Fast Simulation of Queueing Networks Using Stochastic Gradient Techniques and Importance Sampling

M. Devetsikiotis
W. A. Al-Qaq
J. A. Freebersyer
J. K. Townsend

Center for Communications and Signal Processing
Department of Electrical and Computer Engineering
North Carolina State University

TR-94/5
April 1994

Submitted to *IEEE/ACM Transactions on Networking*

Fast Simulation of Queueing Networks Using Stochastic Gradient Techniques and Importance Sampling

Michael Devetsikiotis†, *Member, IEEE*
Wael A. Al-Qaq, *Student Member, IEEE*
James A. Freebersyser, *Student Member, IEEE*
J. Keith Townsend, *Member, IEEE*

†Dept. of Systems & Computer Engineering
Carleton University
Ottawa, Ontario K1S 5B6, Canada
Tel: (613) 788-2600, Fax: (613) 788-5727

Dept. of Electrical & Computer Engineering
North Carolina State University
Raleigh, North Carolina 27695-7911, U.S.A.
Tel: (919) 515-5200, Fax: (919) 515-5523

Abstract

To obtain large speed-up factors in Monte Carlo simulation using importance sampling (IS), the modification, or bias of the underlying probability measures must be carefully chosen. In this paper we present two stochastic gradient optimization techniques that lead to favorable IS parameter settings in the simulation of queueing networks, including queues with bursty traffic. Namely, we motivate and describe the *Stochastic Gradient Descent* (SGD) algorithm, and the *Stochastic (Important Event) Frequency Ascent* (SFA) algorithm.

We demonstrate the effectiveness of our algorithms by applying them to the problem of estimating the cell loss probability of several queueing systems: first, a queue with an Interrupted Bernoulli arrival process, geometric service times, and finite capacity K (denoted here by IBP/Geo/1/K); then, single and tandem configurations of queues with two arrival streams, a Modified Interrupted Bernoulli stream and a Markov Modulated Bernoulli stream with batch arrivals, deterministic service times, and finite capacity K (denoted here by M-IBP+MMBBP/D/1/K). Such queueing systems are useful building blocks in performance models for ATM nodes and networks. Speed-up factors of 1 to 8 orders of magnitude over conventional Monte Carlo simulation are achieved for the examples presented.

1. Manuscript received: _____

2. This work was supported in part by the Center for Communications & Signal Processing, North Carolina State University, and by the Telecommunications Research Institute of Ontario, Project on Interconnected Networks for Multimedia Traffic. James A. Freebersyser is supported by the U.S. Air Force Palace Knight Program. Wael A. Al-Qaq is an IBM Graduate Fellow.

3. Portions of this paper were presented at the IEEE Global Telecommunications Conference, GLOBECOM '93, Houston, November 29 – December 2, 1993. Other portions of this paper have been submitted for presentation at GLOBECOM '94, San Fransisco, November 27 – December 1, 1994.

1 Introduction

A significant problem when using Monte Carlo (MC) simulation for the performance analysis of communication networks is the long run times required to obtain accurate estimates. Under the proper conditions, Importance Sampling (IS) is a technique that can speed up simulations involving rare events of network (queueing) systems [1, 2, 3, 4, 5].

Large speed-up factors in simulation run time can be obtained by using IS if the modification or bias of the underlying probability measures is carefully chosen. It is not typically possible to analytically minimize the variance of the importance sampling estimator, or IS-variance, with respect to the IS biasing parameter settings for large (multiqueue) networks with bursty traffic. Fast simulation methods based on Large Deviation Theory (LDT) [1, 3] utilize asymptotical analytical knowledge, and analytical/numerical manipulations of the system statistics which are not feasible for many realistic systems. An example of the complexity involved with such analysis is the recent extension of LDT-type solutions to a large class of multiqueue networks with input traffic consisting of multiple Markov-modulated Poisson streams [5].

A technique for finding near-optimal bias parameter values, based on repetitive, short simulation runs and statistical measures of performance, which included statistical estimates of the estimator variance has been previously presented [6]. Such techniques are not restricted to any specific type of random process, and do not require any knowledge of the internal workings of the system being simulated. The Mean-Field Annealing (MFA) global optimization algorithm, which is a form of Simulated Annealing (SA), has also been used for finding near-optimal IS biasing parameter values for queueing system simulation [7]. The MFA approach is very effective and general, but can be affected by long run times and dimensionality problems.

In [8] we formulated stochastic gradient techniques in the different context of digital communication systems. In this paper we present two stochastic gradient optimization techniques for the near-minimization of IS-variance in the simulation of queueing systems. Both the *Stochastic Gradient Descent* (SGD) algorithm, and the *Stochastic (Important Event)*

Frequency Ascent (SFA) algorithm involve MC estimates of the gradient of a cost function using *likelihood ratio methods*, as described in [9, 10, 11, 12, 13]. The SGD uses estimates of the IS-variance and its gradient with respect to the biasing parameters, and follows a stochastic steepest descent path to near-optimal IS settings. The SFA uses estimates of the biased *frequency of important events* (FIE), e.g., cell-loss frequency, and its gradient with respect to the biasing parameters, and searches for the minimum IS-variance in the direction of steepest ascent in the FIE.

We illustrate the effectiveness of the techniques with numerical examples. Large speed-up factors (1 to 8 orders of magnitude) over conventional MC simulation, at low cell loss probabilities (e.g., 10^{-13}) are obtained for queueing systems with bursty traffic, including single IBP/Geo/1/K queues, and single and tandem M-IBP+MMBBP/D/1/K queues.

2 Simulation of Communication Networks

2.1 MC Estimation in Network Analysis

Let $\mathbf{X}_i$ be the vector of observations relevant to a slotted-time communications network at time i. Assume that $\{\mathbf{X}_i\}_{i\geq 0}$ is a discrete-time Markov chain with transition matrix $\mathbf{P}$. In addition, assume that $\{\mathbf{X}_i\}_{i\geq 0}$ has a steady-state distribution, and converges in distribution to $\mathbf{X}$. Following the formulation in [14], the goal is to estimate the expectation $E[f(\mathbf{X})]$ of some function $f(\mathbf{X}) = h(\mathbf{X})/g(\mathbf{X})$. Let $\mathbf{r}$ be a regeneration state. Then, the expectation of f can be written as

$$E[f] = \frac{E\left[\sum_{i=0}^{\tau_1 - 1} h(\mathbf{X}_i)\right]}{E\left[\sum_{i=0}^{\tau_1 - 1} g(\mathbf{X}_i)\right]} \tag{1}$$

where $\mathbf{X}_0 = \mathbf{r}$, and τ_1 is the first time greater than zero that $\mathbf{X}_i = \mathbf{r}$.

2.2 Efficient Simulation Using IS

To obtain an estimator of (1), first write $H(\mathbf{s}) = \sum_{i=0}^{\tau_1 - 1} h(\mathbf{X}_i)$ and $G(\mathbf{s}) = \sum_{i=0}^{\tau_1 - 1} g(\mathbf{X}_i)$, where $\mathbf{s}$ denotes a sample path in the evolution of the system under study. Let $E_P[G(\mathbf{s})]$ denote the expectation of $G(\mathbf{s})$ with respect to the probability measure $P(\mathbf{s})$. IS is based on the

observation that the expectation $E[G(\mathrm{s})]$ under measure P can be written as $E_P[G(\mathrm{s})] = E_{P^*}[G(\mathrm{s})L^*(\mathrm{s})]$, where $L^*(\mathrm{s}) = P(\mathrm{s})/P^*(\mathrm{s})$ and provided that $P^*(\mathrm{s}) \neq 0$ whenever $G(\mathrm{s})P(\mathrm{s}) \neq 0$. L^* is a likelihood ratio or, in the language of IS, a weight function. Then $E[f]$ can be estimated by

$$\widehat{E_{P^*}[f]} = \frac{1/N \sum_{k=1}^{N} \sum_{i=0}^{\tau_1 - 1} h(\mathbf{X}_{ik}) L_{ik}^*}{1/M \sum_{k=1}^{M} \sum_{i=0}^{\tau_1 - 1} g(\mathbf{X}_{ik}) L_{ik}^*} \tag{2}$$

where $L_{ik}^* = P(\mathbf{X}_{0k}, \ldots, \mathbf{X}_{ik})/P^*(\mathbf{X}_{0k}, \ldots, \mathbf{X}_{ik})$. In general, the numerator and denominator of (2) can be estimated separately, with $M \neq N$ and different IS distributions [14]. Let the transition probabilities of the Markov chain be $p(\mathbf{X}_j, \mathbf{X}_{j+1})$. Within each regeneration cycle (RC) k, the individual weights $p(\mathbf{X}_{ik}, \mathbf{X}_{i+1,k})/p^*(\mathbf{X}_{ik}, \mathbf{X}_{i+1,k})$ must be distinguished from the total or cumulative weight L_{ik}^*, at time i. Furthermore, when more than one independent random event determines the transition (e.g., arrivals, service completions), individual weights are again the product of component weights corresponding to these more fundamental random events. Then, by the Markov chain property, and with initial distribution unbiased under IS,

$$L_{ik}^* = \prod_{j=0}^{i-1} p(\mathbf{X}_{jk}, \mathbf{X}_{j+1,k}) / \prod_{j=0}^{i-1} p^*(\mathbf{X}_{jk}, \mathbf{X}_{j+1,k}) \tag{3}$$

In (2) above, the likelihood ratio (or weight) at time i during the simulation depends on all random transitions which previously occurred in the same RC. For tandem networks, this time dependence will become slightly more complicated.

