ABSTRACT

HSU, CHIH-CHIEH. Efficient Evaluation of Highly Available Services: Fast Simulation and Testing. (Under the direction of Professor Michael Devetsikiotis).

Modern technologies have provided us with highly available services. Systems such as optical backbone networks, robust web servers, and reliable software can provide a service with unavailability probability lower than 10^{-6}. Although rare, service unavailability can cause serious problems such as significant performance drop, or violation of Service Level Agreements (SLA). Moreover, providers of these services need to know the value of service unavailability probability so they can provide reasonable SLAs and corresponding Quality of Service (QoS). However, due to the extremely low values of the service unavailability probabilities, estimating them using traditional simulation or testing methods can require a vast amount of time to obtain a satisfactory confidence interval. As a result, efficient evaluation techniques are necessary.

In this dissertation, we propose efficient evaluation methods based on importance sampling (IS). For fast simulation, we introduce several types of IS tuning methods: Our static IS method, which is based on asymptotically efficient IS biasing methods for a single queue, is proven to have bounded relative error. Our adaptive IS method, which is based on guidelines of "optimal biasing", is efficient and can be widely employed. Moreover, IS methods that are stochastically optimized by simulated annealing can be used when the system is complicated, or when the knowledge of the system is limited. Finally, for performance evaluation and optimization of a system under various parameter settings, we propose a framework based on IS and metamodeling methodologies. All of these methods provided in this dissertation are verified by either proof or simulation to be both accurate and efficient.

Efficient Evaluation of Highly Available Services:
Fast Simulation and Testing

by

Chih-Chieh Hsu

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Engineering

Raleigh, NC

2006

Approved By:

Dr. Do Young Eun	Dr. Harry Perros
Dr. Yannis Viniotis	Dr. Stephen Roberts

Dr. Michael Devetsikiotis
Chair of Advisory Committee

To my family, for all the love and unconditional support.

Biography

Chih-Chieh (Jay) Hsu was born in Pan-Chiao, Taiwan on January 1976. He received the B.S. and M.S. degrees from National Taiwan University, Taipei, Taiwan, in 1998 and 2000, majoring in Electrical Engineering and Communications, respectively. Since 2003, he has been pursuing his Ph.D. degree, majoring in Computer Engineering, at the department of Electrical and Computer Engineering in North Carolina State University. His research interests include performance evaluation of highly available services and systems, efficient simulation and testing methodologies, stochastic approach to performance measurement, and modeling and optimization of modern communication networks.

Acknowledgements

First and foremost, I would like to give my sincere appreciation to my advisor, Dr. Michael Devetsikiotis, for his guidance, patience, and support in every way during my PhD study. Dr. Devetsikiotis has brought me into the real world of academic research, and together we solved many interesting problems. This would undoubtedly be the most precious experience for my future career.

I would also like to express my gratitude to all the members in my advisory committee: Dr. Do Young Eun enriched my mathematical background, Dr. Steve Roberts solidified my basic and advanced simulation techniques, Dr. Harry Perros helped me verify my proposed methods against the numerical examples done by his team, and Dr. Yannis Viniotis helped to provide the testbed environment for me to apply my methods on. Without the help and valuable advises from them, my PhD research would be impossible to finish.

In addition, I would like to thank many people from IBM, especially Dr. Andy Rindos and Dr. Steve Woolet. During our cooperation, they shared their knowledge and experiences on practical software performance testing with me and enriched the scope of this dissertation.

Finally, I would like to thank my mother and my sister for all their endless love and support, even though we were thousands miles away from each other. I would also like to express my gratitude to my late father, Ching-Huei Hsu, for it is his ideals of life and philosophy that have made me who I am today. Last but not least, I would like to thank my girl friend, Chian-Fen, for all her love, understanding, care, and encouragements.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Modern communication and computer systems are, or required to be, able to provide highly available services [55]. Examples of such systems include robust network servers, optical networks, reliable software, next generation wireless networks, and large computing grids. In such systems, high availability can be achieved either by low component failure rates (also in the sense that average fault recover time is much smaller than mean time between component failures), or by using a very large capacity or redundancy of resources.

In systems that provide highly available services, unavailability due to complete system failure or lack of capacity can become *rare events* (in fact, this is routinely expected from such systems). In such cases, simulation or actual system testing based on standard Monte-Carlo methods may require an extremely long runtime, and usually incurs large relative errors. *Importance Sampling* (see [28]) has been known as a technique to improve the accuracy of estimates of stochastic events, which permits large speed-ups of estimation of extremely low failure probabilities. The system under study is simulated or emulated in a way that the "important" events occur more frequently by "biasing" the underlying probability distribution. In this dissertation, we explore and propose methods based on importance sampling for efficient simulation and testing of systems that provide highly available services.

In this chapter, first we will introduce several examples of highly available services that will be used in this dissertation. Then the motivation and contributions of our research works will be summarized. Furthermore, an outline of all the chapters in this dissertation

will be illustrated in the last section.

1.1 Highly Available Services

Highly available services are usually encountered in systems which can provide services with very low unavailability probabilities, generally between 10^{-6} and 10^{-9}, or even less. In this section, several illustrations of highly available services are introduced, including optical networks, robust servers, and reliable software. These illustrations of highly available services will be used as examples and for simulation experiments throughout this dissertation.

1.1.1 Optical networks

Traffic groomed optical networks

Modern optical techniques such as wavelength division multiplexing (WDM) have enabled the capability of carrying several Terabits per second by using multiple wavelengths, each of which can carry traffic streams at the order of Gigabits per second, in each fiber. However, in many cases, a traffic stream may only need a small fraction of the wavelength. *Traffic grooming* (see [16]) technique allows the bandwidth of a wavelength to be divided into smaller sub-rate capacities called sub-wavelength units. A customer can require one or more sub-wavelengths to a maximum of the bandwidth of a wavelength. The nodes that connect the links are add/drop multiplexors (ADMs). An ADM is the place where some of the traffic goes through while some other is dropped, which means that the traffic stream is directed to local traffic. Meanwhile, new traffic may be added from local sources, if there is sufficient capacity remaining.

A new connection, usually referred as a new *call*, may arrive at any node and have a destination of any other node. Along its path, it will require some amount of the sub-wavelength units according to its bandwidth requirement. If the requirement of a call cannot be fulfilled when it arrives, the call will be blocked. The probability that a new call will be blocked is, thus, an important indicator of the service availability provided by this network. Therefore in the context of service availability, we refer to call blocking as an event of unavailability. In large optical networks, call blocking events can be rare, due to

either the very low call arrival rate, or to the very large capacity of the network.

An example of a traffic groomed optical network can be found in [31]. For validation with numerical methods as in [59], consider a tandem network of multi-rate loss models with simultaneous resource possession. For example, Figure 1.1 illustrates an optical network with seven nodes and six links. In our simulation method, this topology can be extended to a more generalized mesh network without any difficulty.

Figure 1.1 also shows the possible routes in the network; that is, traffic of a call may arrive at any node a, and leave at any other node b, where b is to the right of a. An incoming call may require one or more sub-wavelength units along its path, depending on the demand of that call. All of these sub-wavelength units are assigned to the call simultaneously at the time it is accepted. If a call is not accepted, it is considered *blocked*. When a call departs, all sub-wavelength units on all of the links along its route are simultaneously released.

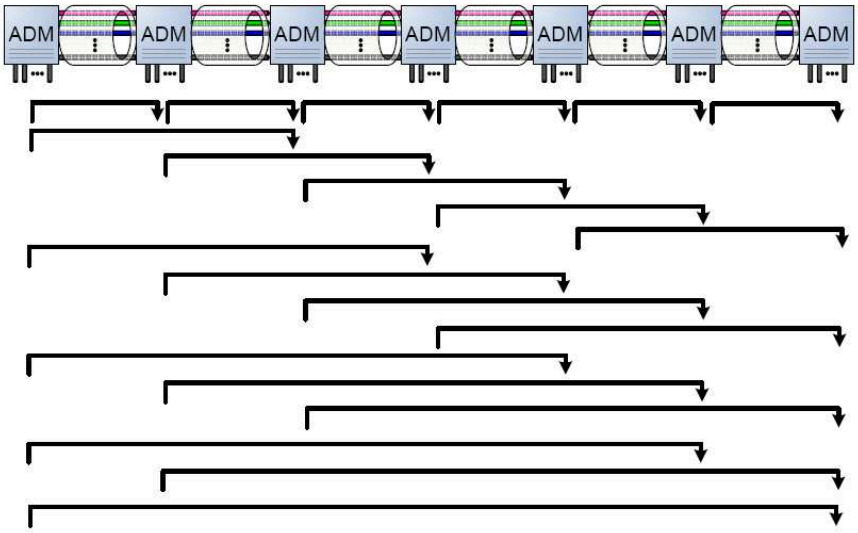

Figure 1.1: A seven-node optical network with all possible routes

Optical burst switching networks

In traffic groomed optical networks introduced in the previous section, a call will obtain all the required sub-wavelength units simultaneously at the time it is accepted. This is referred to as *static simultaneous resource possession*. There are some cases in which the

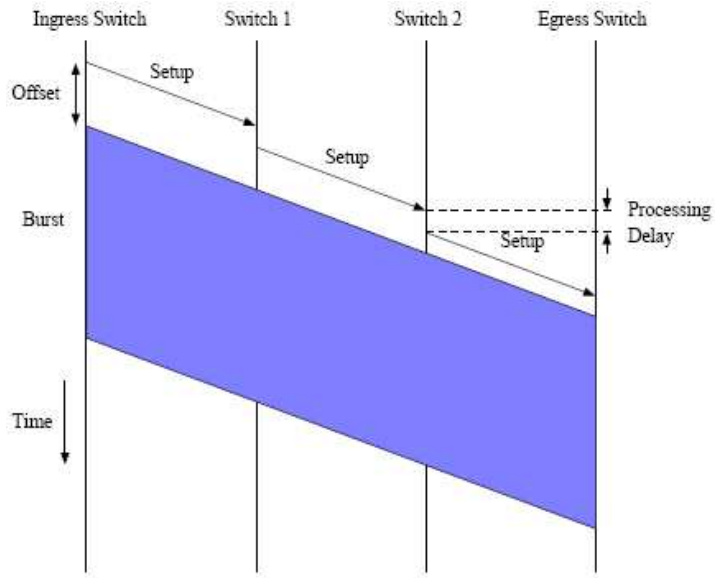

Figure 1.2: OBS network operation.

resources can be assigned and released dynamically as the traffic of the call travels, which is referred to as *dynamic simultaneous resource possession*. For example, in a relatively new optical switching scheme known as *optical burst switching* (OBS), multiple but not all resources can be reserved simultaneously [5].

In an OBS network, data packets are aggregated into various sizes of "bursts" at the edge routers, according to their destination. A burst could be a number of IP packets, an entire file, or frames from a video. Before a burst is transmitted, a control (or setup) signal, which contains the information of the burst such as burst length and expected arrival time, is sent along the same route to the destination as the burst itself to reserve the bandwidth (wavelength) along this route. A node in this route will reserve a wavelength for this burst at or after the control packet arrives, according to the reservation method used. After a time offset, the burst itself will be sent without having any acknowledgement from any of the nodes along the path. Figure 1.2 shows the basic operation of an OBS network.

There are many different wavelength reservation schemes that have been introduced. Among them the Just-in-time (JIT) and Just-enough-time (JET) are most popular. In JIT, a wavelength is reserved at the time the control signal arrives, while in JET a wavelength is reserved at the time the burst arrives. In both cases, a wavelength on a link will be released

when the burst fully passes one of the end nodes of that link.

If a burst is long enough, it is possible that the burst may occupy more than one links simultaneously. That is, it is possible that the burst may occupy a link between adjacent nodes A and B, and at the same time a link between adjacent nodes B and C. As the burst travels through the network, it will release the link that the whole burst has traversed and pick up a new link ahead. Therefore a burst may occupy two or more successive links, but these links change as the burst moves through the network, and thus the resources are dynamically simultaneously possessed.

If we assume that the network has large numbers of wavelengths, such an "one way" reservation mechanism would not sacrifice much performance drop due to burst losses while gain a lot from saving an one-way delay time from the destination to the source. OBS network architecture has been a hot topic for recent years, and there are many different implementation suggests for wavelength reservation and offset setting. However, a dropped burst usually contains mass amount of data and therefore burst dropping probability is considered an important index of service unavailability.

1.1.2 Robust network servers

Following the fast growth of demand for Internet services, network servers have been playing an increasingly important role. For a service provider, a server breakdown event may cause the service unavailable to requests, and possibly huge amounts of capital loss. As a result, redundancy is usually used to prevent servers from breaking down. Techniques such as server clusters [56] are introduced for risk diversification and load balancing. In the meanwhile, even for a single server in the cluster, techniques such as duplicated processors and network interfaces, disk arrays, and backup power sources have become standard in recent network servers. Together with physical redundancy, fast repair teams can also help making breakdown events rare.

Figure 1.3 shows an example of a robust server with duplicate processors, three sets of disk arrays, a multi-interface network controller, and backup power supplies. When a component fails, a redundant one will take over and a repair team will be assigned to repair the failed one. The system is considered breakdown and unavailable to requests whenever all the components of the same kind are fail. In real situations, it is possible that more than one components of same or different kinds will fail at the same time. For example, a

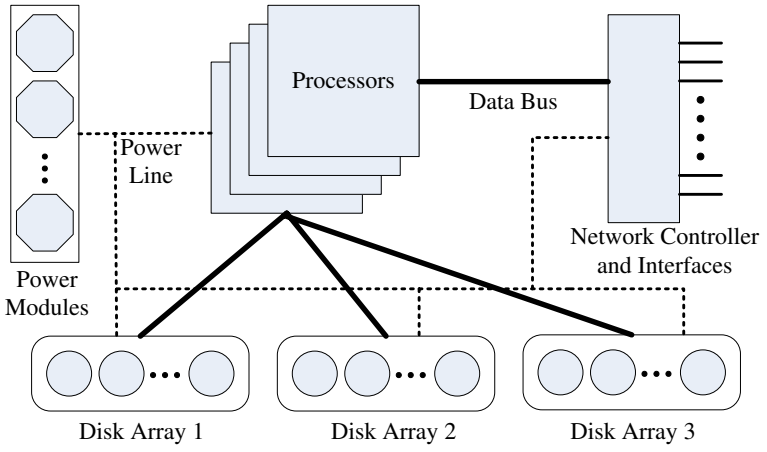

Figure 1.3: An example of robust network servers.

failure of a power supply may also burn out several disks and processors. A single repair team or different teams can be assigned for a multiple component failure, and components failed together can be brought online again together when all failed parts are repaired, or one by one.

1.1.3 Reliable software

Performance evaluation of software could be performed by testing without assuming any model of the software. However, for modern software programs that are highly available and have very low request dropping rates, the use of direct testing could cost a lot of time while still not yielding satisfactory confidence intervals. The use of modeling has several benefits. First, efficient testing techniques could be used by the help of these models. Moreover, if real testing is expensive, simulation using the performance model can be used to assist real testing, so online capacity planning or performance evaluation would be more feasible.

One of the most widely used performance models is the Markov usage model [24]. In this model, a Markov chain is constructed according to standard user behaviors, called *operational profiles*, and the functions provided by the software. Figure 1.4 shows a Markov usage model of a software tool for displaying, sorting and printing items.

The Markov usage model provides detailed user behavior and system states of the

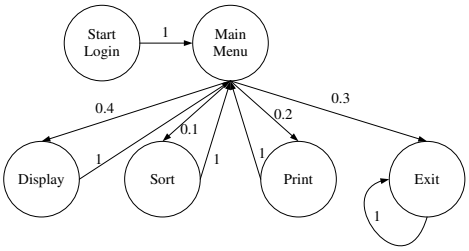

Figure 1.4: A simple example of Markov usage model.

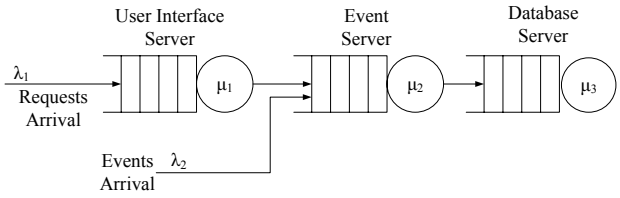

Figure 1.5: Queueing model of a three server software.

software under evaluation. Performance simulation using various techniques can then be applied. However, at the time of evaluation one may not be able to know the detailed behavior of the users. Moreover, the model can be complicated to build, especially when the software is distributed among several physical machines, which is becoming common in modern server software.

Alternatively, a queueing networking model can be considered when trying to model the software behavior. Figure 1.5 shows a tandem queueing network that describes a software that contains three servers: an user interface server, an event server, and a database server. Requests from the user arrive at the user interface server, while the requesting events from other software or trace arrive at the event server. The service time of these queues can be modeled using simple exponential distributions, or in a more generalized manner, coming from a (possibly simplified) model that is similar to the Markov usage model.

1.2 Motivation

For highly available services, including those introduced in the previous section, the service unavailability probability is an important indicator of quality. Due to the rarity of unavailability events, traditional simulation / testing methods are usually not suitable for estimating their probability. Numerical approximation methods have been developed to solve for these indicators. For example, [59] and [6] focus on solving the new call blocking probabilities of optical networks. We would like to develop ways that are both easy to use and accurate in estimating such quality of service indicators, based on Monte Carlo methods and Importance Sampling.

Multi service loss networks are widely used in modeling systems that provide highly available services. For example, [49, 40, 37], and [38] used multi service loss networks to describe these systems. One advantage of such a model is that the likelihood function of importance sampling is easier to calculate, provided that the arrival and service distributions are not too complicated. As a result, we would like to first consider using multi-service loss models in our development of efficient simulation methods.

There are many criteria and goals for the design of importance sampling methods that simulate rare events. Most of these criteria focus on the asymptotic performance of importance sampling methods as the value to be estimated approaches zero. We would like to explore such metrics of performances of our method if possible. On the other hand, since the value to be estimated does not usually approach zero asymptotically, there may exist methods that are better for practical values. In this dissertation we also put efforts in developing methods with these characteristics, especially for those models that are more complex.

