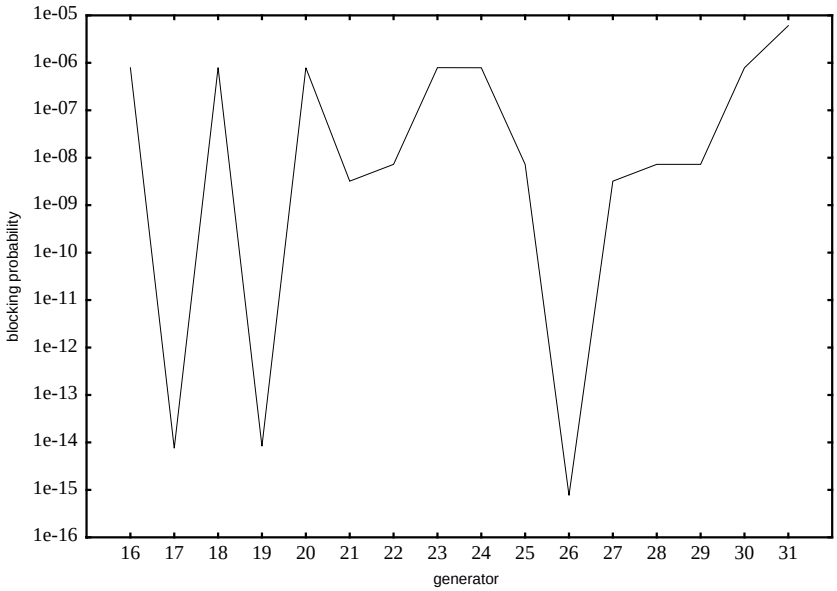


Figure F.3: Plot of the summarised blocking for generator 16-31 from table F.10.

Table F.10: A rough estimate of the blocking probabilities.

k	pool 1	pool 2	pool 3	pool 4	pool 5	pool 6	pool 7	pool 8	pool 9	pool 10	Σ
16	0	7.87e-07	0	0	4.04e-24	0	0	0	0	0	7.87e-07
17	0	0	7.56e-15	0	0	0	0	2.39e-18	0	0	7.56e-15
18	0	7.87e-07	0	0	0	0	0	0	7.25e-09	0	7.94e-07
19	0	0	7.56e-15	0	0	0	0	0	0	7.64e-16	8.32e-15
20	0	7.87e-07	0	0	0	0	0	0	0	7.64e-16	7.87e-07
21	3.21e-09	0	7.56e-15	0	0	0	0	2.39e-18	0	0	3.21e-09
22	0	0	0	0	4.04e-24	0	0	0	7.25e-09	0	7.25e-09
23	3.21e-09	7.87e-07	0	0	0	0	0	0	0	0	7.90e-07
24	0	7.87e-07	0	0	0	0	0	0	0	7.64e-16	7.87e-07
25	0	0	0	0	0	0	1.10e-34	0	7.25e-09	0	7.25e-09
26	0	0	0	0	0	0	0	2.39e-18	0	7.64e-16	7.66e-16
27	3.21e-09	0	7.56e-15	2.40e-25	0	0	0	0	0	0	3.21e-09
28	0	0	0	0	0	0	0	0	7.25e-09	0	7.25e-09
29	0	0	0	0	0	0	0	0	7.25e-09	7.64e-16	7.25e-09
30	0	7.87e-07	7.56e-15	0	0	0	0	0	0	0	7.87e-07
31	3.02e-08	6.05e-06	0	0	0	0	0	0	0	0	6.08e-06

F.4.3 Approximate blocking in the low priority traffic generators mixed with high priority traffic