An additional motivation to use regeneration techniques is to avoid the deleterious effects of large system memory on the efficiency of IS. As was shown in [2], nonregenerative IS breaks down as the length of the simulation approaches infinity. From an IS standpoint, the memory of the system is increasing within each RC. For cases where true regenerations are rare, techniques based on approximate regeneration [15], batch means, or A-cycles [5], can be used to obtain approximately independent trials.

In (2) it is implied that IS is implemented in a static way, where the modified or biased measures P^* do not depend on the state $\mathbf{X}_i$ at time i. However, the requirements of regeneration can be in conflict with static IS [7]. Under certain conditions for the simulation of Markov chains, the optimal IS is dynamic [14]. In order to combine the advantages of

regenerative simulation with efficient IS, IS can be used dynamically within each RC, first achieving efficient estimation of the rare event probability involved, and subsequently driving the system back to the regeneration state [7].

2.3 Statistical Optimization of the IS Estimator

It is well known that the general, non-parametric, globally optimal IS measure represents essentially a tautology, since it requires knowledge of the quantity $E[f]$ to be estimated. Most useful and practical IS schemes are parametric.

In the parametric case, finding the optimal IS settings can be posed as a multidimensional, nonlinear optimization problem, where the values of the IS parameters must be set to optimize some measure of performance, usually the estimator variance, $\sigma^2_{IS}(P, P^*)$.

Assuming an exact, closed-form representation of the IS-variance is not available we have proposed using statistical measures of performance, which are statistical estimates of the variability (scatter) of the MC observations involved, and asymptotical estimates of the estimator variance, $\hat{\sigma}^2_{IS}(P, P^*)$, with respect to the IS parameter values [6, 7]. In [7] we used mean field annealing (MFA), a stochastic global optimization algorithm, to perform this minimization.

Although very general, the MFA approach is potentially slow, especially as the dimensionality of the search (i.e., the number of IS parameters) increases. Therefore, techniques that perform a *directed* search through the parameter space (e.g., using derivative information) while requiring a smaller number of cost function evaluations can provide attractive alternatives to stochastic annealing techniques.

3 Stochastic Gradient Techniques

3.1 MC Estimation of Gradients

Performance measures of communication systems and networks often take the form of an expectation $a(\boldsymbol{\theta})$ that depends on a vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ of parameters. Examples of such performance measures are the bit-error-rate in digital links, or the cell loss probability, mean

delay, and throughput in networks. When analyzing or designing such a stochastic system, it is often desirable to calculate not only $a(\boldsymbol{\theta})$ but also its gradient $\nabla a(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Knowledge of $\nabla a(\boldsymbol{\theta})$ facilitates sensitivity analysis, interpolation techniques, and most importantly design optimization methods, where the vector $\boldsymbol{\theta}_{opt}$ is sought that minimizes (or maximizes) $a(\boldsymbol{\theta})$ [10, 12].

As in the case of the original expectation $a(\boldsymbol{\theta})$, analytical calculation of $\nabla a(\boldsymbol{\theta})$ is often intractable. Numerical calculation may be feasible or even advantageous under certain conditions, but is usually inefficient when numerical evaluation of $a(\boldsymbol{\theta})$ is time consuming [11].

The Monte Carlo estimation of derivatives and gradients of expectations, based on likelihood ratios has been previously investigated in [9, 10, 11, 12, 13]. In a rather general setting this approach introduces a change of measure and the corresponding likelihood ratio, and then, by essentially interchanging the expectation and derivative operators, expresses the derivative in question as the expectation of a new random variable. This expectation can be then estimated using MC simulation.

Because of their "semi-analytic" nature, such estimates have obvious advantages over finite difference approximations [11]. A comparison of likelihood ratio techniques with *perturbation analysis* (e.g., [16]) is given in [10]. In [13] likelihood ratio techniques are analyzed in the context of highly dependable Markovian models, and IS is successfully applied to increasing the efficiency of the gradient estimators when rare events are involved.

Focusing on a discrete-time Markov chain $\{\mathbf{X}_n\}_{n\geq 0}$ and following the notation in [12], an expectation $a(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}} g(\boldsymbol{\theta}, \mathbf{X}_0, \ldots, \mathbf{X}_T)$ can also be written as

$$a(\boldsymbol{\theta}) = E_{\boldsymbol{\theta}'} g(\boldsymbol{\theta}, \mathbf{X}_0, \ldots, \mathbf{X}_T) L_T(\boldsymbol{\theta}, \boldsymbol{\theta}', \mathbf{X}_0, \ldots, \mathbf{X}_T) \tag{4}$$

where for any instant n

$$L_n(\boldsymbol{\theta}, \boldsymbol{\theta}', \mathbf{X}_0, \ldots, \mathbf{X}_n) = \frac{\mu(\boldsymbol{\theta}, \mathbf{X}_0)}{\mu(\boldsymbol{\theta}', \mathbf{X}_0)} \prod_{i=0}^{n-1} \frac{P(\boldsymbol{\theta}, \mathbf{X}_i, \mathbf{X}_{i+1})}{P(\boldsymbol{\theta}', \mathbf{X}_i, \mathbf{X}_{i+1})} \tag{5}$$

In the following we will use $L_n(\boldsymbol{\theta}, \boldsymbol{\theta}')$ instead of $L_n(\boldsymbol{\theta}, \boldsymbol{\theta}', \mathbf{X}_0, \ldots, \mathbf{X}_n)$ in order to simplify the notation. $E_{\boldsymbol{\theta}}$ implies that the transition matrix $\mathbf{P}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$, $P(\boldsymbol{\theta}, \mathbf{X}_i, \mathbf{X}_{i+1})$ is the

transition probabilities under θ, $\mu(\theta, \mathbf{X}_0)$ is the initial state distribution under θ, and T is a finite stopping time. Then, the gradient $\nabla a(\theta)$ is given by

$$\nabla a(\theta) = E_{\theta'} \nabla \tilde{g}(\theta) \tag{6}$$

or

$$\frac{\partial a}{\partial \theta_i}(\theta) = E_{\theta'} \frac{\partial \tilde{g}}{\partial \theta_i}(\theta) \tag{7}$$

where

$$\frac{\partial \tilde{g}}{\partial \theta_i}(\theta) = \frac{\partial g}{\partial \theta_i}(\theta, \mathbf{X}_0, \ldots, \mathbf{X}_T)\, L_T(\theta, \theta') + g(\theta, \mathbf{X}_0, \ldots, \mathbf{X}_T) \frac{\partial}{\partial \theta_i} L_T(\theta, \theta') \tag{8}$$

and

$$\begin{aligned} \frac{\partial}{\partial \theta_i} L_T(\theta, \theta') &= \frac{\partial \mu}{\partial \theta_i}(\theta, \mathbf{X}_0) \frac{L_T(\theta, \theta')}{\mu(\theta, \mathbf{X}_0)} \\ &+ \sum_{j=0}^{T-1} \frac{\partial P(\theta, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i} \frac{L_T(\theta, \theta')}{P(\theta, \mathbf{X}_j, \mathbf{X}_{j+1})} \end{aligned} \tag{9}$$

Most often one takes $\theta' = \theta$.

When $\{\mathbf{X}_n\}_{n \geq 0}$ has a regenerative structure, the above derivation can also be extended to the case where $a(\theta)$ is described by the ratio formula of regenerative analysis [12], and T coincides with regeneration epochs.

Clearly, assuming a unique local minimum, deterministic gradient descent algorithms can have guaranteed convergence to the minimizing point. It is shown in [11] that, assuming a unique minimum, a *stochastic* descent algorithm of the Robbins-Monro type [17, 11],

$$\theta_{n+1} = \theta_n - h(n) \widehat{\nabla} a(\theta_n) \tag{10}$$

that uses MC estimates of the derivatives based on eqs. (6)-(9) can also be guaranteed convergence, for the appropriate selection of step size $h(n)$.