If a real system is available, performance testing can also be performed to estimate the service unavailability of the system. Stochastically generated traffic that is fed into the system could be tuned in similar ways as in the importance sampling simulation. The only difference is that the parameters of the system under test will not be available for IS biasing.

It is common for an evaluation project to include statistics for the performance under different parameter settings. As a result, we also seek the possibility of applying metamodeling techniques together with fast simulation/testing methods to allow one to understand the system behavior as a whole, and even optimize its performance, as in the case of capacity planning.

1.3 Contributions of this Dissertation

The main contributions of this dissertation can be listed as the following:

- **Analytically proven IS method for multiple-class, heterogeneous-demand loss systems**

  Many highly available services of interest can be modeled as *multi-service loss networks*, which will be introduced in Chapter 3. For such models, several methods have been proposed to use Importance Sampling in estimating the blocking probabilities, e.g., [49] and [40] focused on the estimation of the most likely blocking link; and Lassila et al. [37, 38] provided methods for the cases in which more than one link may have contribution to the blocking probability. These methods are based on a product form solution and may need to calculate the very large table of transition probabilities or rates before the simulation begins.

  In [58], the authors proposed Importance Sampling applied to the Standard Clock method (ISSC) and showed that in the single-class, homogeneous demand case, when the arrival rates approach zero, ISSC has bounded relative error in the estimation. In this dissertation, we extend this result to a *multi-class* service model with *heterogeneous demands*, and therefore allow different types of simultaneous failures in the system. We prove this extended Static ISSC method still has the bounded relative error property and, thus, can be applied to efficiently evaluate models that are more complex.

- **Better heuristic near-optimum IS method**

  For the case in which the failure rates do not tend to zero, we propose using *Adaptive ISSC*, that tunes the probability distribution toward the most possible target in each step. Using A-ISSC, we avoid calculating excessively large tables in advance and we can extend existing methods for single link failure to more general cases, while still having very low relative estimation errors in the simulation. We also compare our proposed method with existing IS methods on different simulation cases, and prove empirically that our method can produce accurate estimation, and is more efficient than other methods that have already been published.

- **Generally applicable IS method**

There are cases in which the importance sampling biasing can be done, but due to the complicated system model or incomplete knowledge of the system, minimizing the estimation variance becomes very hard, if not impossible. For such cases, we propose minimizing the variance directly using a stochastic optimization method, namely simulated annealing [1]. The proposed Simulated Annealing optimized ISSC (SA-ISSC) can be applied to almost any models, as well as the likelyhood of the IS method can be calculated correctly. Moreover, SA-ISSC is very easy to apply and can produce favorable results.

- **Automatic metamodeling and optimization framework with efficient trace reuse**

  It is not uncommon that one may want to know the performance of a system for more than single parameter configurations. One could even want to find an optimal configuration of the system that optimizes the performance or related indices. To deal with this, metamodeling methods, such as Response Surface Methodology [9], have been proposed for a long time. On the other hand, it is rarely seen that efficient evaluation techniques being used together with such metamodeling methods. In this dissertation, we propose a combined Response Surface-Importance Sampling (RS-IS) framework that automatically models and even optimizes the performance index of interest. Moreover, due to the nature of local sampling in the response surface method, we also propose a novel IS trace reuse strategy that saves even more evaluation time, while still produces very accurate solutions.

1.4 Outline

Figure 1.6 shows the basic structure of this thesis. The left-hand side of the figure shows some examples of highly available services we will evaluate, and the models used to describe these services. The right-hand side, on the other hand, shows the importance sampling-based methods we develop in this thesis to efficiently evaluate these services described by the models.

Figure 1.7 shows the same structure from a hierarchical point of view. The white rectangles show the examples of highly available services and the models we use to describe them. The colored (shaded) rectangles correspond to the different methods developed, and

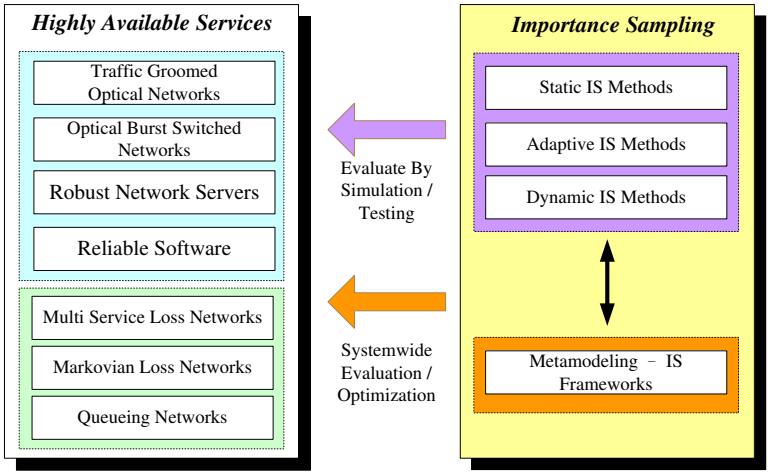

Figure 1.6: The overview structure of this dissertation.

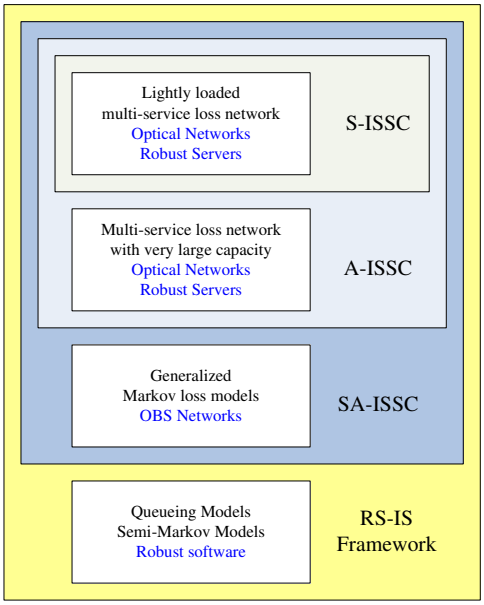

Figure 1.7: The relationship of models and evaluation methods in this dissertation.

the range covered by each rectangle implies the models that such a method can be applied to. Throughout the paper it will be shown that although a method covering more services and models could be applied on all of them, for the models also covered by another smaller rectangle, this more general method is usually not as efficient as the methods denoted by smaller rectangles, for they are generally been optimized according to the model.

The rest of the dissertation is organized as follows. In Chapter 2, we introduce the basic idea of importance sampling, the idea of "optimal" biasing, and some criteria for good IS parameter setting methods. Dynamic IS methods and the method of regressive and A-cycle simulation methods will also be shown. Chapter 3 introduces a multi-service loss network model, following by the a briefly description of the existing IS methods for loss systems, and the generalized S-ISSC method and A-ISSC methods. In Chapter 5, a model for OBS network is introduced, and the SA-ISSC method is proposed to deal with these more complex models. In Chapter 6, the RS-IS automatic framework, which can be used to automatically model and optimize the performance of a highly available service, if introduced. Finally, we show a summary and possible future work in Chapter 7.

Chapter 2

Importance Sampling and Efficient Simulation

2.1 Basic idea of importance sampling

Importance Sampling (IS) is a Monte Carlo (MC) estimation technique which aims to reduce the variance or other cost function of a given simulation estimator. Figure 2.1 illustrates the concept of importance sampling. Assume we want to measure the two-dimensional area of region B by simulation. Traditional Monte-Carlo (MC) simulation generates points in A from a uniform distribution, and the estimation of region B would be $\widehat{B} = \frac{N_B}{N}$, where N_B is the number of "hits" within area B, and N is the total number of samples generated. When the area of B is very small related to the area of A, MC simulation would have a very large variance, due to the very few hits within area B. Therefore, to achieve a satisfactory confidence interval, one has to simulate the system for a very long time. Using IS, we modify our sampling procedure to increase the fraction of samples that result in hits. In this example, if we double the probability that the samples are generated in area D, then the number of hits within B will also be doubled, thus increasing the efficiency of the estimate. However, since we biased the probability, each hit within area B have to be weighted by a factor of $\frac{1}{2}$ to yield a correct, unbiased result.

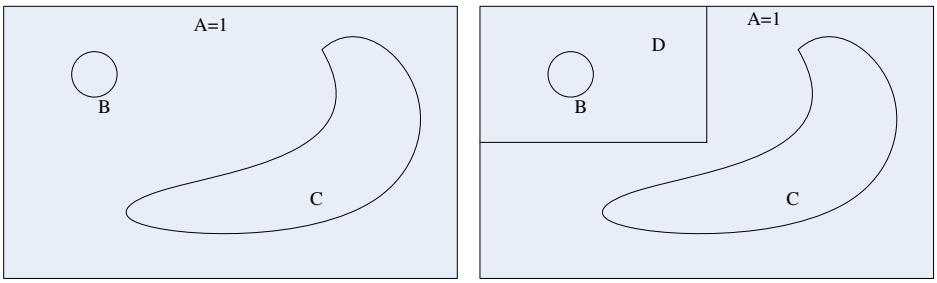

Figure 2.1: Schematic illustration of the concept of IS.

Consider a case in which we want to estimate the probability $P(X \in B)$ for a random variable X with probability density function (pdf) $f(\cdot)$. Traditional Monte-Carlo simulation method generates samples of X and counts the number in B, that is, by estimating $\mathrm{E}[\mathbf{1}_{\{X\in B\}}]$,where $\mathbf{1}_{\{\}}$ is the indicator function, and $E\left[\mathbf{1}_{\{x\in B\}}\right] = P\left(x \in B\right)$. If $P(X \in B)$ is very small, it would require a large number of samples for the estimator to be accurate. This is because the relative error, which is defined to be the standard deviation divided by the mean, $\frac{\sqrt{P(X\in B)(1-P(X\in B))}}{\sqrt{n}P(X\in B)} = \sqrt{\frac{(1-P(X\in B))}{nP(X\in B)}}$ goes to infinity as $P(X \in B) \to 0$. Using IS, we generate samples with a *biased* pdf $f^*(\cdot)$ with $P^*(X \in B) > P(X \in B)$. If each time we observe an x within B, we increment our count by $\frac{f(x)}{f^*(x)}$ instead of 1, then, effectively, we are constructing a new "weighted" random variable, the expectation of which is also equal to $P(X \in B)$:

$$\int\limits_{x\in B} \frac{f(x)}{f^*(x)} f^*(x)dx = \int\limits_{x\in B} f(x)dx = P(X \in B).$$

The function $L(x) \triangleq \frac{f(x)}{f^*(x)}$ is called the Radon-Nikodym derivative, or the likelihood ratio. For a Markovian-type system, if we have a sample path x with M steps, the Radon-Nikodym derivative would be product of the Radon-Nikodym derivative in each step along x, that is,

$$L(x) = \frac{f\left(x\right)}{f^*\left(x\right)} = \prod_{k=0}^{M-1} \frac{f(x_k, x_{k+1})}{f^*(x_k, x_{k+1})} \tag{2.1}$$

where $f(x_k, x_{k+1})$ and $f^*(x_k, x_{k+1})$ denote the original and biased transition probability from state x_k to x_{k+1}, where x_k is the system state at the k^{th} step.

As long as $L(x)$ can be calculated exactly (which also implies that, in the region we want to estimate, $f(x)$ must be absolutely continuous with respect to $f^*(x)$, that is,

whenever $f(x) > 0$, then we must also have $f^*(x) > 0$), the IS estimator is *statistically unbiased*. However, deciding how to choose an appropriate $L(x)$ so that the simulation is most efficient and results in smallest variance is far from trivial, depending on the system of application.

2.2 Optimal biasing parameter and criteria for a good biasing

Consider the following change of measure.

$$f^*(x) = \begin{cases} \frac{f(x)}{P(X \in B)}, & x \in B \\ 0, & otherwise \end{cases} .$$

In other words, this new distribution is simply the original one conditioned on the occuring of the rare event we are interested in estimating. By doing this, we have $L(x)\mathbf{1}_{\{x \in B\}} = P(X \in B)$ for any x obtained in the new sampling distribution, and, thus, the variance will be 0, and only one sample gives us $P(X \in B)$ exactly. Therefore, $f^*(x)$ is the optimal change of measure. Actually, this optimal $f(x)$ can be easily derived directly by using Jensen's inequality [54]. However, since $f^*(x)$ explicitly depends on $P(X \in B)$, the unknown quantity that we are trying to estimate, this change of measure is can not be practically used. If $P(X \in B)$ were known, there would be no need to run the simulation experiment at all. Nevertheless, in the development of efficient importance sampling methods, this "Optimal Biasing" can be used as a guideline in choosing $f^*(x)$. This leads to the following principles for a good $f^*(x)$:

- Choose $f^*(x)$ so that we "hit" the events of interest A as often as possible.
- Choose $f^*(x)$ so that the more likely or higher probability regions of B will be "hit" more often during the simulation than the lower probability or less likely regions of B. That is, we want to find a $f^*(x)$ that the "relative probabilities" of elements inside set B are kept the most.

2.2.1 Bounded relative error (BRE)

Since the optimal change of measure is not feasible, some criteria have to be used to evaluate the efficiency of an IS estimator. A widely used one for evaluating efficiency of

an IS estimator is to determine if it has *bounded relative error (BRE)*, which checks if the relative error, defined as the standard deviation of the estimator divided by the estimator itself, can be bounded as the probability to be estimated approaches zero (see [52] and [28]). The definition is as follows:

Definition 1 *Assume we would like to estimate the probability* $P_\varepsilon(X \in B)$ *of a rare event, where* $P_\varepsilon(X \in B) \to 0$ *as* $\varepsilon \to 0$. *An unbiased IS estimator for* $P_\varepsilon(X \in B)$=$E[L(X)\mathbf{1}_{\{X \in B\}}]$, *has bounded relative error (BRE) under* $f^*(x)$ *if there are constants* $\delta < \infty, \epsilon_0 > 0$ *such that*

$$\sup_{\epsilon \leq \epsilon_0} \frac{\sqrt{\mathrm{Var}^*[L(X)\mathbf{1}_{\{X \in B\}}]}}{P_\varepsilon(X \in B)} \leq \delta.$$

The following lemma is a direct consequence of the above definition [11].

Lemma 2 *If there are constants* α, β *and* γ *such that* $P_\varepsilon(X \in B) \geq \alpha\varepsilon^\gamma$ *and* $L(X)\mathbf{1}_{\{X \in B\}} \leq \beta\varepsilon^\gamma$ *a.s., then the IS estimator for* $P_\varepsilon(X \in B)$ *has BRE.*

We can see that the relative error is actually proportional to the relative half-width of the confidence interval in the simulation. As a result, having an estimator that has BRE is indeed very preferable, since one always needs only a fixed number of iterations to obtain a target relative confidence interval, no matter how rare the corresponding event is.

2.2.2 Asymptotic efficient

Generally, it is not easy to find an IS estimator that has bounded relative error. Another widely used criteria for a good IS estimator is *asymptotical efficiency*. The idea of asymptotically efficient comes from large deviation theory.

As stated in earlier sections,

$$\begin{aligned} NVar(\widehat{P(X \in B)})_n &= \left[E\left\{1^2_{\{X \in B\}} L\left(X\right)\right\} - P(X \in B)^2\right]_n \\ &\triangleq \left(F_B - P(X \in B)^2\right)_n \end{aligned}$$

where n is a rarity index. That is, $P(X \in B)$ goes to 0 as n goes to infinity.

From large deviation theory, we have (under some assumptions)

$$\lim_{n\to\infty} \frac{1}{n} \log(P(X \in B))_n = -I$$
$$\lim_{n\to\infty} \frac{1}{n} \log(P(X \in B))_n = -R$$

Since variance is always greater or equal to zero, we have $R \leq 2I$. If $R = 2I$, this IS method is said to be *Asymptotic Efficient*.

In other words, asymptotic efficient means the relative error will not grow exponentially as the value to be estimated approaches zero.

2.3 Overbiasing problem

When applying importance sampling, choosing of the biasing distributions must be done with care. Carelessly chosen biasing methods may not result in a satisfactory variance reduction. A bad IS design can even enlarge the estimation variance! This is sometimes called a "backfire" in variance reduction methods. In importance sampling, such a problem can be referred to as "overbiasing". Overbiasing is dangerous, since it not only potentially increase the estimating variance, it can even cause error if the simulation is carelessly performed.

For example, consider the following examples. Let $A = \{x_1, x_2\}$, is the area we would like to estimate, where $P(x_1) = 10^{-5}$, and $P(x_2) = 2 \cdot 10^{-5}$. $P(A) = 3 \cdot 10^{-5}$.

The "optimal" biasing that will give us zero variance is

$$P^*(x_1) = \frac{1}{3}, P^*(x_2) = \frac{2}{3},$$

in which both likelihood ratios is equal to $3 \cdot 10^{-5}$.

However, if we use the following biasing,

$$P^*(x_1) = 0.9999, P^*(x_2) = 0.0001$$

we have

$$L(x_1) \sim 10^{-5}, L\,(x_2) = 0.2$$

Although this estimator is still statistically unbiased, in practical we will get x_1 most of the time. If the simulation is not performed long enough, one may have an estimation that $P(A) = 10^{-5}$ with a very low variance, while in fact this estimation is not even near the actual value.

2.4 Dynamic IS simulation techniques

To use importance sampling in rare event simulations, it is good to have an importance sampling distribution with bounded relative error or asymptotical efficient property. However, to have such an distribution, it is usually required to have a good understanding of the system, or the large deviation behavior of the rare event of interest. Therefore, there are many research which aims directly on minimization of the variance of the estimator (for example,[34, 17, 14]). Such techniques are called dynamic IS techniques, or sometimes, adaptive IS techniques.

2.4.1 Stochastic gradient techniques

Let $f_\theta(x)$ denote the family of candidate distributions to be used in importance sampling. To minimize the variance of importance sampling simulation for $P_A = P(X \in A)$ under the original pdf $f(x)$ can be formulated as

$$\min_\theta Var_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)]$$

where $L_\theta(x) = \frac{f(x)}{f_\theta(x)}$. Since

$$\begin{aligned} Var_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)] &= E_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)^2] - E_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)]^2 \\ &= E_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)^2] - P_A^2, \end{aligned}$$

the variance-minimization problem is equivalent to

$$\min_\theta E_{f_\theta}[\mathbf{1}_{\{X\in A\}} L_\theta(X)^2],$$

which is a stochastic optimization problem. Therefore, traditional stochastic approximation algorithms, such as Robbins-Monro algorithm [48] and Kiefer-Wofowitz algorithm [35] can be used.