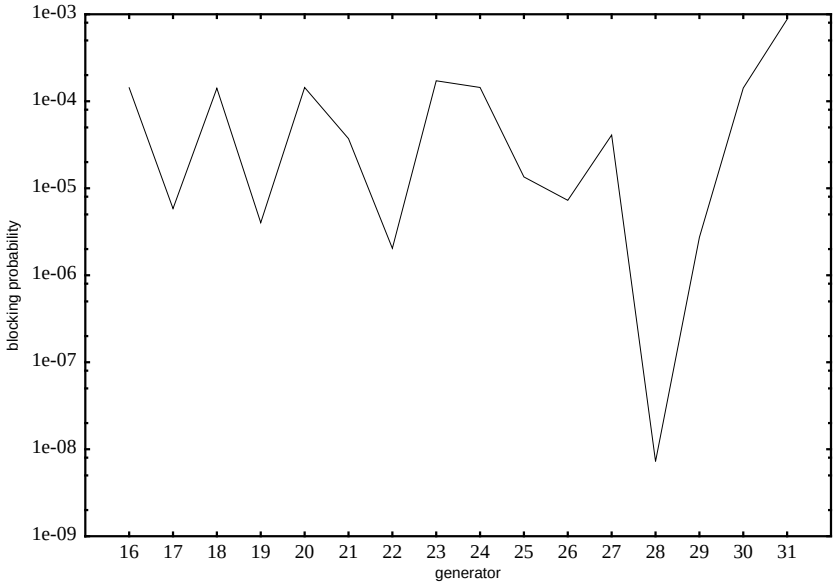


Figure F.4: Plot of the summarised blocking for generator 16-31 from table F.11.

Table F.11: A rough estimate of the blocking probabilities.

k	pool 1	pool 2	pool 3	pool 4	pool 5	pool 6	pool 7	pool 8	pool 9	pool 10	Σ
16	0	1.41e-04	0	0	2.03e-06	0	0	0	0	0	1.43e-04
17	0	0	1.29e-06	0	0	0	0	4.55e-06	0	0	5.84e-06
18	0	1.41e-04	0	0	0	0	0	0	7.25e-09	0	1.41e-04
19	0	0	1.29e-06	0	0	0	0	0	0	2.72e-06	4.02e-06
20	0	1.41e-04	0	0	0	0	0	0	0	2.72e-06	1.44e-04
21	3.13e-05	0	1.29e-06	0	0	0	0	4.55e-06	0	0	3.72e-05
22	0	0	0	0	2.03e-06	0	0	0	7.25e-09	0	2.04e-06
23	3.13e-05	1.41e-04	0	0	0	0	0	0	0	0	1.72e-04
24	0	1.41e-04	0	0	0	0	0	0	0	2.72e-06	1.44e-04
25	0	0	0	0	0	0	1.35e-05	0	7.25e-09	0	1.35e-05
26	0	0	0	0	0	0	0	4.55e-06	0	2.72e-06	7.27e-06
27	3.13e-05	0	1.29e-06	8.27e-06	0	0	0	0	0	0	4.09e-05
28	0	0	0	0	0	0	0	0	7.25e-09	0	7.25e-09
29	0	0	0	0	0	0	0	0	7.25e-09	2.72e-06	2.73e-06
30	0	1.41e-04	1.29e-06	0	0	0	0	0	0	0	1.42e-04
31	1.62e-04	7.13e-04	0	0	0	0	0	0	0	0	8.76e-04

Appendix G

Details from the simulations of the system examples in chapter 6

This appendix contains some details of the direct and importance sampling simulations reported in chapter 6. The results provide the basis for computing the speedups. The tables contain the following:

$\hat{\gamma}$ the estimated property of interest from (2.2),

$S_{\hat{\gamma}} = \sqrt{\mathrm{Var}_f(\hat{\gamma})}$
the standard error of $\hat{\gamma}$, i.e. the square root of (2.3),

$\neg z$ the fraction of regenerative cycles where non-zero observations of the property of interest are made,

R the number of regenerative cycles,

t the elapsed CPU time,

$m_R = 1/(S_{\hat{\gamma}}^2 \cdot R)$
the efficiency measure where the simulation overhead introduced by importance sampling is not included.

m the efficiency measure from (2.11).

The *speedups* given in the tables are the ratio between the efficiency measures for direct and importance sampling simulations.

Only a few comments on the validity of the estimates and the corresponding speedups are included in the following.