3.2 The SGD Algorithm

We observe now that the variance of the IS estimator in (2) is also an expectation parameterized by the IS settings. Recall that we wish to choose IS parameter values in a way that

minimizes this IS-variance. Let Z be equal to the IS estimate in (2). Therefore, by letting $a(\theta) = E_{P^*}\{(Z - E_{P^*}(Z))^2\}$, where θ is the vector of IS parameters in (2) that determine P^*, we can formulate the choice of IS parameter values as a minimization problem (i.e., $\min_\theta a(\theta)$) that can be tackled according to (10).

Such an approach to optimizing the choice of IS settings is a natural complement to our previous statistical optimization techniques, where we determine near-optimal IS values by observing estimates of the cost function, namely the IS-variance. Its greatest potential advantage is that, by exploiting more prior knowledge and information about the problem at hand (i.e., derivative information), it can potentially zero-in on the optimal IS settings faster than global search techniques like the MFA and similar annealing methods. The essential difference from annealing techniques is that gradient-based techniques are local optimization methods.

Another factor in overall efficiency is the choice of the appropriate step size $h(n)$. An $h(n)$ small enough to guarantee almost sure (a.s.) convergence can lead to impractically long run times, while an $h(n)$ that is too large can lead to divergence. Clearly, a trial-and-error procedure is required to establish the best trade-off choice.

For a given point θ in the parameter space the random numbers used to estimate $\nabla a(\theta)$ are drawn using $\theta' = \theta$. Therefore, during the search the simulation sampling distribution is continuously changing while approaching the optimal IS distribution as $\theta \rightarrow \theta_{opt}$. Thus, the algorithm tends to constantly improve the IS-variance until the near-optimal is found. Our Stochastic Gradient Descent (SGD) algorithm is outlined in Fig. 1.

3.3 The SFA Algorithm

A necessary condition for speed-up when choosing IS settings is that the raw frequency of important events has to be significantly increased with respect to the original sampling distribution before any speed-up factor is realized. Thus, when choosing an IS sampling distribution a primary concern must be to increase the frequency of important events (FIE).

It has been a common heuristic assumption among practitioners of IS simulation that the IS parameters have to be modified so that the effect on the FIE is maximized. This

$n \Leftarrow 0$ /* Initialize iteration count */
$\boldsymbol{\theta}_0 \Leftarrow \boldsymbol{\theta}_{start}$ /* Initialize IS parameter values */
/* Perform gradient descent on estimated IS-variance */
/* until empirical precision α is sufficient */
do {

Calculate $h(n)$ /* Get new step size */
$\boldsymbol{\theta}_{n+1} \Leftarrow \boldsymbol{\theta}_n - h(n)\,\widehat{\nabla} var(\boldsymbol{\theta}_n)$ /* Perform descent step */
Calculate $\widehat{var}(\boldsymbol{\theta}_{n+1})$ using N_A RC's
Calculate $\hat{P}_{n+1}$ using N_A RC's

} while { $\sqrt{\widehat{var}(\boldsymbol{\theta}_{n+1})}/\hat{P}_{n+1} > \alpha$ }

Figure 1: Pseudo-code describing the SGD algorithm.

suggests biasing in the "direction of steepest ascent in FIE", since this direction maximizes the increase in FIE for the same "total" modification of the parameter values. More precisely: Let $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ such that no IS biasing is applied and then iteratively set

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n + h'(n)\widehat{\nabla} P_{IE}(\boldsymbol{\theta}_n) \tag{11}$$

where $P_{IE}(\boldsymbol{\theta}_n)$ denotes the probability of the important event as a function of $\boldsymbol{\theta}$, until $P_{IE}(\boldsymbol{\theta}_n)$ saturates, i.e., attains its maximum value. Then pick $\boldsymbol{\theta}_{op}$ from the above "search path" such that the IS-variance is minimum (or near-minimum). Clearly, the same likelihood ratio estimation techniques previously discussed can be applied to yield efficient and accurate estimates of the partial derivatives required.

The effectiveness of this directed search procedure is supported by the fact that the true (or asymptotical) global minimum of the IS variance lies on the trajectory (11) for some interesting cases, namely the case of detection errors in a linear filter under additive white Gaussian noise (AWGN) [18], and the case of cell loss probability in a M/M/1/K queue, as both theory and empirical results indicate. Empirical results also indicate that this is true for the case of detection errors in mildly nonlinear filters under AWGN (as those in [3]). For an M/M/1/K queue the probability that k customers are in the system is given by

$$p_k = \begin{cases} \frac{1-\lambda/\mu}{1-(\lambda/\mu)^{K+1}}\,(\lambda/\mu)^k & , \; 0 \le k \le \mathrm{K} \\ 0 & , \; \text{otherwise} \end{cases}$$

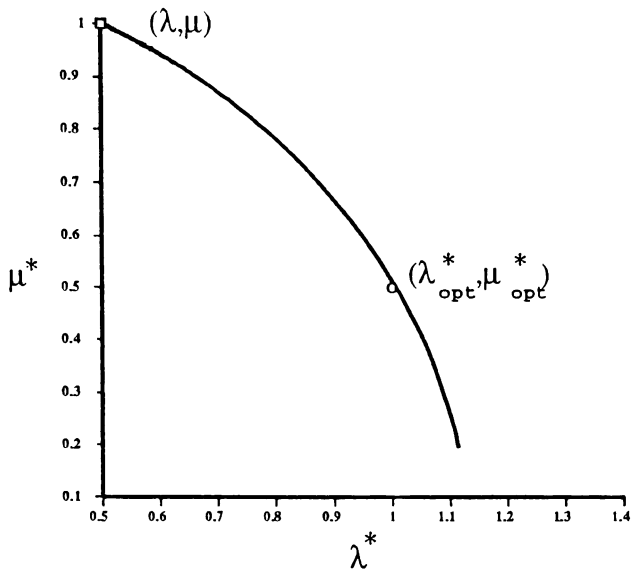

Figure 2: Example of steepest ascent trajectory for an M/M/1/K queueing system, with $\lambda = 0.5$, $\mu = 1.0$ and K = 10. The optimal IS point is $\lambda^*_{opt} = 1.0$ and $\mu^*_{opt} = 0.5$.

where λ is the arrival rate and μ the service rate. The probability that a customer will be lost is equal to $p_{\rm K}$. Let $\rho = \lambda/\mu$. Under IS

$$\frac{\partial p^*_{\rm K}}{\partial \lambda^*} = \frac{\partial p^*_{\rm K}}{\partial \rho^*}\frac{\partial \rho^*}{\partial \lambda^*} = \frac{\partial p^*_{\rm K}}{\partial \rho^*}\left(\frac{1}{\mu^*}\right)$$

and

$$\frac{\partial p^*_{\rm K}}{\partial \mu^*} = \frac{\partial p^*_{\rm K}}{\partial \rho^*}\frac{\partial \rho^*}{\partial \mu^*} = \frac{\partial p^*_{\rm K}}{\partial \rho^*}\left(-\frac{\lambda^*}{\mu^{*2}}\right)$$

where the asterisk $(*)$ denotes the biased quantities, and

$$\begin{aligned}\frac{\partial p^*_{\rm K}}{\partial \rho^*} &= \rho^{*{\rm K}}\frac{-(1-\rho^{*{\rm K}+1}) + (1-\rho^*)({\rm K}+1)\rho^{*{\rm K}}}{(1-\rho^{*{\rm K}+1})^2} \\ &\quad + \frac{1-\rho^*}{1-\rho^{*{\rm K}+1}}{\rm K}\rho^{*{\rm K}-1}\end{aligned}$$

Define the *steepest ascent trajectory*, $\mathcal{S}$, as the path that starts from the point (λ, μ) (the unbiased operating point of the queueing system) and follows the gradient of $p^*_{\rm K}$, in the space of λ^* and μ^*. Such a trajectory can be defined iteratively from

$$\lambda^*_{l+1} = \lambda^*_l + \left.\frac{\partial p^*_{\rm K}}{\partial \lambda^*}\right|_{\lambda^*=\lambda^*_l}\Delta\lambda^*$$

$$
\begin{aligned}
&= \lambda_l^* + \left.\frac{\partial p_{\mathrm{K}}^*}{\partial \rho^*}\right|_{\rho^*=\rho_l^*} \left(\frac{1}{\mu_l^*}\right) \Delta\lambda^* \\
\mu_{l+1}^* &= \mu_l^* + \left.\frac{\partial p_{\mathrm{K}}^*}{\partial \mu^*}\right|_{\mu^*=\mu_l^*} \Delta\mu^* \\
&= \mu_l^* + \left.\frac{\partial p_{\mathrm{K}}^*}{\partial \rho^*}\right|_{\rho^*=\rho_l^*} \left(-\frac{\lambda_l^*}{\mu_l^{*2}}\right) \Delta\mu^* \\
\rho_l^* &= \frac{\lambda_l^*}{\mu_l^*}
\end{aligned}
$$

for $l = 0, \ldots, l_{max} - 1$, with $\lambda_0^* = \lambda$, $\mu_0^* = \mu$, and $\Delta\lambda^*$, $\Delta\mu^*$ are sufficiently small.