R-M algorithm uses the following recursion to obtain the optimal θ :

$$\theta_{n+1} = \Pi_\Theta(\theta_n - \frac{a}{n+1}\widehat{\nabla h}(\theta_n))$$

,where Π_Θ is the projection operator onto Θ, $h(\cdot)$ is the objective function (that is, $E_{f_\theta}[\mathbf{1}_{\{X\in A\}}L_\theta(X)^2]$) and $\widehat{\nabla h}(\theta_n)$ is an estimate of ∇h at θ_n. To get $\widehat{\nabla h}(\theta_n)$, several techniques such as infinitesimal perturbation analysis [20], likelihood ratio methods [22], Conditional Monte Carlo [19], and the "push-out" approach [50] can be applied. K-W algorithm also uses the same recursion formula, but use different method to estimate $\nabla h(\theta_n)$.

Using importance sampling for accelerating simulation by finding an approximate minimizer of variance of the estimator has been applied in various applications, especially in queueing and reliability models; see, e.g. [2, 14, 13].

2.4.2 Stochastic optimization techniques

Simulated annealing (SA) [36] is a widely used optimization method that can avoid being trapped in local optimum. In simulated annealing, in addition to the moves that decrease the cost function, a move which causes an increasing cost of ΔC will be taken with some probability related to a parameter c, which is called the *temperature*. From time to time, the temperature is lowered from $T_{\max}$ to $T_{\min}$, thus lowering the probability of accepting uphill moves and forcing the system into a global minimum.

Simulated annealing can be used to minimize IS variance. In [14], a variation of SA, Mean field annealing (MFA) [8], is used to obtain minimum variance of importance sampling simulation for a single queue. The MFA algorithm keeps the ability of SA to avoid local minima, and converges faster than SA, but is not guaranteed to converge.

2.5 Regeneration and A-cycle methods

We are often interested in *steady-state* properties in simulations of stochastic systems. For example, in queueing models, one might be interested in $P(Q > x)$ where Q is the steady state queue size distribution. If the system is regenerative, we can use the *"regenerative method"* to estimate steady state QoS parameters. In a regenerative system, there exists a particular state such that the process will visit it infinitely often, and after the state is visited, the stochastic evolution of the system is independent of the past and has the same

distribution, as if the process were started at the state. The system evolution between two consecutive visits of this state is called a *regenerative cycle*. Let T_i denote the length of the i_{th} regenerative cycle. If $E[T_i] < \infty$, the system has steady state distribution X. Let $h(\cdot)$ be a function on the state space and define $Y_i = \int_{i_{th} \text{ cycle}} h(X_t)dt$, where X_t denotes the sample in time t. Then

$$E[h(X)] = \frac{E[Y_i]}{E[T_i]}.$$

For a highly available service, usually a non-zero Y_i is a rare event. Therefore, importance sampling can be applied to estimate the numerator. Importance sampling is used until a non-zero value of Y_i, occurs, and then it is "turned off", and the system will then return to the regenerative state naturally. The denominator is simply the expected cycle time, so it is not necessary to use importance sampling to estimate the denominator. If under importance sampling, the system is still regenerative, importance sampling may be left "on" when the rare event occurs. However, in many cases the importance sampling distribution is chosen so that the system will not regenerate (even unstable). In such cases, importance sampling should be turned off in order to permit regeneration. In [14], the authors even use another set of importance sampling distribution to force the regeneration to happen.

In some applications, the model may not be regenerative, but a similar formula still exists. For a quasi-regenerative set A which will be visited infinitely often, define A-cycles to begin whenever the process enters A. Then

$$E[h(X)] = \frac{E[Y_i(A)]}{E[T_i(A)]}.$$

where $Y_i = \int_{i_{th} \ A \text{ cycle}} h(X_t)dt$ and $T_i(A)$ is the length of i_{th} A-cycle. Similarly, importance sampling can be used to estimate the numerator, while standard simulation can be used to estimate the denominator. Such method is usually called A-cycle method.

In A-cycle method, a technique sometimes called "backbone and rib" [58] can be used. As illustrated in figure 2.2, an ordinary Monte Carlo simulation is used to estimate the denominator, as well as to provide entry points for importance sampling. In each importance sampling simulation, once the desired rare event happens, IS is turned off and ordinary simulation continues until the system leaves set A. Using this method, importance sampling simulations can be done in parallel, if possible. Moreover, if one is interested in more than one different A sets, the same ordinary Monte Carlo simulation can be used as

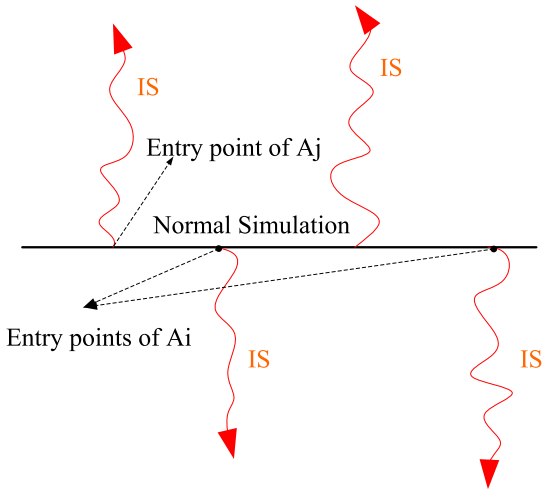

Figure 2.2: "Backbone and rib" A-cycle Method.

the entry points for the different IS simulations. For confidence interval in this method, Jackknife technique may be used.

2.6 Conclusions

In this chapter, the basic idea of importance sampling is introduced. Moreover, several applications of IS, as well as the criteria for good biasing methods and potential problems, are also explained. Our goal in this dissertation is to develop IS based methods that can either be proven to meet some of the criteria, or to have good performance in practical. For the following chapters, we will start to introduce the IS methods proposed in this dissertation.

Chapter 3

Static and Adaptive Importance Sampling for Highly Available Services

Most of the importance sampling techniques mentioned in previous chapter focus on estimating bit error rate or overflow probability in a queue. In other words, there is only one objective that is of interest. For example, if we are interested in the buffer overflow probability of one certain queue in a feed forward queueing network, we will try to change the measure so that buffer overflow in that queue will not be a rare event. This procedure may not be true in real-world highly available services, where a rare event we are interested in may happen in more than one place. For example, an highly available optical network will be well designed so that there will not likely to be a single bottleneck for a call that uses multiple links. Instead, the call blocking, which is a rare event, could happen because any of the links it uses does not have enough wavelengths available.

In this chapter, first we will introduce a multi-service loss network, which can be used to describe the behaviors of the traffic groomed optical network and the robust server introduced in the first Chapter, as well as many other highly available services. In addition,

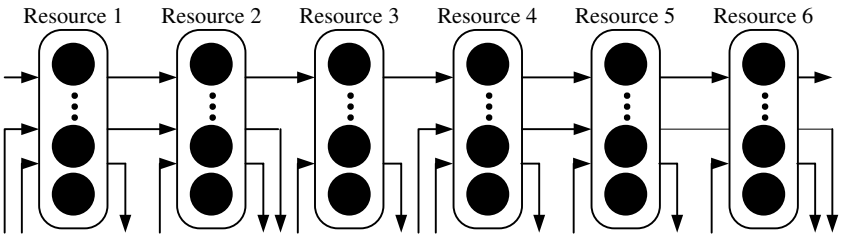

Figure 3.1: Multi-service loss network.

a brief survey of existing importance sampling methods for highly available services will be introduced. Then, we propose two importance sampling methods that can be used to estimate the blocking probability of the multi-service loss model. The generalized Static Importance Sampling using Standard Clock (S-ISSC) method is first introduced and proven to have BRE property, even under a multi-class service model with heterogeneous demands. Then the Adaptive Importance Sampling using Standard Clock (A-ISSC) method, which is based on the approximation of the "optimal biasing", is introduced.

3.1 Multi-service loss network modeling

Among the services introduced in the first chapter, the traffic groomed optical network, the robust server, and many other similar services can be modeled as multi-service loss networks. This allows us to unify the notation and the features of the importance sampling technique that we will present in the following sections.

Figure 3.1 shows a loss network with six *types* of resources, each with possibly different capacities, and with several possible call demands, shown as arrowed lines. A call can require two or more types of resources that are not physically connected. An arriving call may require different numbers of units from different types of resources, depending on the *type* of that call. The resources in this model can represent sub-wavelength units in a traffic-groomed network, and redundant components in a robust server. Arrivals and departures of calls represent call arrivals and service completions in an optical network, and component failures and repair completions in a robust server. The duration of a call then represents the service time of an optical traffic, or time to repair a group of failed components of a robust server. A call blocking event in the loss network corresponds to a call blocking event

in an optical network, or to an event of the network server being in the breakdown mode. In both cases, this means the service is unavailable.

Assume there are $R \in \mathbf{N}$ types of calls and $L \in \mathbf{N}$ types of resources, each type $r, r = 1, 2, \ldots, R$ call has a demand of $d_{r,l} \in \mathbf{Z}^+$ units of type l, $l = 1, 2, \ldots, L$ resource when it arrives. An arriving call of type r will be accepted only when every resource type l in the *demand set,* $D_r \triangleq \{l : d_{r,l} > 0\}$, that the call needs has at least $d_{r,l}$ units available. If a call is not accepted as the time it arrives, it will be blocked. A blocked call will simply be discarded, and the system state will not change.

In general, we assume that resource type l in the loss network has a capacity of $W_l \in \mathbf{N}$ units. Type r calls arrive at rate $\lambda_r \in \mathbf{R}$, and all calls have a mean call duration of $\frac{1}{\mu}, \mu \in \mathbf{R}$. In this dissertation, we assume that both inter-arrival times and call durations are exponentially distributed. This could be relaxed in the future. As noted, a call of type r will be accepted only if there are at least $d_{r,l}$ units of resource type l available, $\forall l \in D_r$. For an optical network, traffic demands for a type of call are usually the same for each type of resource (optical link), that is, $d_{r,l}$ are equal for all $l \in D_r$. This is referred to as homogeneous demand here in this dissertation. Moreover, for optical networks D_r always forms a "*route*" that is physically connected. Therefore, we may also categorize the calls according to the route and the class of a call.

In practice, the call blocking probability for each type of call is important in the case of optical networks, since a large unavailability probability can mean violation of the service contract (or "SLA", server level agreement) for a certain type of call. For robust servers, knowing the probability of a certain failure type being the last failure to cause system breakdown may also be important, since this relates to the possible cost of recovering from the breakdown. Moreover, if we know the probability of breakdown seen by every type of error, the overall service unavailability probability will be easy to calculate.

These different measures of performance can all be formulated in the loss network model, as long as we pay additional attention to each individual system. Although not all conceivable reliability situations can be modeled using such a loss network, this generalized model can indeed be used in several common situations where failure happens when one or more resources are exhausted.

3.2 Existing IS methods for highly available services modeled as loss systems

Methods for accelerating the simulation of packet-based networks of queues have been known for a while (see [28, 53, 17] and references therein). For reliable systems that can be modeled as Markov chains, there have been several publications regarding fast performance evaluation of highly . See [44] for a survey of such methods. The general idea of these methods is to change the "weights" of every possible path to failure state. Several of them are proven to have the BRE property. However, as mentioned in the introduction, when the system grows large, the number of states of the Markov chain can be very large, and similarly when multiple component failures are allowed. Since this complicates the transition of the Markov chain, such methods may become inefficient. By using loss networks, one has only to consider ways to "fill" the resources, rather than to account for every path to failure states, and therefore loss networks could offer a preferable model for such cases.

Consider a multi-service loss network as stated in section 2.3. In the demand set of a type of call, if call blocking events happen for a certain type of resource, say l', far more frequently than in all of the other types of resources, we can estimate the call blocking probability by changing the distribution under importance sampling (also called *biasing* the distribution) towards overwhelming resource type l'. This can be called a *single target system* and is discussed in [49] and [40]. However, modern design of highly available services would prevent the existence of this kind of bottleneck and would, instead, try to balance the load among all resource types. Under such a model, biasing the system toward a certain resource type would likely cause an *overbiasing* problem [53] and, thus, underestimate the blocking probability. In [37] and [38], the authors provide inverse convolution (IC) and approximation methods to distribute blocking among links, but these depend on the product form solution and may have to pre-calculate the entries of large tables before simulation begins.

3.3 Static importance sampling using standard clock method (S-ISSC)

In this section, we introduce an importance sampling simulation technique called static IS using standard clock (ISSC). The original static ISSC method is introduced in [58] to estimate the call blocking probability in a cellular telephone network with dynamic channel assignment. In [4], a dynamic method is used to estimate the call blocking probability in wavelength continuous WDM networks with single wavelength demand.

3.3.1 Asymptotic efficient biasing for a single queue

The idea of static ISSC comes from the asymptotically efficient exponential IS biasing methods in single server queues. In an M/M/1 queue, this method is simply exchange of arrival rate and service rates. As a result, in static ISSC we would like to extend this method to exchange of arrival and service rates of calls that have relationship with the call that we are interested in.

3.3.2 S-ISSC for multi-service loss networks

In this section, we generalize the formulation of the static ISSC method from [58] to include multi-class systems and prove that the method still exhibits a BRE in this more general setting, allowing us to apply the method to the large class of loss systems outlined in the previous sections.

Using a similar but generalized notation as in [4], we define S_l as the union of all call types that use the resource type l, and C_r, which is called cluster r, to be the union of all call types that use any resource call type r also uses. That is,

$$S_l = \{\text{call type } i : l \in D_i\},$$
$$C_r = \{\text{call type } i : D_r \cap D_i \neq \emptyset\}.$$

If we define an *event* to be either an arrival or a departure of a call, we can use N_r to represent the number of active type r calls in the system. We can also represent the system state by $N = \{N_1, N_2, \ldots, N_R\}$. Moreover, *valid states* can be written as $\{N$ s.t.

$\sum_r (N_r \cdot d_{r,l}) \leq W_l$, $\forall l = 1, \ldots, L\}$, where, as defined in Section 2, $d_{r,l}$ represents units of resource type l a type r call demands, and W_l stands for the capacity of resource type l. In other words, a system state N is said to be valid if every type of resource does not exceed its capacity. Finally, for type r calls, the system reaches a *blocking state* if any resource type l has less than $d_{r,l}$ units unused. Since a blocked call will be discarded and will not change the system state, a system starting from any valid state will remain in the set of valid states.

Assume inter-arrival times and service times are exponentially distributed. Under the *Standard Clock* simulation approach [57], consider applying Importance Sampling. We would like to change the arrival rates and service rates of some calls, so the blocking probability for the call type r, in which we are interested, becomes a non-rare event. The *event rate* for the system at state N before applying IS is

$$\Lambda_N = \sum_{i=1}^{R} \lambda_i + \mu \sum_{i=1}^{R} N_i.$$

Consider the change of measure that swaps the aggregate arrival rate of calls in cluster r, $\lambda = \sum_{i \in C_r} \lambda_i$, with service rate μ [58]. That is, the new inter-arrival times of call type i, $i \in C_r$, are exponentially distributed with rate $\lambda_i^* = \frac{\lambda_i}{\lambda}\mu$. As a result, the total arrival rate of calls in C_r becomes $\lambda^* = \mu$. Service times for these calls are now exponentially distributed with rate $\mu^* = \lambda$. Inter-arrival and service times outside the cluster have the original exponential distributions. The new event rate at state N becomes

$$\begin{aligned}\Lambda_N^* = \mu + \sum_{i \notin C_r} \lambda_i + \lambda \sum_{i \in C_r} N_i \\ + \mu \sum_{i \notin C_r} N_i.\end{aligned}$$

Now consider a system with IS "turning ON" at event "0", which is the event immediately before the "first event" in the following discussion. Using $N(j) = \{N_1(j), N_2(j), \ldots, N_R(j)\}$ to denote the system state just after the j^{th} event, the corresponding Radon-Nikodym derivative at the m^{th} event is

$$\begin{aligned}L(m) = \prod_{j=0}^{m-1} \frac{\Lambda_{N(j)} e^{-\Lambda_{N(j)} T_{j+1}}}{\Lambda_{N(j)}^* e^{-\Lambda_{N(j)}^* T_{j+1}}} \\ \times \prod_{j=0}^{m-1} \frac{\Lambda_{N(j)}^*}{\Lambda_{N(j)}} H(j+1),\end{aligned}$$

where T_j is the time between the $(j-1)^{\text{th}}$ event and the j^{th} event, and

$$H(j) = \begin{cases} \dfrac{\lambda}{\mu}, & j^{\text{th}} \text{ event } = \text{ arrival of call type } i, i \in C_r \\ \dfrac{\mu}{\lambda}, & j^{\text{th}} \text{ event } = \text{ departure of call type } i, i \in C_r \\ 1, & \text{otherwise.} \end{cases}$$

Let a be the total number of arrivals to cluster r in m events, and d be the total number of departures to cluster r in m events. We can rewrite $L(m)$ as

$$L(m) = e^{-\sum_{j=0}^{m-1}(\Lambda_{N(j)} - \Lambda^*_{N(j)})T_{j+1}} \left(\frac{\lambda}{\mu}\right)^a \left(\frac{\mu}{\lambda}\right)^d.$$

Moreover,

$$\begin{aligned} \Lambda^*_{N(j)} - \Lambda_{N(j)} =& \mu + \sum_{i \notin C_r} \lambda_i + \lambda \sum_{i \in C_r} N_i(j) \\ &+ \mu \sum_{i \notin C_r} N_i(j) - \left(\sum_{i=1}^{R} \lambda_i + \sum_{i=1}^{R} N_i(j)\mu \right) \\ =& -(\mu - \lambda)(\sum_{i \in C_r} N_i(j) - 1). \end{aligned}$$

Therefore,

$$L(m) = e^{-(\mu-\lambda)\left[\sum_{j=0}^{m-1}(\sum_{i \in C_r} N_i(j) - 1)T_{j+1}\right]} \left(\frac{\lambda}{\mu}\right)^{a-d}. \tag{3.1}$$

Finally, assuming that B_r is the event that the system is in any blocking state for call type r in steady state, and M is the random number of events just after the system first enters a blocking state while using IS, then we have

$$\begin{aligned} P(B_r) &= E[1(B_r)] \\ &= E^*[L(M) \cdot 1(B_r)] \end{aligned}$$

where $E^*[\cdot]$ means the expectation with respect to the IS biasing distributions.