G.1 Case 2.1: Arriving calls are lost

Table G.1: Detailed results and speedups.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	2.18e-04	7.90e-06	2.45e-03	500000	1.11e+03	3.20e+04	1.44e+07
IS	1.80e-04	9.00e-06	5.71e-01	100000	1.05e+03	1.23e+05	1.18e+07
Speedup						3.85e+00	8.17e-01

G.2 Case 2.2: Arriving calls are connected via secondary route

Table G.2: Detailed results and speedups.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	8.10e-07	2.10e-07	1.47e-05	1500000	4.81e+03	1.51e+07	4.71e+09
IS	3.14e-07	3.20e-08	4.32e-02	100000	4.01e+03	9.77e+09	2.43e+11
Speedup						6.46e+02	5.16e+01

G.3 Case 3.1: Low priority traffic only

In this case the blocking probabilities are in the order of 10^{-7}. A direct simulation experiment including 75 million cycles was conducted. After more than 5 CPU days, the estimates were more than 1 order of magnitude less than the importance sampling estimates. The importance sampling estimates were in the same order of magnitude as the approximate blocking values given in table F.7 in appendix F. Hence, the speedups given in tables G.3 and G.4, are misleading as an indication of the speedup given by importance sampling.

Table G.3: Detailed results and speedups, generator 23.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	6.81e-09	2.49e-09	2.27e-07	75000000	4.40e+05	2.15e+09	3.66e+11
IS	2.70e-07	1.86e-08	1.14e-01	100000	2.53e+03	2.89e+10	1.14e+12
Speedup						1.35e+01	3.12e+00

Using the *relative error* in the efficiency measures m_R and m, instead of the sample variance, a tremendous speedup is observed, see tables G.5 and G.6.

Table G.4: Detailed results and speedups, generator 31.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	7.05e-08	1.10e-08	9.47e-07	75000000	4.40e+05	1.11e+08	1.89e+10
IS	2.41e-06	1.84e-07	4.50e-01	100000	2.53e+03	2.95e+08	1.17e+10
Speedup						2.67e+00	6.20e-01

Table G.5: Detailed results and relative error speedups, generator 23.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	6.81e-09	2.49e-09	2.27e-07	75000000	4.40e+05	9.96e-08	1.70e-05
IS	2.70e-07	1.86e-08	1.14e-01	100000	2.53e+03	2.11e-03	8.33e-02
Speedup						2.12e+04	4.91e+03

Table G.6: Detailed results and relative error speedups, generator 31.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	7.05e-08	1.10e-08	9.47e-07	75000000	4.40e+05	5.50e-07	9.37e-05
IS	2.41e-06	1.84e-07	4.50e-01	100000	2.53e+03	1.72e-03	6.78e-02
Speedup						3.12e+03	7.24e+02

G.4 Case 3.2: Low priority mixed with high priority traffic

Table G.7: Detailed results and speedups, generator 23.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	1.60e-05	1.90e-06	7.67e-04	150000	4.99e+03	1.85e+06	5.55e+07
IS	2.95e-05	7.10e-06	2.18e-01	10000	4.92e+03	1.98e+06	4.03e+06
Speedup						1.07e+00	7.25e-02

Table G.8: Detailed results and speedups, generator 31.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	7.99e-05	5.80e-06	2.56e-03	150000	4.99e+03	1.98e+05	5.96e+06
IS	1.22e-04	1.90e-05	5.63e-01	10000	4.92e+03	2.77e+05	5.62e+05
Speedup						1.40e+00	9.44e-02

G.5 Case 3.3: Low priority mixed with high priority traffic and exposed to link failures

Table G.9: Detailed results and speedups, generator 23.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	1.60e-05	1.90e-06	7.67e-04	150000	5.26e+03	1.85e+06	5.26e+07
IS	2.10e-05	5.00e-06	2.28e-01	10000	4.94e+03	4.00e+06	8.10e+06
Speedup						2.17e+00	1.54e-01

Observe that the direct simulation approach produced the same estimates in cases 3.2 and 3.3. However, the CPU time is increased by 5.4% because the state space is increase from $K = 31$ to $K = 32$ dimensions. In the direct simulation of case 3.3, no link failures were observed.

Table G.10: Detailed results and speedups, generator 31.

Type	$\hat{\gamma}$	$S_{\hat{\gamma}}$	$\neg z$	R	t [sec.]	m_R	m
Direct	7.99e-05	5.80e-06	2.56e-03	150000	5.26e+03	1.98e+05	5.65e+06
IS	1.48e-04	5.00e-05	5.48e-01	10000	4.94e+03	4.00e+04	8.10e+04
Speedup						2.02e-01	1.43e-02