An example of a steepest ascent trajectory is shown in Figure 2, for $\lambda = 0.5$, $\mu = 1.0$, and $\mathrm{K} = 10$. Here, $\Delta\lambda^* = \Delta\mu^* = 0.035$ and $l_{max} = 200$. For the M/M/1/K queue, the optimal (and unique asymptotically efficient) IS operating point is $\lambda_{opt}^* = \mu$, $\mu_{opt}^* = \lambda$, similar to what was shown in [1] for the M/M/1/∞ queue. For this example (and every other combination of λ and μ we tried) the trajectory $\mathcal{S}$ did include the optimal IS point.

Furthermore, in support of this search path, we observe that the IS-variance can be written as

$$E_{\boldsymbol{\theta}_0}\{1_{IE} L^*(\boldsymbol{\theta})\}$$

while the FIE can be written as

$$E_{\boldsymbol{\theta}_0}\{1_{IE} / L^*(\boldsymbol{\theta})\}$$

where 1_{IE} is the indicator function of an important event, and $L^*(\boldsymbol{\theta})$ is the cumulative weight of (2). This indicates that increasing the FIE should tend to decrease the IS-variance. More specifically, at $\boldsymbol{\theta} = \boldsymbol{\theta}_{BF}$ (i.e., at the brute-force MC point) $\nabla P_{IE}(\boldsymbol{\theta})$ is *parallel* to $\nabla \sigma_{IS}^2(\boldsymbol{\theta})$, which means that, for a sufficiently small step size $h'(n)$, the first step away from the brute-force MC point will necessarily lead to a decrease in the IS-variance. Since this alternative algorithm picks the IS parameter values with the minimum variance over the trajectory (11), the IS settings chosen have a guaranteed speed-up over the conventional MC estimator.

It is reasonable to assume that the $\widehat{\nabla} P_{IE}$ estimator will be more accurate for the same sample size than the $\widehat{\nabla}\sigma_{IS}^2$ estimator ($\widehat{\nabla}\sigma_{IS}^2$ involves estimates of *second order* moments). Furthermore, in all practical cases that we considered, the FIE was monotonic with respect

to the biasing parameter values. Also, there are no convergence effects to slow SFA down, since the gradients guiding the search are not the gradients of the cost function. For these reasons, following trajectory (11) is rather general and robust, regardless of the existence of local minima in the IS-variance space. Finally, searching along trajectory (11) is guaranteed to provide speed-up (potentially equivalent to that provided by the SGD algorithm).

Partials $\frac{\partial P_{IE}}{\partial \theta_i}$ can be found by replacing $a(\boldsymbol{\theta})$ with P_{IE} in eq. (7), where $g(\boldsymbol{\theta}, \mathbf{X}_0, \ldots, \mathbf{X}_T)$ is the number of blocked cells in a RC. Hence $g(\boldsymbol{\theta}, \mathbf{X}_0, \ldots, \mathbf{X}_T) = \sum_{j=0}^{T} h(\boldsymbol{\theta}, \mathbf{X}_j)$, where

$$h(\boldsymbol{\theta}, \mathbf{X}_j) = \begin{cases} 1 & \text{if a cell is blocked during slot } j \\ 0 & \text{otherwise} \end{cases}$$

Choosing the same initial distributions $\mu(\boldsymbol{\theta}, \mathbf{X}_0) = \mu(\boldsymbol{\theta}', \mathbf{X}_0)$ and $\boldsymbol{\theta}' = \boldsymbol{\theta}$, and choosing T, the end of a RC, as the stopping time, it follows that $\frac{\partial g}{\partial \theta} = \sum_{j=0}^{T} \frac{\partial h}{\partial \theta} = \sum_{j=0}^{T} 0 = 0$, and

$$\frac{\partial P_{IE}}{\partial \theta_i} = E_{\boldsymbol{\theta}} \left[g(\boldsymbol{\theta}, \mathbf{X}_0, \ldots, \mathbf{X}_T) \sum_{j=0}^{T-1} \frac{\partial P(\boldsymbol{\theta}, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i} \frac{1}{P(\boldsymbol{\theta}, \mathbf{X}_j, \mathbf{X}_{j+1})} \right] \tag{12}$$

Finally, in order to estimate the partials of $P_{IE}(\boldsymbol{\theta})$ with respect to the elements of $\boldsymbol{\theta}$, one can draw numbers using $\boldsymbol{\theta}$, and then take a sample mean over i.i.d. random repetitions of the quantity in brackets on the right-hand-side of eq. (12). These partial derivatives can then be used in the SFA algorithm.

When important events are rare the accuracy of $\widehat{\nabla} P_{IE}(\boldsymbol{\theta}_n)$ will be very poor as long as P_{IE} is low. In [13], IS is applied to $\widehat{\nabla} P_{IE}(\boldsymbol{\theta}_n)$ as follows: Let $a(\boldsymbol{\theta}_n) = P_{IE}(\boldsymbol{\theta}_n)$. Then

$$\nabla a(\boldsymbol{\theta}_n) = E_{\boldsymbol{\theta}''} \nabla \tilde{g}(\boldsymbol{\theta}_n) L_T(\boldsymbol{\theta}', \boldsymbol{\theta}'') \tag{13}$$

where $\boldsymbol{\theta}''$ is chosen in order to increase the accuracy of the estimation. Therefore, we can use $\boldsymbol{\theta}''$ for the sampling distribution to estimate $\nabla P_{IE}(\boldsymbol{\theta}_n)$ until P_{IE} becomes large enough to allow us to start using $\boldsymbol{\theta}_n$ itself. We call this additional biasing "second-order IS". Heuristic arguments similar to those we present later for choosing a starting point for the SGD algorithm can be used for the selection of a favorable "second-order-IS" for the SFA algorithm.

Our Stochastic (Important Event) Frequency Ascent (SFA) algorithm is given in Fig. 3. The outline in Fig. 3 is intended to be general and does not address issues such as search

$n \Leftarrow 0$ /* Initialize iteration count */
$\theta_0 \Leftarrow \theta_{BF}$ /* Initialize IS parameter values */
/* Perform gradient ascent on estimated "raw" */
/* probability $\hat{P}_{IE}(\theta_n)$ until saturation occurs */
do {

Calculate $h(n)$ /* Get new step size */
$\theta_{n+1} \Leftarrow \theta_n + h(n)\,\hat{\nabla} P_{IE}(\theta_n)$ /* Perform ascent step */
Calculate $\widehat{var}(\theta_{n+1})$ using N_B RC's
Calculate $\hat{P}_{IE}(\theta_{n+1})$ using N_B RC's

} while { $\hat{P}_{IE}(\theta_{n+1}) - \hat{P}_{IE}(\theta_n) > \epsilon$ }
Return n, θ_n such that $\widehat{var}(\theta_n)$ is minimum

Figure 3: Pseudo-code describing the SFA algorithm.

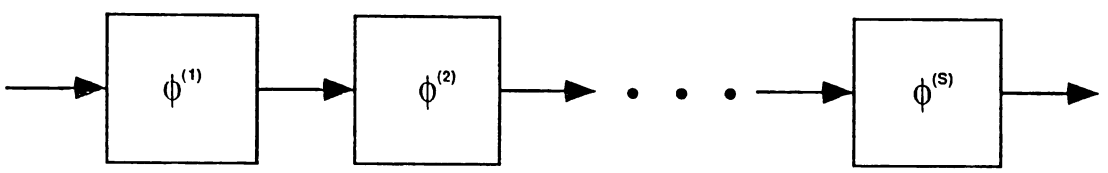

Figure 4: General tandem queueing network.

resolution and statistical accuracy. In [6] we have described a detailed algorithmic procedure for near-minimization the IS-variance based on statistical estimates taken on an *optimal search direction*, when simulating digital communication links with linear receivers. In the case of [6] the optimal direction was the direction of the impulse response of the linear receiver filter. The SFA approach essentially generalizes that algorithm by providing a (heuristic) favorable search direction for a much larger class of systems.

4 Stochastic Gradients for Tandem Networks

4.1 Likelihood Ratio Techniques for Tandem Networks

As shown in Figure 4, a general tandem queueing network consists of several stages of queues, the output of one queue feeding the input of the next queue. Assuming a discrete-time or slotted approach, at least one time slot is need for a cell or packet to propagate through a single stage of the queueing network.