For the simulation, we use the quasi-regenerative (or A-cycle) technique introduced in the precious chapter. If an A-cycle is started from the stationary distribution conditioned on the process just entering A, the blocking probability can be expressed as

$$\begin{aligned} P(B_r) &= \frac{E[T_B]}{E[T_A]} \\ &= \frac{E[T_B|A_B]P(A_B)}{E[T_A]} \end{aligned}$$

where T_A, T_B and A_B are the A-cycle length, time spent in blocking states of call type r in an A-cycle, and the event that an A-cycle contains any blocking state or call type r, respectively.

These quantities (A, T_A, T_B, and A_B) should all be related to r, the type of the call we are interested in, but here we omit the r for simplicity. We can estimate $P(A_B)$ by "turning ON" IS when the system state reaches A from A', and until it reaches a blocking state. Then we "turn off" IS and perform ordinary simulation from this point until the system leaves the set A, and $E[T_B|A_B]$ can be estimated. Moreover, since it is not rare for the system to be in an A cycle (A-cycles typically have a significant, non-infinitesimal length), we can simply use ordinary simulation to estimate $E[T_A]$ without IS.

As mentioned above, an A-cycle should be started from the stationary distribution conditioned on the process just entering A. However, IS changes the underlying distribution and therefore the simulation run after an IS A-cycle cannot be re-used for another A-cycle. A "splitting" technique [28] can be used, in which an ordinary simulation is run in parallel to the IS to provide entry-points for IS A-cycles. Each IS simulation starts at one of these entry points with IS "turned on", then "turns off" IS when the system reaches a blocking state, and finally ends as the system leaves set A. Actually this ordinary simulation can be run in advance, and can also be used to estimate $E[T_A]$. For different call types that have different definitions of the set A, the same ordinary simulation can be re-used for all the entry points of these A-cycles. In this chapter, we define set A for call type r to be the set of all states in which the number of calls in cluster C_r is greater than zero.

3.3.3 BRE property of S-ISSC

In the following, we extend our result from [31], which follows a generalized procedure as in [58], to prove that under *lightly loaded* situation, the static ISSC method will have Bounded Relative Error (BRE).

Theorem 3 *Assume there are constants* $\epsilon, \varphi_i \in \mathbf{R}$ *such that* $\lambda_i = \varphi_i \epsilon$, $\forall i$, *and* $\mu > \lambda = \sum_{i \in C_r} \lambda_i$, *then the static ISSC estimator for* $P(A_B)$, *introduced in this Section, has the BRE property as* $\epsilon \longrightarrow 0$.

Proof Without loss of generality, assume that we are interested in estimating $P(A_B)$ of call type r. We would like to prove the BRE using Lemma 2. Consider a blocking state of type r calls that need the least number of active calls in cluster r. For each resource type $l \in D_r$, the demand set of call r, let

$$b_{l1} = \left\lfloor \frac{W_l}{\max_i d_{i,l}} \right\rfloor, \; r_1 \triangleq \arg\max_i d_{i,l}$$

$$b_{l2} = \left\lfloor \frac{W_l - b_{l1} \cdot d_{r_1,l}}{\max_{i,i \neq r_1} d_{i,l}} \right\rfloor, r_2 \triangleq \arg\max_{i,i \neq r_1} d_{i,l}$$

$$\vdots$$

$$\text{until min } i \text{ s.t. } \sum_i b_{li} \cdot d_{r_i,l} \geq W_l - d_{r,l}$$

$$b_l = \sum_i b_{li},$$

and

$$b = \min_{l \in S_r} b_l.$$

In other words, we *exhaust* resource type l by first using b_{l1} call arrivals of the type which needs the most units of resource type l, until no more such calls can be accepted. Then the second-most demanding type of calls are used to exhaust the resource type, and repeated until reaching a blocking state for type r calls. Minimizing over $l \in S_r$, we obtain the minimal number of active calls needed for a blocking state for type r calls.

From the definition above, we know that if the system reaches a blocking state for type r calls just after the m^{th} event happens, there are at least b active calls within cluster r. Moreover, from the definition of A, the active calls within cluster r will be at least one.

Since $\mu > \lambda$ and $\sum_{i \in C_r} N_i(j) \geq 1$ in set A, we have

$$(\mu - \lambda) \left[\sum_{j=0}^{m-1} \left(\sum_{i \in C_r} N_i(j) - 1 \right) T_{j+1} \right] \geq 0.$$

Furthermore, since $a - d \geq b$ and $\mu > \lambda$, we have

$$\left(\frac{\lambda}{\mu} \right)^{a-d} < \left(\frac{\lambda}{\mu} \right)^{b}.$$

From (3.1), we obtain

$$L(m)\mathbf{1}_{\{A_B\}} \leq \left(\frac{\lambda}{\mu}\right)^b = \alpha\epsilon^\gamma,$$

where

$$\alpha = \left(\sum_{i\in C_r} \varphi_i/\mu\right)^b,$$

and

$$\gamma = b.$$

Next, we need to show that $P(A_B) \geq \beta\epsilon^\gamma = \beta\epsilon^b$. Consider a sample path to reach the state of active calls mentioned above, by arrival events only. This will bring the system into a blocking state of call type r, and the probability of this sample path is a lower bound of $P(A_B)$.

Using $\lambda(j)$ to denote the arrival rate of the call type which arrives as the j_{th} event, we have

$$\begin{aligned} P(A_B) &\geq \prod_{j=0}^{b-1} \left(\frac{\lambda\,(j+1)}{\Lambda_{N(j)}}\right) \\ &\geq \left(\frac{\min_j \lambda(j+1)}{\max_j \Lambda_{N(j)}}\right)^b \\ &\geq \left(\frac{\min_j \varphi_j\epsilon}{b\mu + \sum_i \lambda_i}\right)^b \\ &= \beta\epsilon^b \end{aligned}$$

where $\beta = \left(\frac{\min_j \varphi_j}{b\mu+\sum_i \lambda_i}\right)^b$.

From all of the above and Lemma 3.2, it follows that static ISSC has bounded relative error (BRE).

3.4 Adaptive-ISSC (A-ISSC)

The assumption of a lightly loaded system in last section may not always be practical in real situations. In this section, we would like to consider, instead, the cases in which the arrival rates do *not* go to zero. Service unavailability probabilities may still be rare events due to *high capacity* or fast service. Using the idea of adaptive biasing from [27], combined

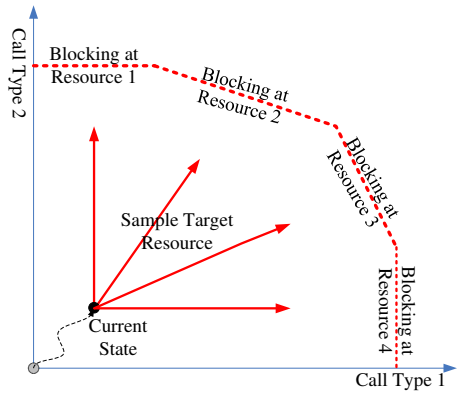

Figure 3.2: Adaptive Importance Sampling.

together with ISSC and the A-cycle method described in the previous section, we introduce Adaptive ISSC (A-ISSC) in this section.

Assume that we want to estimate the blocking probability in a multi-target system, where the blocking could happen due to the lack of more than one type of resource. The idea of adaptive biasing is introduced in [26] for multi-class, homogeneous demand networks, where an arriving call requires the same number of units among the resource types in its demand set. In [27], this method is applied to non-homogeneous demands, and calls with different priorities.

In this section, we use the basic idea from [27] but introduce a simpler resource sampling approach and a different biasing method, and apply them to the general loss network model for highly available services, with multiple classes and non-homogeneous demands.

Our method works as follows: First pick a type of resource from a probability distribution that reflects the relative *importance* of each resource type that could lead to a blocking state. Then the system is biased using importance sampling, as if the sampled resource type is the only target. After an *event*, another sampling of resource types is done, and the IS biasing distribution is changed again to bias the system toward the newly selected type of resource. These steps are repeated until the system reaches a blocking state for the call type in which we are interested. The idea is shown in Figure 3.2, where a simplified model of four types of resources and two types of calls are shown.

Assume that we want to sample a resource type for the next target from a probability distribution $\widehat{f}_N(l)$, where $l = 1, \ldots, L$ are the resource types, and N is the current system

state. It is known (see [28]) that if we want to use IS to estimate the probability that a random variable X, with pdf $f(x)$, is in a set B, the *optimal* IS distribution $f^*(x)$ is:

$$f^*(x) = \begin{cases} \frac{f(x)}{\int_{x \in B} f(x)dx} & x \in B \\ 0 & \text{otherwise.} \end{cases}$$

That is, sample only the elements in B and also preserve the relative probabilities of elements in B. It is obvious that this distribution requires the understanding of the probability we want to estimate and, thus, is not feasible. However, biasing the probability distribution so that the relative probabilities of elements in the importance set can be preserved does help one prevent over-biasing problems and reduce the estimator variance [28]. In a dynamic case like our loss network, it is usual that even the relative probabilities cannot be easily estimated. As a result, we would like $\widehat{f}_N(l)$ to be a distribution that approximately preserves the relative probabilities the system will be in the blocking states of call type r, due to the lack of resource type l, starting from state N. We can estimate $\widehat{f}_N(l)$ by calculating how easily the blocking state of resource type l can be achieved from the current state N.

In [25], methods are suggested for calculating $\widehat{f}_N(l)$, which are based on a weight for each l that is a product of *contribution* and *likelihood*. In [26], a more complex algorithm is suggested. In [12], the author suggest using cross entropy as the guide to do the sampling. Here we propose using the probability of the *shortest path* from state N to a blocking state that is due to the lack of resource type l as the weight of type l. $\widehat{f}_N(l)$ is a probability distribution that is proportional to the weight of resource type l. Similar ideas using IS based on shortest paths can also be found in simulating reliability systems (see [3] and [10])

To calculate the weight, consider the most likely path to reach a blocking state for call type r from current state N. For a system with arrival and service rates independent of system states, such as the model we use in this chapter, the shortest path would correspond to successive arrivals of call of traffic type i, whose $d_{i,l}\frac{\lambda_i}{\mu}$ is the largest among all call types that use class l, until the system reaches the blocking state of call type r. Since $d_{i,l}\frac{\lambda_i}{\mu}$ is independent of system states, this can be calculated in advance; and in each step, we only need to calculate $\|D_r\|$ weights, where $\|D_r\|$ is the number of resource types in D_r, one for each type of resource used by the call type in which we are interested. Our simulation result shows that using this method, $\widehat{f}_N(l)$ can be calculated efficiently (as in the following algorithm), while the estimation of blocking probability is still accurate.

The resource sampling method in A-ISSC could also be used even if the arrival and/or

the call duration rates are *dependent* on the system state. For such cases, we have to consider all possible sample paths, which may require an excessive amount of computing time. One possibility is to choose the *largest* $d_{i,l}\frac{\lambda_i(N)}{\mu_i(N)}$ for each step and use this resulting path as the shortest path, where $\lambda_i(N)$ and $\mu_i(N)$ are arrival and service rates for call type i in system state N. This method requires $O(\|D_r\| \cdot N(c))$ computations, where $N(c)$ is the mean value of the number of call types that use resource type l over the $\|D_r\|$ resource types that the call type of interest, r, is using.

After the target resource type $\widehat{l}$ is decided, we can then bias the system toward that type of resource. Exponential biasing, which is often suggested in single target systems, can be used. For exponentially distributed inter-arrival and service times, a interchange of the aggregate arrival rate of the call types that use resource type $\widehat{l}$ with the aggregate service rate of the active calls that use resource type $\widehat{l}$, may also be used (a biasing which is asymptotically optimal in single-class systems).

In a system with very large capacity, a modification of the dynamic method provided in [58] could be applied for biasing the system toward resource type $\widehat{l}$. Let the number of active calls that uses resource type $\widehat{l}$ be $\widetilde{N}(\widehat{l})$, and consider the following change of measure:

$$\lambda^*(\widetilde{N}(\widehat{l})) = \widehat{\lambda} + \widetilde{N}(\widehat{l})(\mu - \mu^*(\widetilde{N}(\widehat{l})))$$
$$\mu^*(\widetilde{N}(\widehat{l}) + 1) = \frac{\widehat{\lambda}\mu}{\lambda^*(\widetilde{N}(\widehat{l}))},$$

where $\widehat{\lambda} = \sum_{i \in S_l} \lambda$, and $\mu^*(0) = 0$. This method is proved to be optimal in a single resource type, single call type case, and to have the BRE property in a single target case with multiple classes of call demands.

Our Adaptive ISSC algorithm to estimate the blocking probability for call type r, starting from an entry point of an A-cycle, which has the same definition as in the previous section, works as follows:

1. For each resource type, l, $l \in D_r$, find the shortest path from the current state N to the blocking state of type r call due to the lack of the resource type l, and calculate the probability of this path. Call this probability $w_N(l)$. Note that $w_N(l) = 0$ for all $l \notin D_r$.

2. Sample a target resource type $\widehat{l}$ from the distribution that $\widehat{f}_N(l) = \frac{w_N(l)}{\sum_{k \in D_r} w_N(k)}$.

3. Bias the system in favor of $\widehat{l}$, then sum up the arrival and departure rates, and sample the next event and determine its attributes.

4. Calculate the Radon-Nikodym derivative for this new event (assume this new event to be the m^{th} event) as

$$\begin{aligned}\widehat{L}(m) &= \frac{\Lambda_{N(m-1)} e^{-\Lambda_{N(m-1)} T_m}}{\Lambda^*_{N(m-1)} e^{-\Lambda^*_{N(m-1)} T_m}} \times \frac{\Lambda^*_{N(m-1)}}{\Lambda_{N(m)}} (H(m)) \\ &= e^{(\Lambda_{N(m-1)} - \Lambda^*_{N(m-1)}) T_m} H(m),\end{aligned}$$

   where $N(k)$ is the system state just after the k^{th} event, T_m denotes the time from the $(m-1)^{\text{th}}$ event to the m^{th} event, and $H(m)$ differs according to the biasing method used. For example, if we switch the aggregate arrival rate to $\widehat{l}$ with the aggregate inverse average call duration at $\widehat{l}$,

$$H(m) = \begin{cases} \frac{\lambda}{\mu}, & \text{the } m^{\text{th}} \text{ event is an arrival to } \widehat{l} \\ \frac{\mu}{\lambda}, & \text{the } m^{\text{th}} \text{ event is a departure from } \widehat{l} \\ 1, & \text{o.w.} \end{cases}$$

5. Repeat steps 1-4 until the system reaches any of the blocking states. The overall Radon-Nikodym derivative of this sample path is the product of the derivatives of all steps.

As in the previous section, the IS biasing is then "turned off" and simulation continues to estimate $E[T_B|A_B]$, until the system leaves the A-cycle.

3.5 Simulation model and results

3.5.1 Simulation Models

In this section, we introduce two highly available service examples that will be simulated. One is a robust network server with small component failure probability, and the other is a traffic groomed optical network that has large sub-wavelength capacities. For the first system, we will be comparing performance of the two ISSC methods against ordinary

Monte-Carlo simulation and the adaptive biasing method introduced in [27]. For the second scenario, we will be comparing the performance of A-ISSC against the adaptive biasing method, and also the dynamic ISSC method introduced in [58].

Robust network server

Consider a robust server as in Figure 1.3. Assume there are four processors, three sets of disk arrays with ten disks in each array, a five-interface network controller, and three duplicated power sources for this server. Components can fail one at a time, or *simultaneously* with the same or different kinds of other components. Errors happen according to Poisson distributions with different arrival rates. Table 3.1 shows the twelve types of errors considered here, and table 3.2 shows the arrival rates of these types of errors and the numbers of each kind of server components they correspond to.

Repair times for different fail types are exponentially distributed with rate $\mu = 130$. Two values of λ, 1 and 0.5, are used in this simulation. A group repair policy, as stated in section 2, is applied. If all components of the same kind fail, the system is considered in breakdown and unavailable to service requests.

To obtain a comprehensive comparison, the Static ISSC, Adaptive ISSC, traditional Monte Carlo, and adaptive biasing (denoted as AB) methods are used to simulate the system breakdown probability of this server. For ISSC methods, A-cycles are defined to be the time between the beginning of two consecutive busy periods of cluster r, as stated in the previous sections. An ordinary simulation is performed for the entry point and average A-cycle length for the A-cycle method for each type of error. The first $100,000$ entry points of the ordinary simulation are used for ISSC methods. The Jackknife method is used to estimate the relative error of the estimators.

Traffic groomed optical network

Consider a seven-ADM tandem optical network as in Figure 1.1. Each link between two adjacent ADMs is modeled as a single wavelength that is groomed to carry 1024 sub-wavelength units. In this case, a call will have same demand for sub-wavelength units throughout all the links it uses, and different *classes* of calls need different units along their *route*. Therefore we can further categorize the call types into routes u and classes k.

Table 3.1: Error types in robust server simulation.

Type	Processor	Disk 1	Disk 2	Disk 3	Net port	Power
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	0	0	0	0	0	1
7	1	3	3	3	1	1
8	0	2	2	3	2	1
9	0	2	4	3	0	0
10	1	0	0	0	2	0
11	1	4	2	3	0	1
12	1	0	0	0	1	1

Table 3.2: Error arrival rates in robust server simulation

Error type	1	2	3	4	5	6	7	8	9	10	11	12
Arrival rate	4λ	8λ	8λ	8λ	4λ	2λ	λ	λ	λ	λ	λ	λ

Table 3.3: Call arrival rates in the simulated network.