Let $\mathbf{V}_i^{(s)}$ be the vector of observations representing the state of the arrival processes and the queue for stage s of S in the tandem network at time i. Let the vector of observations relevant to the tandem network at stage S at time i be $\mathbf{X}_i = \left(\mathbf{V}_{i-S+1}^{(1)}, \mathbf{V}_{i-S+2}^{(2)}, \ldots, \mathbf{V}_i^{(S)}\right)$. The time shift in the observations at each stage results from a cell requiring at least one time slot to propagate through each stage. Now $\{\mathbf{X}_i\}_{i\geq 0}$ is Markovian under general conditions. Let $\theta_l^{(s)}$ be the parameter associated with the l-th random process at stage s for $l = 1, \ldots, m_s$, and $\phi^{(s)} = \left(\theta_1^{(s)}, \theta_2^{(s)}, \ldots, \theta_{m_s}^{(s)}\right)$ be the vector of parameters belonging to stage s so that $\theta^{(S)} = \left(\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(S)}\right)$ is the overall vector of parameters.

An expectation $a^{(S)}(\theta^{(S)}) = E_{\theta^{(S)}} g(\theta^{(S)}, \mathbf{X}_0, \ldots, \mathbf{X}_T)$ at the S-th stage in the tandem network can also be written as

$$a^{(S)}(\theta^{(S)}) = E_{\theta'^{(S)}} g(\theta^{(S)}, \mathbf{X}_0, \ldots, \mathbf{X}_T) L_T^{(S)}(\theta^{(S)}, \theta'^{(S)}, \mathbf{X}_0, \ldots, \mathbf{X}_T) \tag{14}$$

where T is a finite stopping time. The expectation $a^{(S)}(\theta^{(S)})$ is taken at the S-th stage in the network, otherwise S stages would not be needed in the simulation. We assume that the optimization of the bias parameters is performed for only a single expectation which is at the S-th stage in the network, since simultaneous optimization of multiple estimators could lead to conflicting bias parameter settings that would not result in performance speed-up.

At stage S in the tandem network, the likelihood ratio at time n during the simulation, $L_n^{(S)}(\theta^{(S)}, \theta'^{(S)}, \mathbf{X}_0, \ldots, \mathbf{X}_n)$, depends on all random transitions which previously occurred at stage S up to time n, as well as all random transitions which previously occurred at stages 1 to $S-1$ at times $n-S$ to $n-1$, respectively. As stated before, this time shift results because one time slot is needed for a cell to propagate through a single stage in the network due to the slotted time operation of the tandem queueing network, as well as the simulation. The stages at positions downstream from S in the tandem network will have no effect on the likelihood ratio $L_n^{(S)}(\theta^{(S)}, \theta'^{(S)}, \mathbf{X}_0, \ldots, \mathbf{X}_n)$ at stage S because cells flow only from stage s to stage $s+1$ in the tandem network. The random processes parameterized by the vector $\phi^{(s)}$ are independent from stage to stage, even though the expectation $a^{(S)}(\theta^{(S)})$ is dependent on each $\phi^{(s)}$. Because of the Markovian nature of $\{\mathbf{X}_n\}_{n\geq 0}$, the likelihood ratio for any instant

$n \Leftarrow 0$ /* Initialize iteration count */
$\theta_0^{(S)} \Leftarrow \theta_{start}^{(S)}$ /* Initialize IS parameter values */
/* Perform gradient descent on estimated IS-variance */
/* until empirical precision α is sufficient */
do {

Calculate $h(n)$ /* Get new step size */
$\theta_{n+1}^{(S)} \Leftarrow \theta_n^{(S)} - h(n)\,\widehat{\nabla} var^{(S)}(\theta_n^{(S)})$ /* Perform descent step */
Calculate $\widehat{var}^{(S)}(\theta_{n+1}^{(S)})$ using N_A RC's
Calculate $\hat{P}_{S,n+1}$ using N_A RC's

} while { $\sqrt{\widehat{var}^{(S)}(\theta_{n+1}(S))}/\hat{P}_{S,n+1} > \alpha$ }

Figure 5: Pseudo-code describing the SGD algorithm for tandem networks.

n at the S-th stage in the network is

$$L_n^{(S)}(\theta^{(S)}, \theta'^{(S)}, \mathrm{X}_0, \ldots, \mathrm{X}_n) = \prod_{i=0}^{n-1} \frac{P(\theta^{(S)}, \mathrm{X}_i, \mathrm{X}_{i+1})}{P(\theta'^{(S)}, \mathrm{X}_i, \mathrm{X}_{i+1})} \tag{15}$$

where $P(\theta^{(S)}, \mathrm{X}_i, \mathrm{X}_{i+1})$ is the transition probabilities under $\theta^{(S)}$. In the following discussion, we use $L_n^{(S)}(\theta^{(S)}, \theta'^{(S)})$ instead of the expression shown in (15) in order to simplify the notation.

4.2 The SGD Algorithm for Tandem Networks

The SGD algorithm for tandem queues is outlined in Fig. 5. IS is applied by replacing the original parameter $\theta_0^{(S)}$ with $\theta^{(S)}$. The performance measure of interest is the mean-square value $E_{\theta^{(S)}}(\hat{P}_S^2)$ and its gradient with respect to the biased parameters, $\nabla_{\theta^{(S)}} E_{\theta^{(S)}}(\hat{P}_S^2)$, at the S-th stage of the tandem network, where $\hat{P}_S = \sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)})$ is the estimate of the cell loss at the S-th stage of the tandem network, $I_j^{(S)}$ is the indicator function of a cell block in slot j at stage S, and

$$L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) = \prod_{i=0}^{j-1} \frac{P(\theta_0^{(S)}, \mathrm{X}_i, \mathrm{X}_{i+1})}{P(\theta^{(S)}, \mathrm{X}_i, \mathrm{X}_{i+1})} \tag{16}$$

Then,

$$\frac{\partial}{\partial \theta_i^{(s)}} E_{\theta^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \right]$$

$$= \frac{\partial}{\partial \theta_i^{(s)}} E_{\theta'^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 L_T^{(S)}(\theta^{(S)}, \theta'^{(S)}) \right]$$

$$= E_{\theta'^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \frac{\partial L_T^{(S)}}{\partial \theta_i^{(s)}}(\theta^{(S)}, \theta'^{(S)}) \right.$$

$$\left. + \left\{ \frac{\partial}{\partial \theta_i^{(s)}} \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \right\} L_T^{(S)}(\theta^{(S)}, \theta'^{(S)}) \right] \tag{17}$$

where $\theta'(S)$ parameterizes the actual sampling distribution for the S-th stage of the tandem network. Since

$$\frac{\partial L_T^{(S)}}{\partial \theta_i^{(s)}}(\theta^{(S)}, \theta'^{(S)}) = \sum_{j=0}^{T-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i^{(s)}} \frac{L_T^{(S)}(\theta^{(S)}, \theta'^{(S)})}{P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})} \tag{18}$$

this results in

$$\frac{\partial}{\partial \theta_i^{(s)}} E_{\theta^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \right]$$

$$= E_{\theta'^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \sum_{j=0}^{T-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i^{(s)}} \frac{L_T^{(S)}(\theta^{(S)}, \theta'^{(S)})}{P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})} \right.$$

$$\left. + \left\{ \frac{\partial}{\partial \theta_i^{(s)}} \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \right\} L_T^{(S)}(\theta^{(S)}, \theta'^{(S)}) \right] \tag{19}$$

Using $L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) = 1/L_j^{(S)}(\theta^{(S)}, \theta_0^{(S)})$, to show that

$$\frac{\partial L_j^{(S)}}{\partial \theta_i^{(s)}}(\theta_0^{(S)}, \theta^{(S)}) = -\sum_{k=0}^{j-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})}{\partial \theta_i^{(s)}} \frac{L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)})}{P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})} \tag{20}$$

Then, from

$$\frac{\partial}{\partial \theta_i^{(s)}} \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{S}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 = 2 \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{S}(\theta_0^{(S)}, \theta^{(S)}) \right) \sum_{j=0}^{T-1} I_j^{(S)} \frac{\partial L_j^{(S)}}{\partial \theta_i^{(s)}}(\theta_0^{(S)}, \theta^{(S)})$$

$$= 2 \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right) \sum_{j=0}^{T-1} I_j^{(S)}$$

$$\bullet \left[-\sum_{k=0}^{j-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})}{\partial \theta_i^{(s)}} \frac{L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)})}{P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})} \right] \tag{21}$$

and taking $\boldsymbol{\theta}'^{(S)} = \boldsymbol{\theta}^{(S)}$ (i.e., drawing random numbers using $\boldsymbol{\theta}^{(S)}$) which implies $L_T^{(S)}(\boldsymbol{\theta}^{(S)}, \boldsymbol{\theta}'^{(S)}) = 1$, results in

$$\frac{\partial}{\partial \theta_i} E_{\boldsymbol{\theta}^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\boldsymbol{\theta}_0^{(S)}, \boldsymbol{\theta}^{(S)}) \right)^2 \right]$$

$$= E_{\boldsymbol{\theta}^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\boldsymbol{\theta}_0^{(S)}, \boldsymbol{\theta}^{(S)}) \right)^2 \sum_{j=0}^{T-1} \frac{\partial P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i^{(s)}} \frac{1}{P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})} \right.$$

$$\left. - 2 \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\boldsymbol{\theta}_0^{(S)}, \boldsymbol{\theta}^{(S)}) \right) \sum_{j=0}^{T-1} I_j^{(S)} \sum_{k=0}^{j-1} \frac{\partial P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})}{\partial \theta_i^{(s)}} \frac{L_j^{(S)}(\boldsymbol{\theta}_0^{(S)}, \boldsymbol{\theta}^{(S)})}{P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})} \right] \quad (22)$$

Thus, in order to estimate the partials of the IS mean-square term with respect to the elements of $\boldsymbol{\theta}^{(S)}$, numbers should be draw using $\boldsymbol{\theta}^{(S)}$, and then a sample mean taken over i.i.d. random repetitions of the quantity in brackets on the right-hand-side of (22). These partial derivatives can then be used in the SGD algorithm.