Route \ Class	1	2	3	4
1	5000	4950	4900	4850
2	4800	4750	4700	4650
3	4600	4550	4500	4450
4	4400	4350	4300	4250
5	4200	4150	4100	4050
6	4000	3950	3900	3850
7-11	2500	2500	2500	2500
12-15	2000	2000	2000	2000
16-18	1500	1500	1500	1500
19-20	1000	1000	1000	1000
21	500	500	500	500

In the simulation model, all 21 possible routes are considered, that is , $u = 1 \dots 21$, and for each route assume there are 4 classes of traffic, that is, $k = 1 \dots 4$. The demands for sub-wavelength units of the four classes are $d_1 = 2$, $d_2 = 4$, $d_3 = 6$, and $d_4 = 8$.

Assume that the routes in Figure 1.1 are numbered from 1 to 21 from the top-left route to the very bottom one. For example, the route between the first and the second nodes to the left is route 1, the route between the first and the third nodes to the left is route 7, and the route that uses all of the links is route 21, and so on. The arrival rates (calls per second) are assigned as shown in Table 3.3.

The service rate for all calls is set to be 500 calls per second. Inter-arrival times and call duration for all call types are assumed to be exponentially distributed with the above rates.

We can see that this is indeed a *high capacity* scenario, since although arrival rates are larger than service rate, the blocking probabilities are still rare events due to the very high capacity. Therefore, Adaptive ISSC for high capacity is used to simulate the call blocking rate. Moreover, traditional Monte Carlo simulation, the adaptive biasing method, as well as the dynamic ISSC method introduced in [58] are also implemented for comparison. Finally, in this configuration, the numerical decomposition method introduced in [59] can be performed to estimate a theoretical blocking probability. For A-ISSC method, the same definition as in the robust server case is used for A-cycles for each route r. For each call type, an ordinary simulation is done for the entry points and average A-cycle length for

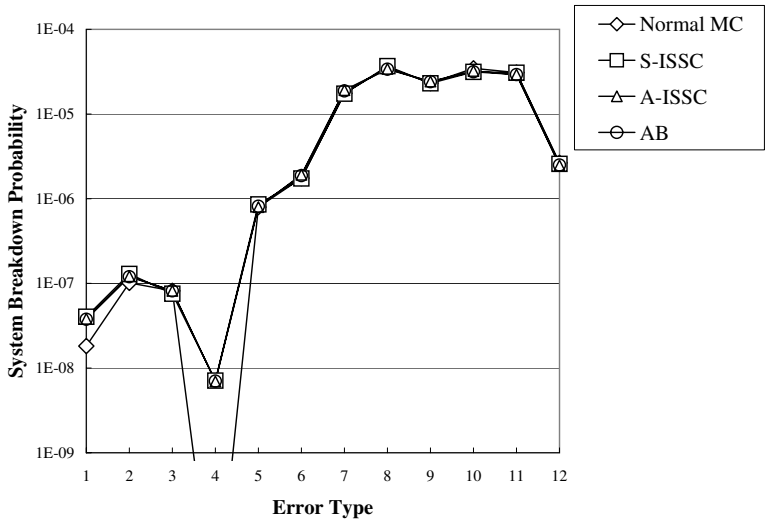

Figure 3.3: Server breakdown probability caused by different kinds of error, λ=0.5.

A-ISSC method. The first 100, 000 entry points of the ordinary simulation are used.

3.5.2 Simulation Results

Robust network server

Figure 3.3 and 3.4 show the server breakdown probability seen by each type of error under the two different service rates. From these figures, we can see that for $\lambda = 0.5$, the breakdown probability for error type 4 is too small for ordinary MC to estimate. In cases other than this, results from the all the simulation methods agree well with each other. Thus we can conclude that the methods we propose generate accurate and unbiased estimates.

For the performance of these methods, Figure 3.5 shows the relative error, which is the standard deviation divided by the mean value of the estimate, versus the breakdown probabilities, for all the breakdown probabilities estimated in this system, including both λ values. In this graph, we can see the relative error of the ordinary Monte-Carlo simulation explodes as the probabilities become rare. On the other hand, the relative errors for all the IS methods remain small and stable. The result shows that for the cases in which the blocking probabilities are rare due to small call arrival rates, all the IS methods can provide accurate estimation of the breakdown probabilities, although the A-ISSC method (and the

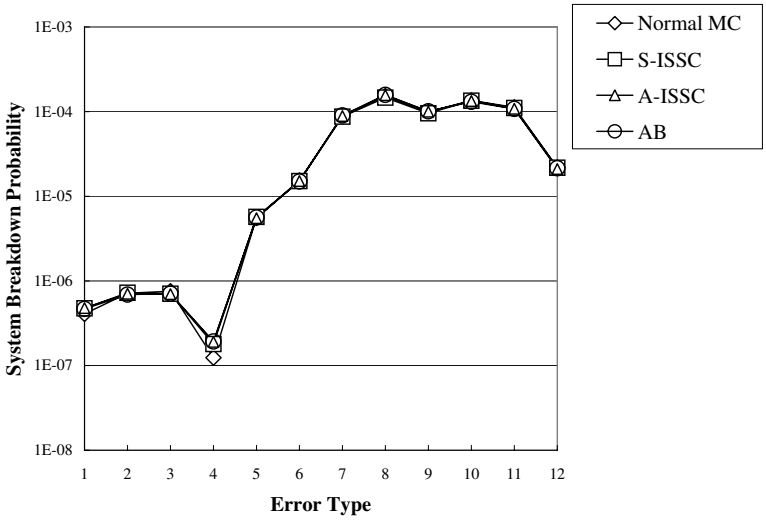

Figure 3.4: Server breakdown probability caused by different kinds of error, λ=1.

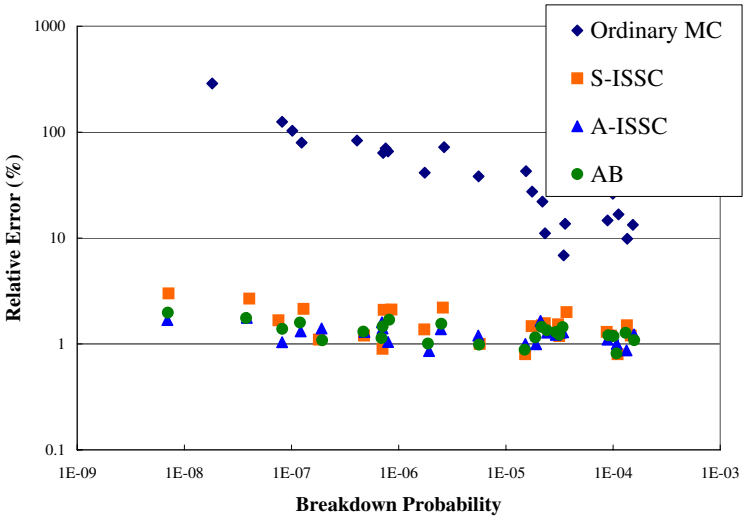

Figure 3.5: Relative error versus breakdown probability.

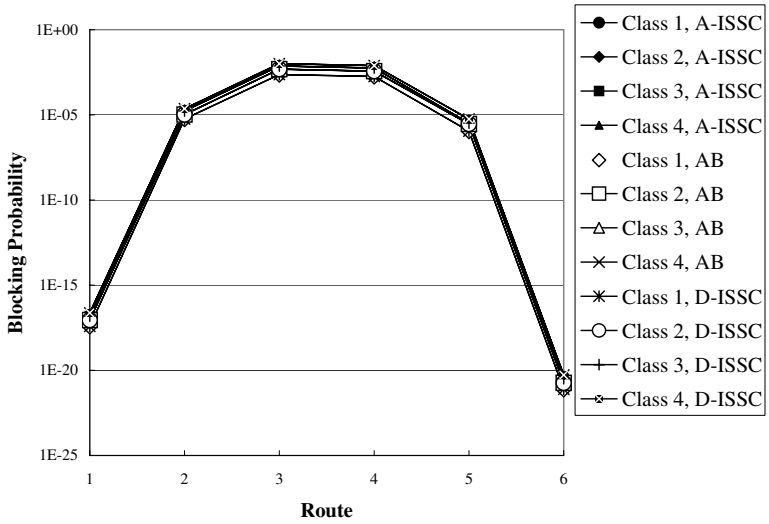

Figure 3.6: Blocking probabilities for routes using one link.

Table 3.4: Blocking probabilities and relative errors of Route 11, type 1.

Method	Blocking Prob	Relative Variance (%)
Ordinary MC	8.49E-07	161.6430008
A-ISSC	8.79E-07	2.101492862
D-ISSC	8.85E-07	9.322973778
AB	8.79E-07	2.247205765
APP2	8.78E-07	N/A

AB method) has not been shown to have the BRE property.

From the efficiency point of view, since all three IS methods provide accurate estimates, the method that requires the *least CPU time* to generate an event would be the most preferable. Static ISSC would be the one in this case, since it does not require any extra effort to calculate the arrival rates throughout the simulation.

Traffic groomed optical network

Call blocking probabilities for calls from different routes and classes in the traffic groomed optical network are shown in Figure 3.8, Figure 3.9, Table 3.4, and Table 3.5. In these figures, results from A-ISSC, dynamic ISSC ([58], denoted as D-ISSC), and adaptive

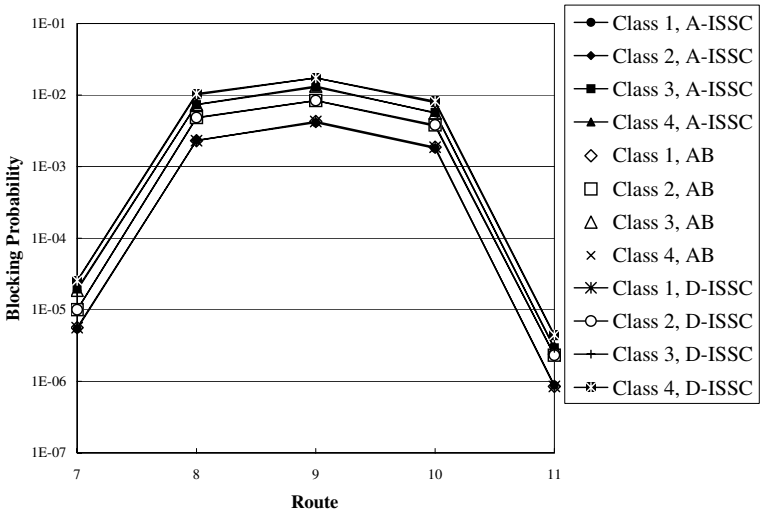

Figure 3.7: Blocking probabilities for routes using two links.

Table 3.5: Blocking probabilities and relative errors of Route 11, type 2.

Method	Blocking Prob	Relative Variance (%)
Ordinary MC	2.35E-06	117.2955219
A-ISSC	2.32E-06	1.750831756
D-ISSC	2.31E-06	8.386684392
AB	2.32E-06	1.863236585
APP2	2.32E-06	N/A

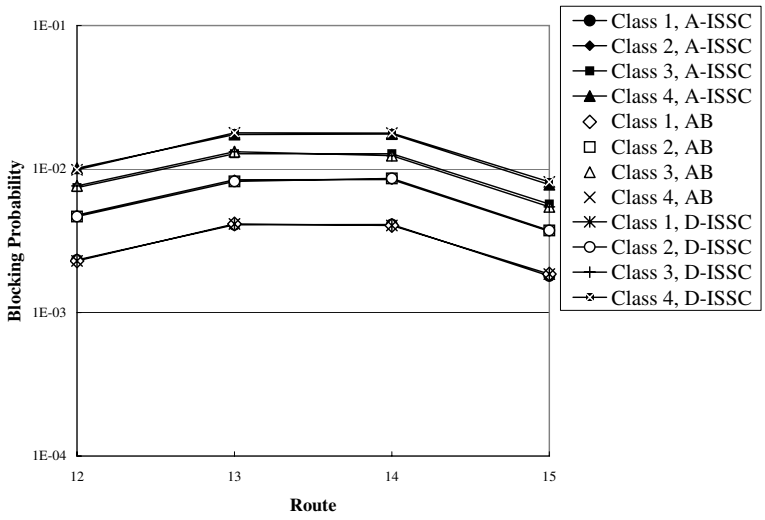

Figure 3.8: Blocking probabilities for routes using three links.

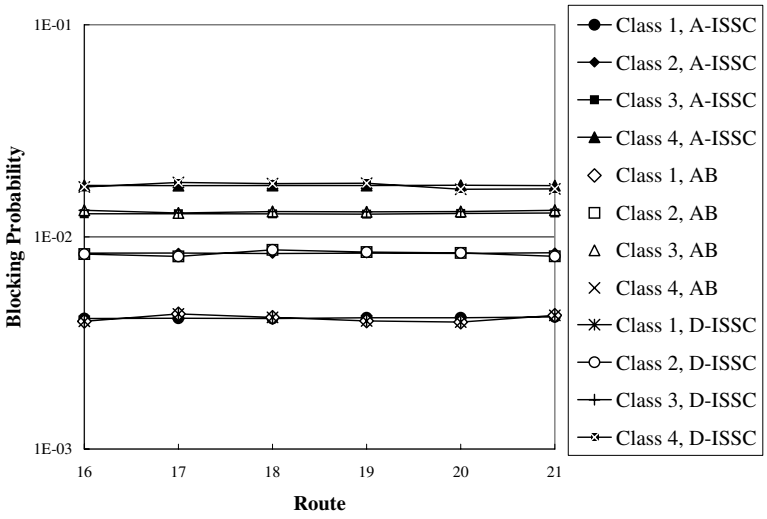

Figure 3.9: Blocking probabilities for routes using four to six links.

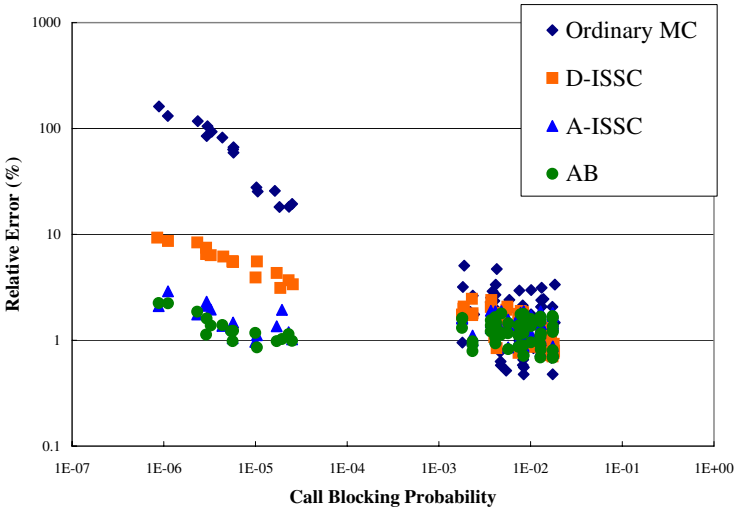

Figure 3.10: Relative error versus blocking probability.

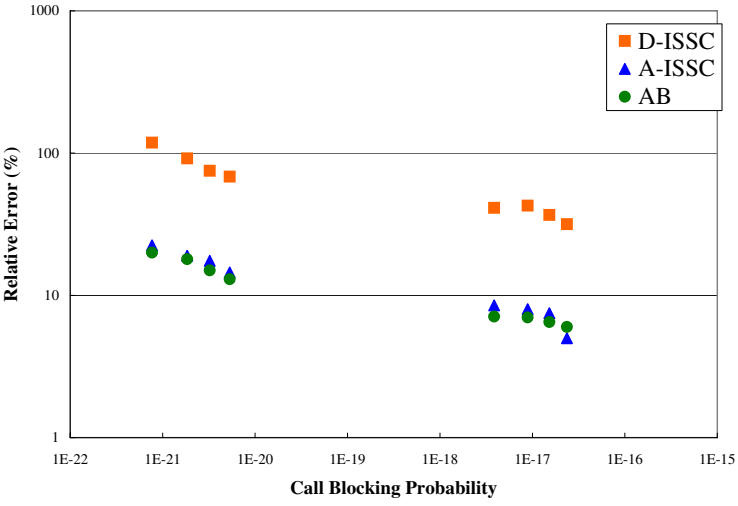

Figure 3.11: Relative error versus blocking probability.

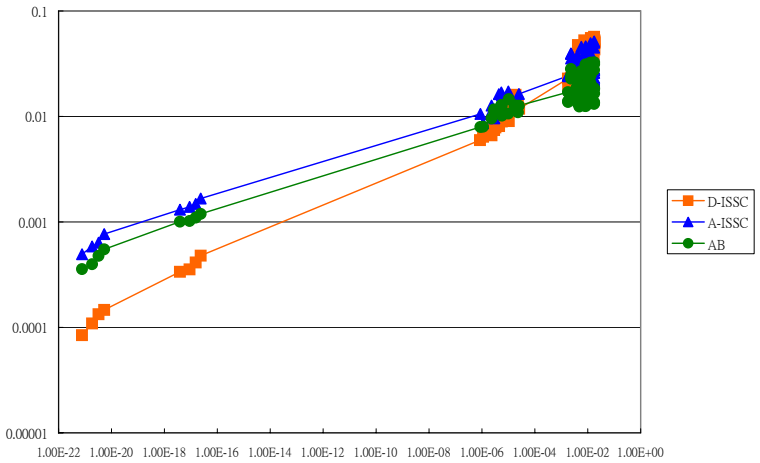

Figure 3.12: Efficiency versus blocking probability.

biasing ([27], denoted as AB) are illustrated. We can see that all the methods produce consistent simulation results, while the call blocking probabilities lower than 10^{-8} are too small for the ordinary MC method. The tables show the estimated blocking probabilities as well as relative errors for two call types of route 11 from the ordinary MC method, the IS methods, and also the numerical decomposition estimate (denoted as APP2) from [59]. From the Tables we can see again that the estimates are all consistent, and A-ISSC has the minimal relative error for both cases.