The task of obtaining partial derivatives of $P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})$ is facilitated, among others, by the multiplicative nature of the one-step transition probabilities (due to independence of the random choices involved). For the random processes considered here, a transition in a slot depends on all independent (discrete) random events, each occurring with conditional probability $p_i^{(s)}$ under the original, unbiased settings. Under IS, these probabilities can be biased so that they become $\theta_l^{(s)} p_l^{(s)}$. Then $\boldsymbol{\theta}^{(S)} = (\theta_1^{(1)}, \ldots, \theta_{m_S}^{(S)})$ and $P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1}) = \prod_{s=1}^{S} \prod_{l=1}^{m_s} \theta_l^{(s)} p_l^{(s)}$, leading to

$$\frac{\partial P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})}{\partial \theta_l^{(s)}} = p_l^{(s)} \prod_{j=1}^{S} \prod_{k=1 (k \neq l \text{ and } j \neq s)}^{m_j} \theta_k^{(j)} p_k^{(j)} \quad (23)$$

and

$$\frac{1}{P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})} \frac{\partial P(\boldsymbol{\theta}^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})}{\partial \theta_l^{(s)}} = \frac{1}{\theta_l^{(s)}} \quad (24)$$

Several heuristic arguments can be used to identify a starting point for the search in (10) when important events are rare. For example, near-optimal IS settings for a single or tandem queue case where the important events are not rare (e.g., smaller buffer size for cell loss probability) can be found first and then extrapolated to obtain a starting point for the

and taking $\theta'^{(S)} = \theta^{(S)}$ (i.e., drawing random numbers using $\theta^{(S)}$) which implies $L_T^{(S)}(\theta^{(S)}, \theta'^{(S)}) = 1$, results in

$$\frac{\partial}{\partial \theta_i} E_{\theta^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \right]$$

$$= E_{\theta^{(S)}} \left[\left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right)^2 \sum_{j=0}^{T-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})}{\partial \theta_i^{(s)}} \frac{1}{P(\theta^{(S)}, \mathbf{X}_j, \mathbf{X}_{j+1})} \right.$$

$$\left. - 2 \left(\sum_{j=0}^{T-1} I_j^{(S)} L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)}) \right) \sum_{j=0}^{T-1} I_j^{(S)} \sum_{k=0}^{j-1} \frac{\partial P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})}{\partial \theta_i^{(s)}} \frac{L_j^{(S)}(\theta_0^{(S)}, \theta^{(S)})}{P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})} \right] \quad (22)$$

Thus, in order to estimate the partials of the IS mean-square term with respect to the elements of $\theta^{(S)}$, numbers should be draw using $\theta^{(S)}$, and then a sample mean taken over i.i.d. random repetitions of the quantity in brackets on the right-hand-side of (22). These partial derivatives can then be used in the SGD algorithm.

The task of obtaining partial derivatives of $P(\theta^{(S)}, \mathbf{X}_k, \mathbf{X}_{k+1})$ is facilitated, among others, by the multiplicative nature of the one-step transition probabilities (due to independence of the random choices involved). For the random processes considered here, a transition in a slot depends on all independent (discrete) random events, each occurring with conditional probability $p_i^{(s)}$ under the original, unbiased settings. Under IS, these probabilities can be biased so that they become $\theta_l^{(s)} p_l^{(s)}$. Then $\theta^{(S)} = (\theta_1^{(1)}, \ldots, \theta_{m_S}^{(S)})$ and $P(\theta^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1}) = \prod_{s=1}^{S} \prod_{l=1}^{m_s} \theta_l^{(s)} p_l^{(s)}$, leading to

$$\frac{\partial P(\theta^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})}{\partial \theta_l^{(s)}} = p_l^{(s)} \prod_{j=1}^{S} \prod_{k=1 (k \neq l \text{ and } j \neq s)}^{m_j} \theta_k^{(j)} p_k^{(j)} \quad (23)$$

and

$$\frac{1}{P(\theta^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})} \frac{\partial P(\theta^{(S)}, \mathbf{X}_i, \mathbf{X}_{i+1})}{\partial \theta_l^{(s)}} = \frac{1}{\theta_l^{(s)}} \quad (24)$$

Several heuristic arguments can be used to identify a starting point for the search in (10) when important events are rare. For example, near-optimal IS settings for a single or tandem queue case where the important events are not rare (e.g., smaller buffer size for cell loss probability) can be found first and then extrapolated to obtain a starting point for the

rare event case. We will also demonstrate that the near-optimal bias parameters for a S-stage tandem queue network can be used as a starting point to quickly obtain the near-optimal bias parameters for a $(S+1)$-stage tandem queue network.

5 Experimental Examples

5.1 The IBP/Geo/1/K Queue

The IBP/Geo/1/K queue [1] is the slotted-time counterpart of the IPP/M/1/K queue [2]. For this queue, although the service process is memoryless, the arrival process is *bursty*, making this system a useful and widely used model for the bursty processes involved in B-ISDN and ATM analyses. Exact solutions for this queue can be obtained by numerical solution of the corresponding Markov chain. We include it in our experiments to provide further validation of our techniques, as was also done in [7].

There are two states of the arrival process: active and idle. In the active state, an arrival can occur with probability α while in the idle state no arrivals can occur. While the arrival process is in the active state, there is a probability p at each slot that the state will remain active and a probability $1-p$ that it will change to idle. While the arrival process is in the idle state, there is a probability q at each slot that the state will remain idle and a probability $1-q$ that it will change to active. When the server is busy, there is a probability $1-\sigma$ in each slot that a customer will depart. Arrivals and service completions are independent. There is a finite capacity of K customers in the system. In our experiments, α was assumed to be equal to 1. The squared coefficient of variation C^2 of the interarrival times is used to measure the burstiness of the arrival process. Typical values are $C^2 = 1$ corresponding to Poisson arrivals, $C^2 \approx 20$ for voice and C^2 ranging from 10 to 10,000 for video. A numerical technique that evaluates cell loss probabilities for this queueing system can be found in [19].

Under regenerative IS, we choose the times that a customer arrives to an empty system and the arrival process has just changed to active, as the regeneration points. In each

[1] IBP stands for Interrupted Bernoulli process, Geo stands for Geometric

[2] IPP stands for Interrupted Poisson process

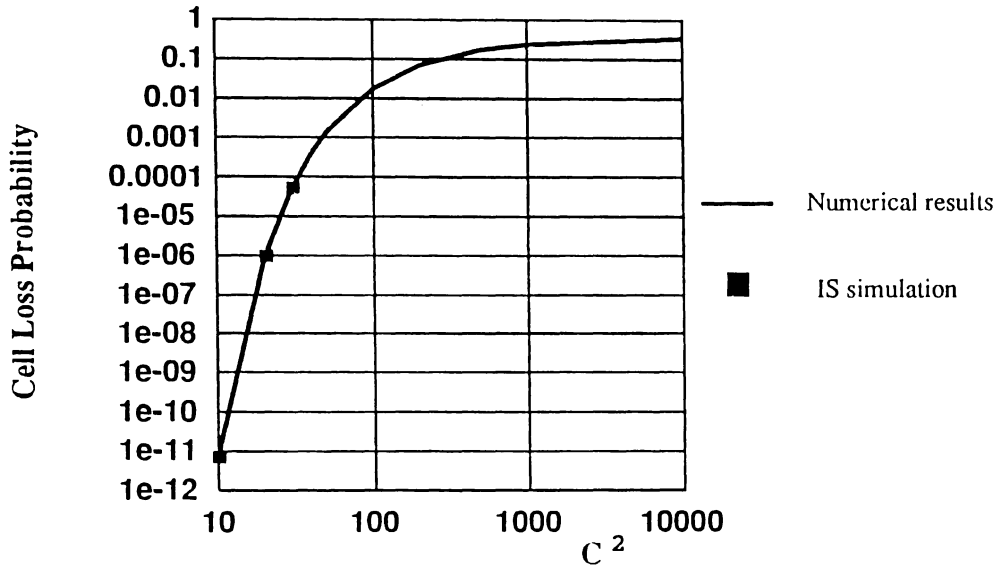

Figure 6: Estimated cell loss probabilities and numerically calculated probabilities [19] for the IBP/Geo/1/K queue.

C^2	CPU Time, seconds
10.0	0.7315
20.0	1.417
30.0	2.099

Table 1: CPU Time for 1,000 RC's of the IBP/Geo/1/K queue on a DECStation 5000/25 when no IS is applied (K=200).

regeneration cycle (RC), we bias initially p, q and σ to p_1^*, q_1^* and σ_1^*, until one customer has been blocked, then change IS parameters to p_2^*, q_2^* and σ_2^* in order to allow fast regeneration.

In our experiments, we set $p_2^* = p$, $q_2^* = q$, $\sigma_2^* = \sigma$, and optimized with respect to the settings of $\theta_1 = p_1^*/p$, $\theta_2 = q_1^*/q$ and $\theta_3 = \sigma_1^*/\sigma$ using the SGD and the SFA algorithms from above. Results were obtained for queue set-ups that corresponded to three different values of C^2, namely 10.0, 20.0, and 30.0 (see Table 2). Estimated loss probabilities are in agreement with the numerically calculated probabilities in [19], as illustrated in Figure 6. The simulation time required for 1,000 RC's on a DECStation 5000/25 when no IS was applied for these three cases is given in Table 1. The queue capacity K was set equal to 200.

In applying the SGD algorithm we used in each case the near-optimal biasing for K = 50 as a starting point. Obtaining the near-optimal IS biasing for K = 50 was not difficult, since

System	Pr[loss]	$\theta_{op1}, \theta_{op2}, \theta_{op3}$	$\widehat{\Pr}$[loss]	95% Interval	R_{net}
IBP/Geo/1/K $C^2 = 10.0$, $\sigma = 0.35147$, K = 200 $p = 0.932075471$, $q = 0.954716981$	7.530×10^{-12}	1.0457 0.9244 1.0763	7.536×10^{-12}	$(7.405 \times 10^{-12}$, $7.666 \times 10^{-12})$	1.3×10^{8}
IBP/Geo/1/K $C^2 = 20.0$, $\sigma = 0.35147$, K = 200 $p = 0.965048543$, $q = 0.976699029$	8.301×10^{-7}	1.0231 0.9586 1.0414	8.259×10^{-7}	$(8.135 \times 10^{-7}$, $8.382 \times 10^{-7})$	1.4×10^{3}
IBP/Geo/1/K $C^2 = 30.0$, $\sigma = 0.35147$, K = 200 $p = 0.976470588$, $q = 0.984313725$	4.829×10^{-5}	1.0156 0.9710 1.0231	4.809×10^{-5}	$(4.730 \times 10^{-5}$, $4.887 \times 10^{-5})$	2.1×10

Table 2: Estimated cell loss probabilities and speed-up factors using the SGD algorithm for the IBP/Geo/1/K queue. For these estimates: $N_R = 100$, $N_{RC} = 300$.

the corresponding loss probabilities were high and the space could be searched efficiently with the SGD algorithm starting from the brute-force MC point. Furthermore, we used $N_A = 300$ RC's per simulation run for $C^2 = 20$ and $C^2 = 30$, and $N_A = 3,000$ for $C^2 = 10$, The algorithm converged in all cases after $I_A < 1,000$ iterations. The step size h was obtained by trial-and-error and varied from 5×10^{-3} ($C^2 = 30$) to 10^{13} ($C^2 = 10$).

In applying the SFA algorithm we used in each case the near-optimal biasing for K = 50 as the "second-order" IS. Obtaining the near-optimal IS biasing for K = 50 was not difficult, since the corresponding loss probabilities were high and the space could be searched efficiently with the SFA algorithm without any "second-order" IS. Furthermore, we used $N_B = 300$ RC's per simulation run for $C^2 = 30$, and $N_B = 3,000$ for $C^2 = 10$ and $C^2 = 20$, The algorithm required between $I_B = 300$ and $I_B = 800$ to "scan" the search space. The step size $h = 10^{-4}$ was obtained by trial-and-error. The saturation tolerance ϵ was set to 0.05.

Tables 2 and 3 summarize the results, including the near-optimal IS biasing parameter values $(\theta_{op1}, \theta_{op2}, \theta_{op3})$ found by the SGD and the SFA algorithms, respectively, the corresponding estimated loss probabilities, the estimated confidence intervals and the speed-up factors with respect to conventional MC simulation. Numerically evaluated loss probabilities were taken from [19]. In order to determine confidence intervals and speed-up factors, N_R of N_{RC} RC's each were run using the chosen IS biasing values. As in [7], speed-up factors were obtained assuming consecutive cell losses are independent within each RC for conventional MC simulation. Furthermore, RC's were assumed to correspond to a constant number of arrivals equal to the estimated average number. This is a conservative assumption, since

System	Pr[loss]	$\theta_{op1}, \theta_{op2}, \theta_{op3}$	$\widehat{\Pr[\text{loss}]}$	95% Interval	R_{net}
IBP/Geo/1/K $C^2 = 10.0$, $\sigma = 0.35147$, K = 200 $p = 0.932075471$, $q = 0.954716981$	7.530×10^{-12}	1.0444 0.9793 1.0026	8.136×10^{-12}	$(5.123 \times 10^{-12}$, $1.114 \times 10^{-11})$	4.4×10^5
IBP/Geo/1/K $C^2 = 20.0$, $\sigma = 0.35147$, K = 200 $p = 0.965048543$, $q = 0.976699029$	8.301×10^{-7}	1.0244 0.9883 1.0008	7.807×10^{-7}	$(7.243 \times 10^{-7}$, $8.372 \times 10^{-7})$	8.3×10
IBP/Geo/1/K $C^2 = 30.0$, $\sigma = 0.35147$, K = 200 $p = 0.976470588$, $q = 0.984313725$	4.829×10^{-5}	1.0179 0.9925 1.0004	5.082×10^{-5}	$(4.530 \times 10^{-5}$, $5.633 \times 10^{-5})$	MC*

Table 3: Estimated cell loss probabilities and speed-up factors using the SFA algorithm for the IBP/Geo/1/K queue. For these estimates: $N_R = 100$, $N_{RC} = 300$. The asterisk (*) is used to denote points where the use of IS did not result in speed-up over MC simulation, hence the point used is that found by MC simulation.

when important events are bursty more such events would have to be observed for the same statistical accuracy (see [20]). The net run time speed-up over conventional MC simulation is denoted by R_{net} and takes into account the increase in the length of RC's when IS is used.

The speed-up factors given here describe the factor by which an IS estimator that uses our chosen parameter values is more accurate than a conventional MC estimator based on the same sample size. The computer run time required to search for these favorable IS values has not been included in this calculation. For the examples shown here, that overhead would reduce the overall speed-up factor by up to 2 orders of magnitude for some cases.

As expected, the estimated speed-up factors are low for high loss probabilities but increase consistently as the loss probability decreases. This is a desirable effect since increasing speed-up factors are crucial in order to estimate very low probabilities within a realistic amount of run time. Taking into account this trend, as well as the overhead involved in the search for near-optimal IS settings, one can determine in each case the break-even loss probability, below which employing IS is favorable in terms of total run time required. Our results clearly indicate that, for realistically low loss probabilities ($\geq 10^{-7}$), the statistically optimized IS settings yield significant speed-up factors over MC simulation. Furthermore, near-optimal IS parameter values are consistent with those found by MFA in [7].