To compare the relative accuracy of the IS methods, Figures 3.10 and 3.11 show the relative error versus the call blocking probabilities. In Figure 3.10, the relative error for the ordinary Monte Carlo simulation still increases rapidly when the blocking probability becomes very low. The relative error for the D-ISSC method also grows, but with a much slower rate. For A-ISSC and AB, the relative errors rise even slower. Figure 3.10 shows the relative error of A-ISSC, D-ISSC, and AB versus very low call blocking probability values. For such low values, the ordinary MC method cannot observe any blocking event for the simulation duration. From the figure we can see the relative errors for all the methods grow, as the blocking probability goes down, since neither of the methods are shown to have a BRE property for such an environment. However, we can see the relative error behaviors of A-ISSC and AB are similar, while they maintain smaller than that of D-ISSC.

Regarding the efficiency of the methods, we can define the index of efficiency to be

$1/((RE)^2 * T)$, where RE is the relative error of the estimator, and T is the CPU time to execute the simulation. Figure 3.12 shows the efficiency index versus the call blocking probabilities for our simulation runs. From the figure, we can see that when the blocking probability is of the order of 10^{-2}, D-ISSC is actually the most efficient, due to the lower CPU time it requires to finish the simulation. However, as the probability becomes smaller, the A-ISSC and the AS methods become more efficient. Since the A-ISSC and AB methods have similar relative error behaviors, the shorter CPU time in A-ISSC does provide an advantage in overall efficiency.

3.6 Conclusions

In this chapter, we have described two importance sampling methods to evaluate the probability of unavailability in highly available services that can be modeled as multi-service loss networks. In the lightly loaded cases, we have proved that static ISSC has a bounded relative error (BRE) property when applied to loss networks, even if different types of calls may require different amounts of units in different resources. Second, for cases other than that of light load, we argued that Adaptive ISSC may instead provide an efficient and robust way to set biasing parameters. We have also performed simulation comparisons between our approach and results obtained by applying directly the methods in [58] and in [27]. In simulation results, using A-ISSC provides very accurate estimates of the call blocking probability that lead to low relative variance and high efficiency.

The methods introduced in this chapter can be extended for many other kinds of highly available services, such as highly dependable software [23] and next generation wireless networks, or more generalized, "mesh" networks, without much additional effort. Moreover, unlike the methods that depend on product form solutions, adaptive ISSC may be used in multiple target system models in which there exist optimal or good biasing methods in single target cases. In such cases, we may extend asymptotically optimal bias methods of single target systems to multi-target systems by using Adaptive ISSC.

Chapter 4

Stochastically Optimized Importance Sampling Method

The static and adaptive ISSC methods introduced in the last chapter enable the efficient evaluation of highly available services that can be modeled by multi-service loss networks. However, there are still many highly available services that can not be modeled as multi-service loss network models. An example is the Optical Burst Switching Networks introduced in Chapter 1. In an OBS network, all the resources (wavelengths) a call needed are not simultaneously possessed at the arrival time of that call. Instead, wavelengths on link are reserved only when part of the burst is currently using the link. To model a behavior like this, more complex models have to be used. To use importance sampling methods such models, even the approximation of the "optimal biasing" could be difficult.

In this chapter, we suggest using stochastic optimization methods to directly minimize the variance of an IS estimator. We choose to use Simulated Annealing, a commonly used stochastic optimization method that is famous for its ability to avoid being trapped in local minimal. The resulted IS method, SA-ISSC, is easy to use and can efficiently find good biasing parameters for the IS estimation. For the following sections, we first introduce the performance models used to describe OBS networks, then the SA-ISSC model will be proposed. Finally the simulation results validate that the SA-ISSC can reach an optimal solution in a single OBS node, and converges to a solution with minimal variance in a more

general OBS network.

4.1 Performance Model for Optical Burst Switching Networks

In many publications [39, 33], an OBS node is modeled as an M/G/C/C loss-type queue. In an M/G/C/C queue, customers arrive according to a Poisson process, and the service time is independent and identically distributed, but not necessarily exponential. The blocking probability of a call can be described using the well-known Erlang-B formula,

$$P_b = \frac{\rho^c/c!}{\sum_{i=0}^{c} \rho^i/i!},$$

where ρ stands for the offered load to the system. Moreover, the authors of [41] showed that the distribution of burst lengths would not affect the burst loss probability in an OBS node and therefore many papers also assume that the length of a burst is also distributed according to an exponential distribution, which leads to an M/M/C/C model.

Because the Erlang-B formula is simple, M/M/C/C models are widely used as a basis for more complicated OBS networks, such as converter limits, etc. However, M/M/C/C models (and also generalized M/G/C/C models) are generally used to model a *single* OBS node, while they do not describe the characteristics of a whole OBS network well. For example, in a real OBS network a burst may span more than one link if the nodes are not far from each other and the burst is long. The M/M/C/C method would not be able to model such a behavior, while a generalized Markov chain model could have too many states to be evaluated efficiently analytically.

The authors of [7] suggest using a Markov chain model for a specific routing path, using the number of bursts at a node currently as the state of the node. However, instead of calculating the equilibrium probability distribution of the whole chain directly, they suggest using decomposition algorithms to approximate the equilibrium probability distribution of the chain. The idea of decomposition is shown in figure 4.1. The system is decomposed to two sub-systems, and numerical methods are used to approximate the equilibrium probability in each sub systems, using information from adjacent sub-systems. In [7], the control messages are not modeled at all, that is, the bursts themselves compete for wavelengths.

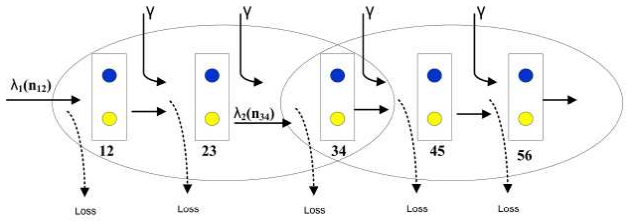

Figure 4.1: Decomposition of an OBS network into two sub systems.

This is indeed a model for the JIT method if we consider the arrival of first bit of the control message and the departure of last bit of real burst as the duration of a burst in this model. Moreover, it can also be used to approximate the behavior of a JET method with each call assigning similar lengths of offset time.

The decomposition method works very fast, but one problem of the approximation is the error introduced in the decomposition. As a result, simulation methods may be a preferable answer to estimating the burst blocking probability. The authors in [51] use OPNET to simulate an OBS network with four user nodes and two OBS nodes, using JIT reservation method. In their simulation model, the authors recreate the actual node behavior, including packet generation using OPNET built-in packet generation modules, burst aggregation, and control message and wavelength reserving, using the finite state machines provided in the OPNET software. Moreover, the users in [46] simulate a 16-node NSFNET topology. The authors also record the time to run different lengths of simulation as an indicator of the efficiency of such simulation methods. Although the simulation model is relatively easier to implement, the required CPU time to run the model may be considered unacceptably high, as the authors mention in their concluding notes. For real-sized OBS networks, efficient simulation techniques are still definitely needed.

For our IS based simulation method, we will apply the Markov model introduced in [7], which models an OBS network with a burst that can span two consecutive links. As the authors in [7] mention, this is one of the few models that can be used to describe the behavior of a whole OBS network. The model is shown in figure 4.2.

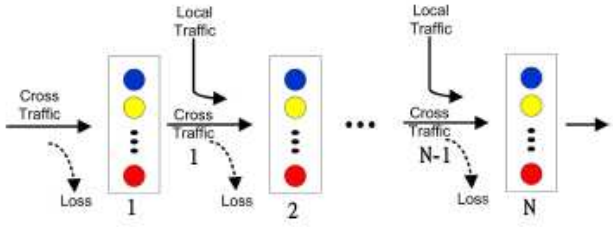

Figure 4.2: OBS model used in this chapter.

4.2 Simulated Annealing Optimized Importance Sampling Method

As we can see in the previous section, although the burst dropping probability can be calculated for a simplified single OBS node very quickly using Markovian models, for a practical size of network it would be impossible due to the explosion of states. Simulation, on the other hand, can be used to estimate the burst blocking probability of the entire network, provided a good model can be created. However, to obtain a satisfactory confidence interval, the time needed for simulation grows fast as the burst loss probability decreases. This prevents the simulation from estimating the blocking probability of real sized networks, in which the blocking probability can be less than 10^{-6}, much lower than the typical 10^{-1} order in all the papers using the simulation approach in the literature.

As a result, we believe that it would be beneficial to apply fast simulation techniques, such as importance sampling, in the simulation of such an OBS model. However, for this model, or even more complex model in which an OBS burst can be span more than two links, a good IS "bias" would be hard to find from the analysis of the model. Even the approximation of the theoretical optimal biasing would be hard. Therefore, we propose using an optimization method to directly optimize the variance of the estimator. Since the variance itself is also a stochastic variable, optimization methods such as simulated annealing, which have the ability to prevent being trapped in a local minimal, would be beneficial to use.

4.2.1 Simulated Annealing

Simulated annealing [1] is a popular stochastic optimization method that can prevent the objective value from being trapped in local minima. This is achieved by accepting worse solutions with a probability that is a function of both current and candidate values, as well as a parameter called "temperature". Assume we have a function $g(\mathbf{x})$ for which we would like to find a value $\mathbf{x}_{opt}$ that minimizes $g(\mathbf{x})$. Simulated annealing works as follows. Consider the current state $\mathbf{x}_i$ and a candidate state $\mathbf{x}_j$, which is usually uniformly sampled in a neighborhood of state $\mathbf{x}_i$. If $g(\mathbf{x}_j) \leq g(\mathbf{x}_i)$, then $\mathbf{x}_j$ will be accepted as the next state. However, if $g(\mathbf{x}_j) > g(\mathbf{x}_i)$, $\mathbf{x}_j$ could be still accepted with a probability $e^{(g(\mathbf{x}_i)-g(\mathbf{x}_j))/c}$, where c is the current temperature. It is shown that starting at a high temperature, and if we run the system long enough and decrease the temperature slowly enough, the system will finally converge to a global minimum.

Whether and how the objective function will converge depends on how the temperature is changed, as well as the range of neighborhood, and how long the system will run for a fixed temperature. In practical applications of simulated annealing, various *cooling schedules* are suggested to drive the system to near-optimal solutions in an acceptable time. Moreover, for objective functions that are themselves random variables, such as the variance of IS simulation here, using a sample value as the objective function may cause problems in the SA algorithm. As a result, mean values with confidence intervals have to be calculated in each SA run. Authors in [47] suggest SA methods that will converge to global optimal values for noisy objective functions under certain conditions. We will apply their methods in our SA optimized IS method.

4.2.2 Simulated Annealing Optimized IS

In this section, we propose using simulated annealing to minimize the variance of IS estimators of the burst loss probability in an OBS network. We call this method "Simulated Annealing optimized Importance Sampling using Standard Clock" (SA-ISSC). The SA-ISSC algorithm is as follows:

- Assume we have k IS parameters $\{p_1, p_2, ..., p_k\}$ that can be tuned.

1. Define the A-cycle and execute an ordinary MC simulation to estimate the average length of A-cycles.

2. Initial setting for IS: Set the parameters in a way so that the important set will occur more frequently
3. Set temperature c to a high initial value
4. Neighborhood selection: sample new parameters $p_i(n+1)$ uniformly from a range $p_i(n) \pm \delta_i$
5. Use new parameter to run R replications of IS simulation and record the variance
6. Repeat step 5 for M times, calculate the mean values for the variance and use as function value at new setting
7. Use simulated annealing acceptance rules to see if this new set of parameters is accepted
8. Continue steps 4-7 for T times
9. Decrease the temperature and repeat steps 2-6, K times, and select the setting that has lowest overall mean as the final result

4.3 Simulation model and results

4.3.1 IS Model for OBS Network Simulation

In Figure 4.2, assume there are N nodes that are linked in tandem in the network, which can be used to model a route of a specific call in an OBS network. A burst that occupies these consecutive two links will be modeled as a customer being served in the node between the links it occupies. For example, a burst occupying link 1 and link 2 will be a customer in service at node 2. Each link, as opposed to each node as in a typical loss network, has a capacity of W wavelengths. Therefore, a burst arriving at node 2 will be blocked at that node if the current number of customers in node 1 plus current number of customers in node 2 is equal to W, or if the current number of customers in node 2 plus the current number of customers in node 3 is equal to W.

For the first and the last nodes, we assume that the links arriving and departing the network have enough bandwidth, so only a single link has to be considered. Therefore,

although each node has W wavelengths that can potentially be used, the actual number available is also determined by the adjacent nodes. Assume bursts of the call arrive at the first node and travel through the route, and leave at node N. In addition to the bursts of the call, there are local traffic streams that arrive at each link. We also assume these traffic streams also head for node N. The destination of local traffic is specified as node N in [7] so the decomposition is possible.

In our simulation model, the destinations can be generalized to be any node in the network. However, as of now we keep the original setting. The bursts from the call and local traffics are assumed to be exponentially distributed, with rates λ and γ, respectively. The service time, regardless of burst types, is also assumed to be exponentially distributed with rate μ for each node.

Likelihood Ratio

Using the *standard clock* method, distributions can be put together into an aggregated system event distribution. When an event happens, other random variables are then used to determine the type of this event. For a Markovian system, the aggregated event distribution is still a Poisson process. When an event occurs, the type of the new event can be determined from a discrete distribution with the rates of different events as masses.

For our OBS model, assuming that n_k represents the number of customers in node k, we have the overall event rate as

$$\Lambda = \lambda + (N-1)\gamma + \mu \sum_{k=1}^{N} n_k.$$

The probability of this event being a burst arrival from the call is $\frac{\lambda}{\Lambda}$, and the probability of a local traffic burst arriving at a node k is $\frac{\gamma}{\Lambda}$. Finally, the probability of this event being a service completion at node k is $\frac{\mu \cdot n_k}{\Lambda}$.

Consider the following IS biasing. Assume we change the burst arrival rate from the call, from the local traffic, and the service rate to be λ^*, γ^*, and μ^*, respectively. The new overall event rate will be

$$\Lambda^* = \lambda^* + (N-1)\gamma^* + \mu^* \sum_{k=1}^{N} n_k.$$

Since this is a Markov chain, the Radon-Nikodym derivative of a sample path is the

product of R-N derivatives of each transition. At the time an event happens under IS, the R-N derivative is as follows: (Assuming the time since last event is T)

$$\begin{aligned} L(T) &= \frac{P(\text{this event})}{P^*(\text{this event})} \\ &= \frac{\Lambda e^{-\Lambda T} \cdot \frac{h}{\Lambda}}{\Lambda^* e^{-\Lambda^* T} \cdot \frac{h^*}{\Lambda^*}} \\ &= e^{(\Lambda^* - \Lambda)T} \cdot \frac{h}{h^*} \end{aligned}$$

where

$$h = \begin{cases} \lambda, & \text{this event is a burst arrival of the call} \\ \gamma, & \text{this event is a burst arrival of local traffic} \\ \mu, & \text{this event is a departure from a node} \end{cases}$$

and h^* is the corresponding IS measure.

4.3.2 Simulation results and analysis

The SA-ISSC method is implemented using the Discrete Event/Queueing Network Simulation Framework, which is a modular simulation framework based on the Microsoft *.net* architecture.

The simulation using SA-ISSC works as follows: first a long simulation with the original λ, μ, and γ is done to estimate the length of an A-cycle, and the batch mean method is used to calculate the variance. Then M IS runs are executed, each containing R replications, using an initial "biased" λ^*, μ^*, and γ^*, that increase the occurrences of burst blocking.

Since this initial biasing is arbitrarily chosen, it is likely to cause overbiasing or underbiasing and thus has larger variance. The variance of this IS run is then used as the initial value of simulated annealing. Then the SA-ISSC algorithm introduced in previous section is used for minimizing the variance by sampling and accepting/rejecting parameters λ^*, μ^*, and γ^*. In the IS simulation, set A in the A-cycle method is defined to be the states at which the system is not empty. This also makes the A-cycles be truly regenerative.

In simulated annealing the parameters are randomly sampled in the neighborhood, therefore, it is possible that the method ends up with parameters that could take a very long time or even forever to complete a replication. To prevent this, a threshold can be set for the simulation time. If a replication takes longer than a specified threshold, this setting

of parameters will be rejected, and another sampling from the previous neighborhood will be done.

Two different OBS scenarios are simulated using SA-ISSC. The first is a single OBS node, which is actually an M/M/C/C loss node, with $\lambda = 4$, $\mu = 16$, and $\gamma = 0$. γ is set to zero, since there is only one node. This setting allows us to compare the result by SA-ISSC with the theoretical blocking probability, as well as the "asymptotically optimal" biasing parameters [28] that can be calculated in the M/M/C/C case. The other scenario is a five-node tandem network with $\lambda = 200$, $\mu = 700$, and $\gamma = 100$. This is a more realistic OBS network. In the system there are $W = 8$ wavelengths, and full wavelength conversion is assumed. That is, each of the eight wavelengths can be seen as the same channel. As long as one is available, any burst can use it regardless of which wavelength it is using in the previous node.

For the single-node scenario, the initial IS parameter are set to $\lambda^* = 40$, $\mu^* = 16$. For the five-node scenario, initial IS parameter is set to $\lambda^* = 600$, $\mu^* = 240$, and $\gamma^* = 110$. As mentioned above, these initial IS parameters are selected arbitrarily so that the blocking will happen more frequently, and could be over- or under- biased. For the IS setting, an initial sampling-and-run procedure is used to estimate the variance values around the initial IS parameter. Then an initial temperature is set so that 90% of the sampled parameters in the original neighborhood will be accepted. Moreover, R is set to 10 and M is set to 5. Temperature is kept constant for every fifteen parameter samples and acceptance/rejections according the SA algorithm above, then is decreased for 20 times by a factor of 0.85 each time.

Figures 4.3-4.7 show the results of the two simulation scenarios. Figure 4.3 shows the SA state evolution trajectory of the IS arrival rate (λ^*) and IS departure rate (μ^*) in the one-node case. We can see that the rates stabilize at $\lambda^* \approx 32$ and $\mu^* \approx 2$, which is the theoretical optimal value for the IS to have minimal variance. That is, $\lambda^* = \lambda \cdot W$, and $\mu^* = \mu/W$. Moreover, although not shown, the estimated blocking probability is consistent with the theoretical value calculated by the Erlang-B formula.

Figures 4.4 and 4.5 show the state evolution rates in a five node OBS network. From these two figures we can see that the rates also converge when using SA. We do not have analytical results for this scenario therefore it is not guaranteed that this biasing is optimal. Still, the important issue here from an engineering point of view is the significant reduction of simulation variance in realistic settings. From Figure 4.6 and 4.7, which show the values of

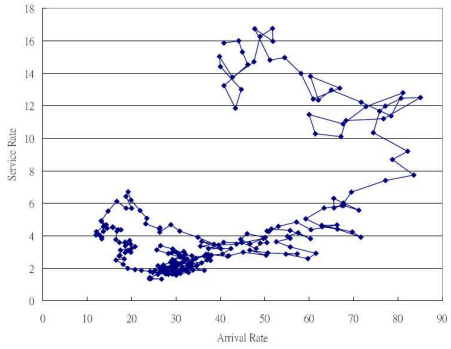

Figure 4.3: State Evolution of SA-ISSC for a Single OBS Node.

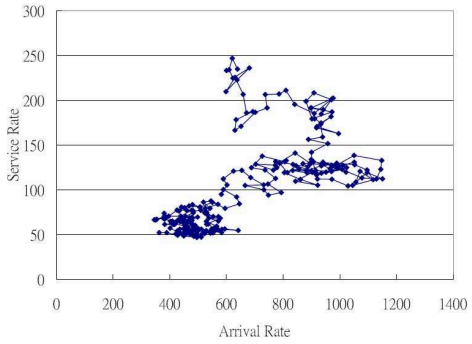

Figure 4.4: State Evolution of SA-ISSC for a 5-Node OBS Network.

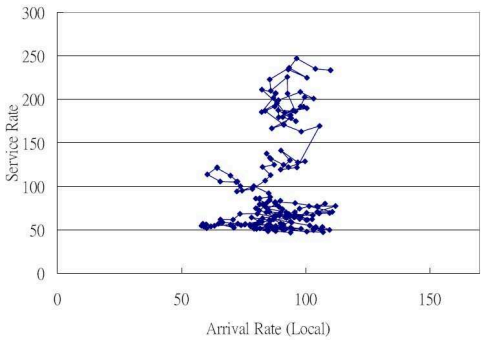

Figure 4.5: State Evolution of SA-ISSC for a 5-Node OBS Network.

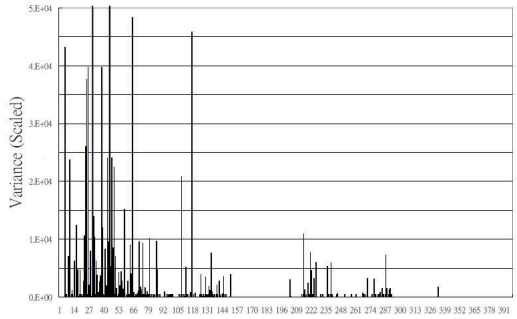

Figure 4.6: Variance Evolution of SA-ISSC for a Single OBS Node.

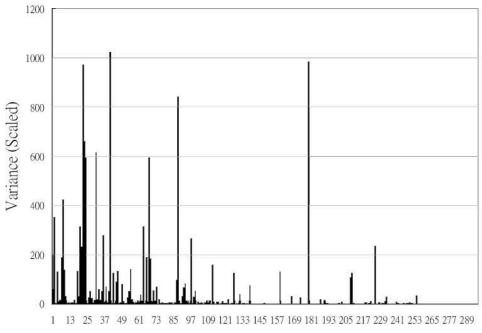

Figure 4.7: Variance Evolution of SA-ISSC for a 5-Node OBS Network.

variance along the SA evolution for both settings, we observe indeed very significant variance reduction in our IS simulations. Not surprisingly, the observed variances do become large sometimes during the course of the simulated annealing optimization procedure, but they both become small and stable toward the end of the simulation.

4.4 Conclusions

In this Chapter, we introduce the SA-ISSC method, which uses simulated annealing method that can deal with random objective functions to minimize the variance of IS estimates. It is hard to find a general guideline for a good IS biasing parameter setting when simulating complex systems. We have shown through the simulation examples of two configurations of OBS network, that by directly optimizing the variance using simulated annealing, SA-ISSC can be used in such cases to provide guidelines that can produce very favorable results.

Chapter 5

Performance Optimization Framework using Importance Sampling and the Response Surface Method

A performance evaluation is usually not complete by just knowing the performance under just one or a few system configurations. For example, one may want to know how the service availability will change as the traffic demand differs, or as the capacity of the system changes, etc. Moreover, a service provider might even want to optimize the system performance, so a satisfactory quality of service can be provided with minimum hardware cost.

Take modern software for example, when evaluating the performance measures of a newly developed software package helps the provider not only in knowing how the software performs, but also in providing potential customers with capacity planning suggestions. A customer may have an idea about how their service requests might look like (in the form of

arrival rates or distributions, etc.). Using such information together with the performance evaluation data, the provider is able to help the customer decide on the optimal hardware specifications (such as CPU speed or the amount of memory installation) that minimizes the overall cost. The latter is represented by a combined objective function of real hardware cost and the penalty incurred by service unavailability or violation of service level agreements (SLA).

Traditionally, such a capacity planning is done in two steps. First the performance values are evaluated using various settings of hardware and traffic load, then a table lookup is used to find the best setting that fits the requirements of the customer. This procedure has several disadvantages: First, the testing procedure could be slow, especially for highly available services. Moreover, different customers may have different load requirements or even different formulas for the unavailability penalty. The table lookup method cannot deal with such differences without a recalculation which may incur more testing to be done.

In this chapter, we propose an automatic framework for performance evaluation. We suggest using the Response Surface Methodology (RSM) to find the capacity setting that minimizes the total cost. As to the performance evaluation at different settings of the factors (parameters), Importance Sampling (IS) based testing/simulation can be used, for fast evaluation with the help of performance modeling. In practice, even when the targets to be evaluated are not rare, IS-based methods can still be used to reduce the variance of the simulation or testing. Moreover, since the response evaluation in an RSM step usually requires simulation/testing in a local neighborhood, the same IS trace can even be reused to further decrease the simulation time. In our proposed Response Surface-Importance Sampling (RS-IS) framework, we suggest an m-out-of-k reuse strategy, which is verified to be both accurate and efficient in our simulation examples.

In the following sections, we introduce the RS-IS framework and use software performance evaluation and capacity planning as an example. Actually, this framework can be used together with all the importance sampling models and methods introduced in this dissertation for performance evaluation and optimization.

5.1 The Response Surface-Importance Sampling (RS-IS) Framework

In this section, we introduce the automatic framework for performance evaluation. In our framework, the response surface methodology (RSM) is applied to optimize the cost, while importance sampling based simulation/testing, with the help of the performance models introduced in the previous section, is used to accurately and efficiently estimate the response in each of the RSM runs. The resulting novel RS-IS framework provides significant advantages in the automated performance evaluation of software.

5.1.1 Response Surface Methodology

Response surface methodology (RSM), first introduced by Box and Wilson [9], is a set of statistical and mathematical techniques that can be used to find optimal settings of parameters (usually called "factors") that minimize or maximize the objective function (also called the "response"). RSM can also be applied to stochastic simulation models, treating the system being evaluated as a black box [15].

Using RSM, first one estimates the local shape of the response, called the "response surface", then optimization is performed by choosing the steepest decent direction on the response surface.

Assume we would like to minimize a response $y \in R$ with factors $x_1, x_2, \ldots, x_n \in R$,, where

$$y = f(x_1, x_2, \ldots, x_n) + \varepsilon.$$

The true from of the function f may be unknown or very complicated, and ε represents other contributions to variability that are not accounted for in f, which also enables the use of RSM for stochastic responses. Usually ε is treated as a stochastic error with zero mean and variance σ^2. As a result, minimizing the expectation of y is the same as minimizing $\eta = f(x_1, x_2, \ldots, x_n)$

To approximate f, a two-phase approach is usually applied. In the beginning, where the starting point and a selected local area is considered to be far from the optimal solution, a first-order model is used to approximate the response. In general, the first-order model

looks like

$$\eta = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n.$$

The coefficients are estimated by using ordinary least square approximation using evaluated points within the current local area. The points to be evaluated are usually chosen according to certain two-level factorial experimental designs of resolution-III. After the model is created and tested for goodness-of-fit [43], a search in the steepest decent direction is performed until there is no improvement, or until other stopping criteria are satisfied.

The next step is to approximate the function f by using a second-order model. As the system evolves and approaches the optimal solution, the real response surface is likely to display a certain curvature. Using a second-order approximation allows to more precisely describe the real surface near the approximation. The second-order model looks like

$$\eta = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \sum_{i=1}^{n} \beta_{ii} x_i^2 + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{ij} x_i x_j.$$

For the second-order model, again the coefficients are determined using a least square approximation. The points, however, are usually chosen according to the Box-Behnken design or the Central Composite Design [42]. Since a stationary point in a second-order model might be a maximum, a minimum or a saddle point, and may fall out of the current local region, therefore extra fitting tests such as ridge analysis [42] should be done to determine whether a stationary point should be accepted or not.

The stationary point, accepted or not, will be used as the center point of the next local area. If it is accepted, which means a local minimal in the second-order model, a new second-order model can be constructed to do another search locally. On the other hand, if a stationary point is a maximum or a saddle point, which could mean the true optimal is still far away, a first-order model can be constructed.

There are many suggestions on when a RSM procedure should stop. Some recommend that RSM should stop after a single second-order model is done [18]. Other suggestions are more conservative. In this dissertation, we stop the RSM when the optimal solution does not improve significantly, as suggested in [43].

In [43], the authors suggest a framework for RSM with the following steps, which is consistent with many approaches, as stated in [42]. The steps are as the following:

1. Approximate the simulation response function in the current region of interest by a first-order model.
2. Test the first-order model for adequacy.
3. Perform a line search in the steepest descent direction.
4. Solve the inadequacy of the first-order model.
5. Approximate the objective function in the current region of interest by a second-order model.
6. Test the second-order model for adequacy.
7. Solve the inadequacy of the second-order model.
8. Perform canonical analysis.
9. Perform ridge analysis.
10. Accept the stationary point.
11. Determine a steepest descent direction from the second-order model.

We will follow these steps in this section.

5.1.2 The Response Surface-Importance Sampling Framework

Although carelessly setting the IS parameters may cause the IS to perform not as well as under optimal biasing, still, if we use the same IS biasing for two (original) distributions that are close enough from each other, the variance introduced by the two distributions would not differ too much.

For example, assume we have two systems with pure Poisson arrivals, one with rate λ and another with rate $\lambda + \Delta\lambda$. Consider using the optimized IS distribution for the system with arrival rate λ, say, Poisson distribution with rate λ^*, to simulate both systems. For any new arrival with inter-arrival time T, the likelihood ratio for these two system will be $e^{-\lambda T}/e^{-\lambda^* T}$ and $e^{-(\lambda+\Delta\lambda)T}/e^{-\lambda^* T}$, respectively. Theoretically, for both systems, the estimators using this IS distribution will both be *statistically unbiased*, no matter how large $\Delta\lambda$ is. The variances, on the other hand, will differ by increasing amounts as $\Delta\lambda$

increases. However, if $\Delta\lambda$ is small enough, the difference of the variances should fall in an acceptable range for practical evaluation. In such a case, we heuristically assume that the same IS distribution, even the same IS simulation trace, can be used to estimate both of the systems.

For each step that needs a response evaluation in the RSM, since the design points to be evaluated are all within a local neighborhood, we can use the same IS trace for more than once. One problem that might occur is the correlation between the evaluation of different points, if all of them are performed by using the same IS trace. To minimize this effect, if there are k points that need evaluation, we will select m points and perform new (optimized) IS simulation/testing for those, and drop the oldest m IS traces in history. The remaining $k - m$ points are then evaluated using the old IS traces with different likelihood ratio, according to the original distribution of these points. The smaller the m is, the faster the evaluation procedure will be, but the higher variance will be incurred. The selection of m will also depend on how large the local neighborhood is, and also on the value of the target to be estimated. For a larger local neighborhood or more rare targets, m should be larger.

The proposed RS-IS automation framework then works as follows:

1. Construct a performance model for the software being evaluated. The more detailed the model is, the more the IS testing/simulation is accurate.

2. Follow the automatic RSM procedure as introduced in section 2. For each time steps 1 and 5 are executed, discard the m oldest IS traces. Randomly select m points out of the k points selected from the experimental design and do new IS simluation/testing using (near) optimal IS biasing for each point. For the remaining $k - m$ points, use the existing IS trace to estimate the responses. Add the m newly created IS traces into the trace pool.

3. Repeat the above step until the system meets the stopping criteria of RSM.

5.2 Simulation Model and Results

5.2.1 Simulation Model

Consider a three queue software performance model, as shown in Figure 1.5. This model can be used to describe the behavior of distributed server software. Assume that an arrival of either a user request or some other software event is treated equally in the system. If the server is idle, the arriving request or event will be served immediately. Otherwise it will enter the queue and occupy a unit of the queue, until the server is available.

Here we assume the queueing system is first come, first served. Different queueing policies, such as priority queues, could also be used. If there is no space available in the queue, the arriving request/event will be dropped. For the inter-arrival time and service time, assume that both the new customer request and event arrivals are Poisson distributed with rates λ_1 and λ_2, respectively, and the service time of each servers are exponentially distributed with rates μ_1, μ_2, and μ_3. Server $i, i = 1, 2, 3$, has a memory (used as buffer in the queue) capacity of M_i units. The rates of dropping for customer requests and events are denoted as D_1 and D_2, respectively.

Assume that the cost for processor in every server is C_P per "service rate" unit, and the cost for memory is C_M per unit. Moreover, assume that the unit cost (penalty) for customer request and event dropping rates are C_{D1} and C_{D2}. We would like to use our RS-IS framework to find the best combination of service rates and memory capacities that minimize the total cost. That is, find $\mu_1', \mu_2', \mu_3', M_1', M_2'$, and M_3' for the following object function:

$$\min \sum_{i=1}^{3} (\mu_i' C_p + M_i' C_M) + \sum_{i=1}^{2} D_i^* C_{Di}.$$

For the RSM, the automatic framework as described earlier in this section is applied. To obtain the response estimates in each step of RSM, importance sampling is used. The IS distribution is selected by interchanging the overall arrival rates with the lowest service rate, which is suggested in [45] and is proved to be the only *asymptotically efficient* IS biasing method for a tandem Jackson network [21]. Assume $j = \arg\min_i \mu_i$, that is, server j has the minimum service rate among the three servers. The arrival and service rates under IS

Table 5.1: RS-IS Simulation Result

Simulation	μ_1^*	μ_2^*	μ_3^*	M_1^*	M_2^*	M_3^*
$m = 20$	58	461	393	1	2	3
$m = k$	55	470	390	1	2	3
OptQuest	55	467	388	1	2	3

m	Relative error	Time to complete
$m = 20$	$2.71 \cdot 10^{-2}$	23 minutes
$m = k$	$2.34 \cdot 10^{-2}$	51 minutes

is:

$$\begin{cases} \lambda_1^* = \mu_j \cdot \frac{\lambda_1}{\lambda_1+\lambda_2} \\ \lambda_2^* = \mu_j \cdot \frac{\lambda_2}{\lambda_1+\lambda_2} \\ \mu_j^* = \lambda_1 + \lambda_2 \\ \mu_i^* = \mu_i \qquad\qquad , i \neq j \end{cases}$$

For the first step of RSM, all k points, each having different sets of processor power and amounts of memory, are evaluated using this IS biasing method. The traces of the events in each simulation are saved. For each of the subsequent steps in RSM, m out of k points are randomly selected, and new estimations are performed using this IS method, while the IS traces of these m IS runs are saved and replace the oldest m IS traces in the trace pool. The other $k - m$ points are evaluated using the $k - m$ IS traces that are not replaced. This procedure continues until the RSM stops.

5.2.2 Simulation Results

The model shown in the previous section with $\lambda_1 = 200$, $\lambda_2 = 100$, $C_1 = 10$, $C_2 = 1500$, $C_{D1} = 10^5$, and $C_{D2} = 10^4$ is used for simulation using the RS-IS framework in this section.

The feasible region of service rates and memory sizes are $\mu_i \geq 50$, and $M_i \geq 1$, which can be seen as the most basic system configuration that can be provided for the servers. For the IS sampling strategy, two settings are used with different m values: One with $m = 20$ and the other with $m = k$ in both of the the first-order and second order models.

For the scenario in which $m = k$, every point to be evaluated is actually simulated using a new set of IS distributions that is optimized for the factor setting of the point. Table 5.1 shows the optimized factors, average relative error (of the estimation of dropping

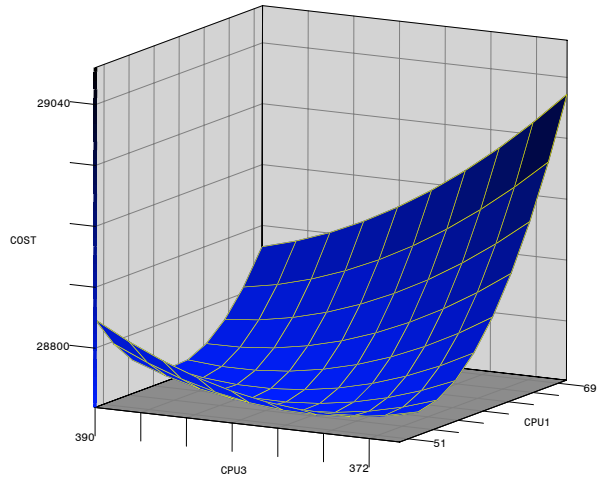

Figure 5.1: Response Surface for factors CPU1 and CPU3, case $m = k$.

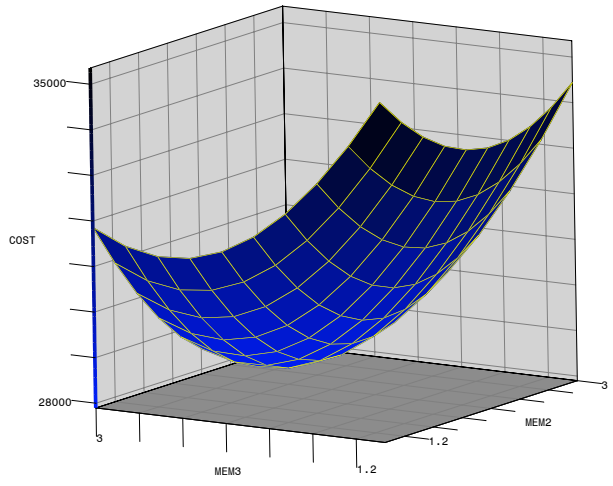

Figure 5.2: Response Surface for factors MEM2 and MEM3, case $m = k$.

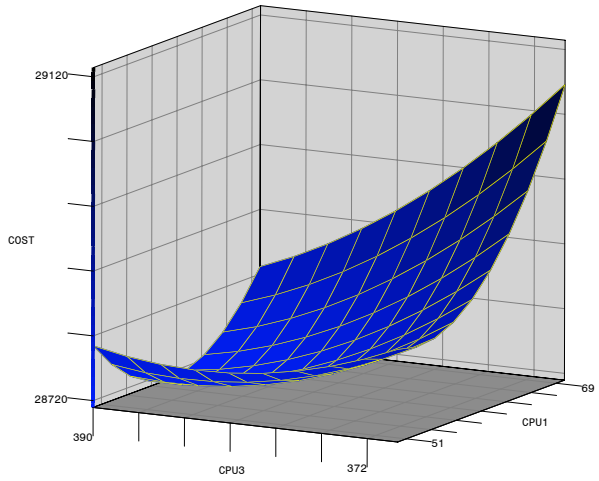

Figure 5.3: Response Surface for factors CPU1 and CPU3, case $m = 20$.

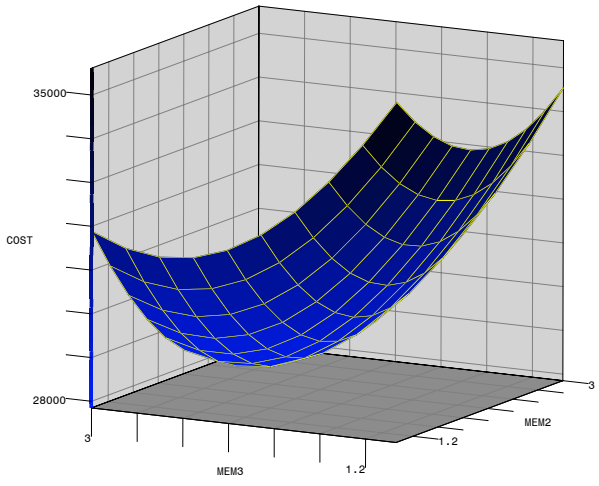

Figure 5.4: Response Surface for factors MEM2 and MEM3, case $m = 20$.

errors), and the time to finish the simulation for both IS strategies.

From the table we can see that, although the case in which $m = 20$ has about 15% more error than the case in which $m = k$, the estimated optimal setting is actually in the same proximity of the $m = k$ case. This optimized factor setting is also verified by the OptQuest program provided with the Arena simulation software – it takes around four full hours for OptQuest to run the optimization. However, the $m = 20$ case only takes less than half of the time to finish the whole RS-IS procedure. Figures 5.1-5.4 show the final response surface near the optimal value for both cases, with respect to the processor service rates (denoted as CPUi) and memory installations (denoted as MEMi), respectively. From the figures we can see that the two response surfaces actually look almost the same near the final local neighborhood, which is also an indication of the accuracy of the $m = 20$ case.

5.3 Conclusions

In this chapter, we propose an automatic framework for performance optimization using response surface methodology (RSM) and importance sampling (IS). The use of RSM not only allows us to accurately optimize the objective function, but also gives an idea what the response surface looks like around different factor settings.

We also propose using the same IS traces for more than one evaluation of points, due to the nature of the local neighborhood evaluation in each RSM step. We compare this method to the method that uses a new simulation for each point in our example, and the performance is almost as good, while it costs much less time. This proposed framework can not only be used in capacity planning as the example in this chapter, but also for other optimization tasks, e.g., maximizing the number of customers under cost restraints.

Chapter 6

Summary of the Dissertation and Future Work

6.1 Summary of achievements

As the availability of services provided by modern systems becomes higher and higher, efficient evaluation of such systems becomes an important factor in both design and implementation. Moreover, to provide customers with reasonable service level agreements and capacity suggestions, the providers have to first know the quality of their services. In this dissertation, we have developed efficient simulation methods that can be used to evaluate highly available services, and an automatic framework for metamodeling and performance optimization.

Among these methods, the static ISSC method is the simplest and can provide answers with bounded related variance in a lightly loaded system, where the request arrival rates are small. The adaptive ISSC method, on the other hand, can deal with more different situations, while still providing very favorable results. Finally, the SA-ISSC method is more complex, but can be widely used in many scenarios, even when the system is too complex to analyze for the best IS parameters. For the cases in which the performance of a system needs to be evaluated in various different settings, or performance itself needs to be

optimized, the RS-IS automatic framework provides a very convenient way to perform these tasks automatically. The reuse of IS traces can further reduce the time and effort needed for testing or simulation. As seen in the first chapter, figure 1.7 shows the relationship of the models and methods introduced in this dissertation.

For these methods, the algorithms and results have been published in or submitted to related conferences and journals. [31] is an introduction to static and adaptive ISSC on traffic groomed optical networks. [59] uses adaptive ISSC to validate with an decomposition based numerical method. Moreover, [29] generalizes the adaptive ISSC to high capacity cases, and adds robust network server models in the simulation capabilities. In [32], the SA-ISSC method is proposed and used for evaluating OBS networks that have dynamic simultaneous resource possession. Finally, the RS-IS framework is introduced in [30] for performance metamodeling and capacity planning of modern software.

In summary, these methods together provide a set of tools, based on importance sampling and related methodologies, for efficient estimation of highly available services by either simulation or testing.

6.2 Future work

Although we have already proposed a rather complete set of methods for efficient performance evaluation, there is still future work that could further improve the methods we propose in this dissertation.

First, it is always good to find more methods that have mathematically proven properties, such as BRE or Asymptotically Efficiency. Although in practice these methods might not perform better than heuristic methods, they do guarantee an upper bound for estimation variance as the estimator goes to zero. Moreover, for more complex performance models, such as those with generalized arrival and service distributions, the approximation of the "optimal guideline" is not easily done. Even when possible, there always exists a trade off between simplicity and accuracy in these approximations. How to find an efficient evaluation method for such models would be another good direction for future work.

In addition, it would also be beneficial to apply other stochastic optimization methods, such as genetic algorithms or steepest gradient decent, in the place of Simulated Annealing for directly optimizing the estimation variance. An automatic framework that adaptively

adjusts the optimization parameters will also be helpful in practice.

Finally, the IS reuse method we propose in the RS-IS framework is a simple m-out-of-k policy. It is possible that applying a more complex policy according to the rarity of the estimated events can further reduce the overall time to finish the whole RS-IS procedure. These all remain good future directions for research.

Bibliography

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines.* Wiley & Sons Ltd., 1989.

[2] W. A. Al-Qaq, M. Devetsikiotis, and J. K. Townsend, "Stochastic Gradient Optimization of Importance Sampling for the Efficient Simulation of Digital Communication Systems," *IEEE Trans. Communications*, vol. 43, no. 12, December 1995.

[3] C. Alexopoulos and B. Shultes, "Estimating Reliability Measures of Highly-Dependable Markovian Systems using Balanced Likelihood Ratios," *IEEE Transactions on Reliability*, vol. 50, no. 3, pp. 265–280, 2001.

[4] L. L. H. Andrew, "Fast Simulation of Wavelength Continuous WDM Networks," *IEEE/ACM Trans. Network*, vol. 12, no. 4, pp. 759–765, Aug 2004.

[5] T. Battestilli and H. Perros, "An introduction to optical burst switching," *IEEE Comm.Optical Magazine*, vol. 41, pp. S10 – S15, 2003.

[6] ——, "End-to-end burst loss probabilities in an obs network with simultaneous link possesion," in *The Third International Workshop on Optical Burst Switching, WOBS3 (co-located with Broadnets 2004)*, October 2004.

[7] ——, "A performance study of an optical burst switched network with dynamic simultaneous link possession," *Computer Networks*, vol. 50, no. 2, pp. 219–236, February 2006.

[8] G. Bilbro, R. Mann, T. M. III, W. Snyder, D. E. V. den Bout, and M. W. White, "Optimization by Mean Field Annealing," in *Advances in Neural Information Processing Systems.* San Mateo, CA: Morgan-Kaufmann, 1989.

[9] G. Box and K. Wilson, "On The Experimental Attainment of Optimum Conditions," *Journal of Royal Statistical Society*, vol. 13, no. 1, pp. 1–38, 1951.

[10] J. A. Carrasco, "Failure Transition Distance-Based Importance Sampling Schemes for the Simulation of Repairable Fault-Tolerant Computer Systems," *IEEE Transactions on Reliability*, vol. 55, no. 2, pp. 207–236, June 2006.

[11] C.-S. Chang, P. Heodelberger, and P. Shahabuddin, "Fast simulation of packet loss rates in a shared buffer communication switch," *ACM Transactions on Modeling and Computer Simulation*, vol. 5, no. 4, pp. 306–325, 1995.

[12] P. de Boer, D. Kroese, S. Mannor, and R. Rubinstein, "A Tutorial on the Cross-Entropy Method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, Jan 2005.

[13] M. Devetsikiotis, W. Al-Qaq, J. A. Freebersyser, and J. K. Townsend, "Stochastic Gradient Techniques for the Efficient Simulation of High-Speed Networks Using Importance Sampling," in *Proc. IEEE Global Telecom. Conf., Globecom '93*, Houston, Dec. 1993.

[14] M. Devetsikiotis and J. K. Townsend, "Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks," *IEEE/ACM Trans. Networking*, vol. 1, no. 3, pp. 293–305, June 1993.

[15] J. Donohue, E. Houck, and R. Myers, "Simulation Designs for the Estimation of Response Surface Gradients in the Presenced of Model Misspecification," *Management Sciense*, vol. 41, no. 2, pp. 244–262, 1995.

[16] R. Dutta and G. Rouskas, "Traffic grooming in wdm networks: Past and future," *IEEE Network*, vol. 16, no. 6, pp. 46–56, 2002.

[17] M. Falkner, M. Devetsikiotis, and I. Lambadaris, "Fast Simulation of Networks of Queues with Effective and Decoupling Bandwidths," *ACM Transactions on Modeling and Computer Simulation*, vol. 9, no. 1, pp. 45–58, Jan 1999.

[18] M. Fu, "Optimization via Simulation: A Review," *Annals of Operations Research*, vol. 53, pp. 199–247, 1994.

[19] M. C. Fu and J. Q. Hu, *Conditional Monte Carlo: Gradient estimation and optimization applications.* Kluwer Academic Publisher, 1997.

[20] P. Glasserman, *Gradient estimation via perturbation analysis*. Boston: Kluwer, 1991.

[21] P. Glasserman and S. Kou, "Analysis of An Importance Sampling Estimator for Tandem Queues," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 5, no. 1, pp. 22–42, Jan 1995.

[22] P. W. Glynn, "Likelihood Ratio Gradient Estimation for Stochastic Systems," *Comm. ACM*, vol. 33, no. 10, pp. 75–84, Oct. 1990.

[23] W. Gutjahr, "Importance Sampling of Test Cases in Markovian Software Usage Models," *Probability in the Engineering and Informational Sciences*, vol. 11, pp. 11–36, 1997.

[24] ——, "Software dependability evaluation based on Markov usage models," *Performance Evaluation*, vol. 40, pp. 199–222, 2000.

[25] P. E. Heegaard, "Adaptive optimisation of importance sampling for multi-dimensional state space models with irregular resource boundaries," in *Proc. 13th Nordic Teletraffic Seminar, Trondheim, Norway*, 1996, pp. 176–189.

[26] ——, "Efficient Simulation of Network Performance by Importance Sampling," in *Proc. 15th International Teletraffic Congress - ITC 15*, 1997.

[27] ——, "A Scheme for Adaptive Biasing in Importance Sampling," *International Journal of Electronics and Communications*, vol. 52, October 1998.

[28] P. Heidelberger, "Fast Simulation of Rare Events in Queueing and Reliability Models," *ACM Transactions on Modeling and Computer Simulation*, vol. 5, no. 1, pp. 43–85, Jan. 1995.

[29] C. Hsu and M. Devetsikiotis, "An Adaptive Approach to Accelerated Evaluation of Highly Available Services," submitted to ACM Transaction of Modeling and Computer Simulation (TOMACS).

[30] ——, "A framework for automatic software performance evaluation and optimization using response surface methodology and importance sampling," submitted to the 40th Annual Simulation Symposium.

[31] ——, "An Adaptive Approach to Fast Simulation of Traffic Groomed Optical Networks," in *Proceedings of Winter Simulation Conference 2004*, Dec 2004, pp. 612–620.

[32] C. Hsu, M. Devetsikiotis, and S. D. Roberts, "Fast Simulation of Optical Burst Switching Networks Using Simulated Annealing," in *Proceedings of 2006 IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, September 2006, pp. 283–292.

[33] C.-F. Hsu and T.-L. Liu, "On deflection routing in optical burst-switched networks," *Journal of High Speed Networks*, vol. 14, no. 4, pp. 341–362, 2005.

[34] M. hua Hsieh, "Adaptive monte carlo methods for rare event simulations," in *Proceedings of the 2002 Winter Simulation Conference*, 2002.

[35] J. Kiefer and J. Wolfowits, "Stochastic estimation of the maximum of a regression function," *Annals of Mathmatical Statistics*, vol. 23, pp. 462–466, 1952.

[36] S. Kiripatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, pp. 671–680, May 1983.

[37] P. E. Lassila and J. T. Virtamo, "Efficient importance sampling for monte carlo simulation of loss systems," in *ITC-16 Conference on Teletraffic Engineering in a Competitive World*, Elsevier, Amsterdam, The Netherlands, June 1999, pp. 787–796.

[38] ——, "Nearly optimal importance sampling for monte carlo simulation of loss systems," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 10, no. 4, Oct 2000.

[39] H. Li and I. L.-J. Thng, "Performance analysis of a limited number of wavelength converters in an optical switching node," *IEEE Photonics Technology Letters*, vol. 17, no. 5, pp. 1130–1132, May 2005.

[40] M. Mandjes, "Fast simulation of blocking probabilities in loss networks," *European Journal in Operational Research*, vol. 101, pp. 393–405, 1997.

[41] D. Morato, M. Izal, J. Aracil, E. Magana, and J. Miqueleiz, "Blocking time analysis of obs routers with arbitrary burst size distribution," in *IEEE Global Telecommunications Conference. GLOBECOM '03.*, vol. 5, 2003, pp. 2488–2492.

[42] R. H. Myers, *Response Surface Methodology.* Wiley, 2002.

[43] H. G. Neddermeijer, G. J. van Oortmarssen, N. Piersma, and R. Dekker, "A Framework for Response Surface Methodology for Simulation Optimization," in *Proceedings of the 2000 Winter Simulation Conference*, 2000, pp. 129–136.

[44] V. F. Nicola, P. Shahabuddin, and M. K. Nakayama, "Techniques for Fast Simulation of Models of Highly Dependable Systems," *IEEE Transactions on Reliability*, vol. 50, no. 3, pp. 246–264, 2001.

[45] S. Parekh and J. Walrand, "A Quick Simulation Method for Excessive Backlogs in Networks of Queues," *IEEE Trans. Automat. Contr.*, vol. AC-34, no. 1, pp. 54–66, Jan. 1989.

[46] S. Parlar and E. Topuz, "Determination of burst drop probability in optical burst switched networks with monte carlo simulation," in *7th International Conference on Transparent Optical Networks*, vol. 2, July 2005, pp. 429–432.

[47] A. A. Prudius and S. Andradottir, "Two Simulated Annealing Algorithms for Noisy Objective Functions," in *Proceedings of the 2005 Winter Simulation Conference (WSC 2005)*, 2005, pp. 797–802.

[48] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematics Statistics*, vol. 22, pp. 400–407, 1951.

[49] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer-Verlag, 1995.

[50] R. Y. Rubinstein, "Sensitivity analysis of discrete event systems by the "push-out" method," *Annals of Operations Research*, vol. 39, pp. 229–237, 1992.

[51] M. Schlosser, H. Woesner, and A. Schroth, "Simulation model for an optical burst switched network," in *The 13th IEEE Workshop on Local and Metropolitan Area Networks (LANMAN 2004)*, April 2004, pp. 103–108.

[52] P. Shahabuddin, "Importance sampling for the simulation of highly reliable markovian systems," *Management Science*, vol. 40, no. 3, pp. 333–352, 1994.

[53] P. J. Smith, M. Shafi, and H. Gao, "Quick simulation: A review of importance sampling techniques in communications systems," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 4, pp. 597–613, May 1997.

[54] R. Srinivasan, *Importance Sampling : Applications in Communications and Detection* . Springer, 2002.

[55] F. Tam, "On the Development of an Open Standard for Highly Available Telecommunication Infrastructure Systems." in *EUROMICRO*, 2002, pp. 347–351.

[56] J. Trezza, H. Hamster, J. Iamartino, H. Bagheri, and C. DeCusatis, "Parallel Optical Interconnects for Enterprise Class Server Clusters: Needs and Technology Solutions ," *IEEE Communications Magazine*, vol. 41, no. 2, pp. S36–S42, Feb 2003.

[57] P. Vakili, "Using a standard clock technique for efficient simulation," *Operations Research Letters*, vol. 10, pp. 445–452, 1991.

[58] F. J. Vazquez-Abad, L. L. H. Andrew, and D. Everitt, "Estimation of blocking probabilities in cellular networks with dynamic channel assignment," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 12, no. 1, Jan 2002.

[59] A. N. Washington, C. Hsu, H. G. Perros, and M. Devetsikiotis, "Approximation Techniques for the Analysis of Large Traffic-Groomed Tandem Optical Networks," in *Proceedings of the 38th Annual Simulation Symposium*, Apr 2005.