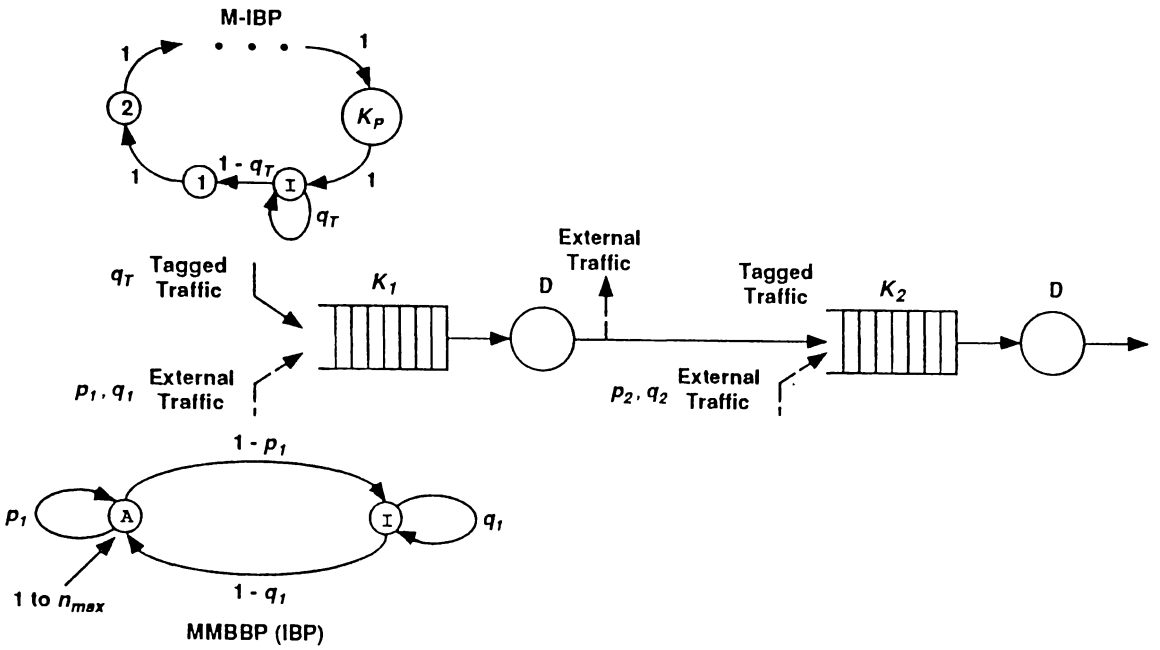

Figure 7: M-IBP+MMBBP/D/1/K tandem queues.

5.2 M-IBP+MMBBP/D/1/K Tandem Queues

5.2.1 Description

As described in [21] and shown in Figure 7, a single stage of this slotted-time queueing model has one server with a deterministic service rate of one cell per slot. There are two independent traffic streams entering the first stage of the tandem M-IBP+MMBBP/D/1/K queue. The first stream, called the tagged traffic [21], is modeled by a Modified Interrupted Bernoulli Process (M-IBP), which differs from the standard IBP in that the busy periods have a deterministic, constant length equal to K_P slots, where K_P is referred to as the packet size or number of cells in a packet, and one cell is assumed to arrive in each busy slot. When the tagged traffic is idle, there are no arrivals, and there is a probability q_T that the traffic remains idle.

The second stream, called the external traffic [21], is modeled by a Markov Modulated Bernoulli Process with Batch arrivals (MMBBP). It differs from the standard MMBP (of which the IBP is a special case) in that more than one cell can arrive during a busy slot, i.e. batch arrivals. The number of cells m arriving in a busy slot is described by some distribution

$b_i(m)$ for each state $i = 1, \ldots, N_S$ of the MMBP. Assume the MMBBP has two states, active and idle, i.e. an Interrupted Bernoulli Batch Process (IBBP). When the external traffic is active, arrivals occur and there is a probability p that the external traffic remains active. In state 1, the active state, $b_1(0) = 0$ and arrivals occur with a uniform batch-size distribution, i.e., $b_1(m) = 1/n_{max}$ for $m = 1, \ldots, n_{max}$. When the external traffic is idle, there are no arrivals, and there is a probability q that the external traffic remains idle. In state 2, the idle state, $b_2(0) = 1$, and $b_2(m) = 0$ for $m = 1, \ldots, n_{max}$. When tagged and external arrivals occur in the same slot, the queue is filled randomly with tagged and external arrivals.

For tandem configurations of M-IBP+MMBBP/D/1/K queues, p and q are indexed for each stage as p_s and q_s for $s = 1, \ldots, S$. Similarly, each queue in the tandem network has a finite buffer of length K_s. The tagged traffic always continues from one node in the network to the next node in the network, while the external traffic exits the system. Thus, the input streams of the stages following the first stage of the tandem network are characterized by the tagged traffic stream exiting the previous stage and an additional MMBBP process modeling the external traffic.

We denote by C_E^2 the squared coefficient of variation or burstiness parameter of the external traffic, which describes the variability of the interarrival time of the external cells entering the network at each stage. The corresponding burstiness parameter for the tagged traffic interarrival time variability, C_s^2, is measured at the input of each stage s in the network for $s = 1, \ldots, S$.

The recent attention paid to ATM technology has made simulation of tandem networks of great interest. Tandem networks of M-IBP+MMBBP/1/D/K queues can comprise an end-to-end model of the nodes in an ATM network, where the M-IBP tagged traffic represents the stream under observation (e.g., a specific virtual circuit), and the MMBBP external traffic represents the aggregation of all the other virtual circuits through the same node. The deterministic server models the link carrying the traffic to the next node in the network.

A numerical technique that evaluates cell loss probabilities for a single stage of the M-IBP+MMBBP/D/1/K queueing system is given in [21], although the numerical stability of that technique is still under study. Since the technique in [21] involves the numerical

System	CPU Time, seconds
1	0.573
2	0.342
3	20.772

Table 4: CPU Time for 1,000 RC's of the single M-IBP+MMBBP/D/1/K queue on a DECStation 5000/25 when no IS is applied.

solution of Markov chains with dimensionality proportional to the queueing capacity, the required run time quickly becomes forbiddingly long for large buffer sizes and/or tandem networks of queues. Furthermore, problems with numerical precision and stability may arise.

In the following, we first consider single M-IBP+MMBBP/D/1/K queues, and then M-IBP+MMBBP/D/1/K queues in tandem.

5.2.2 Stochastic Gradients for the Single M-IBP+MMBBP/D/1/K Queue

We let regeneration epochs be the instants where the queue is empty, the tagged traffic stream is idle, and the external traffic stream is just going active and is generating a cell. In each RC, we bias initially p, q and q_T to p_1^*, q_1^* and $q_{T,1}^*$, until one customer is blocked, then change IS parameters to p_2^*, q_2^* and $q_{T,2}^*$ in order to allow fast regeneration.

In our experiments, we set $p_2^* = p$, $q_2^* = q$, $q_{T,2}^* = q_T$, and optimized with respect to the settings of $\theta_1 = p_1^*/p$, $\theta_2 = q_1^*/q$ and $\theta_3 = q_{T,1}^*/q_T$ using the SGD and the SFA algorithms. For our example cases, the total offered traffic load was held fixed at 0.7, with the offered external traffic load ranging from 0.5 to 0.6. Table 5 describes the system set-up for our three example cases (referred to as systems 1, 2, and 3, respectively). The simulation time required for 1,000 RC's on a DECStation 5000/25 when no IS was applied for these three cases is given in Table 4.

In applying the SGD algorithm we used the same approach as in the previous subsection, always starting with a small queue capacity and increasing until we reached the desired capacity. In each case we used the near-optimal biasing for the immediately smaller queue capacity as a starting point. Obtaining the near-optimal IS biasing for the initial queue

	System	$\theta_{op1}, \theta_{op2}, \theta_{op3}$	$\widehat{\Pr}$[loss]	95% Interval	R_{net}
1.	M-IBP+MMBBP/D/1/K $p = 0.45$, $q = 0.89$, $C_E^2 = 1.68$ $q_T = 0.95$, $K_P = 5$, K = 100	1.3074 0.9149 0.9682	1.564×10^{-7}	$(1.469 \times 10^{-7}$, $1.660 \times 10^{-7})$	1.2×10^3
2.	M-IBP+MMBBP/D/1/K $p = 0.3$, $q = 0.825$, $C_E^2 = 1.1$ $q_T = 0.97778$, $K_P = 5$, K = 100	1.4693 0.8593 0.9745	1.710×10^{-9}	$(1.498 \times 10^{-9}$, $1.921 \times 10^{-9})$	6.5×10^4
3.	M-IBP+MMBBP/D/1/K $p = 0.95$, $q = 0.99$, $C_E^2 = 26.9$ $q_T = 0.95$, $K_P = 5$, K = 1700	1.0234 0.9919 0.9978	1.207×10^{-9}	$(1.174 \times 10^{-9}$, $1.240 \times 10^{-9})$	2.3×10^5

Table 5: Estimated cell loss probabilities and speed-up factors using the SGD algorithm for the M-IBP+MMBBP/D/1/K queue. For these estimates: $N_R = 100$, $N_{RC} = 1,000$.

capacity (typically K = 20 or K = 50) was not difficult, since the corresponding loss probabilities were high and the space could be searched efficiently with the SGD algorithm starting from the brute-force MC point. Furthermore, we used $N_A = 1,000$ RC's per simulation run. The algorithm converged in all trials after $I_A < 7,000$ iterations. The step size h was obtained by trial-and-error, with a typical value of $h = 1 \times 10^{-3}$.

In applying the SFA algorithm to systems 1 and 2 we first found the near-optimal biasing for K = 40 without using any "second-order" IS. This was possible since the corresponding loss probability was high. We then used the near-optimal for K = 40 as the "second-order" IS while searching for the optimal at K = 80. Finally, we used the near-optimal for K = 80 as the "second-order" IS while searching for the optimal at K = 100. We used $N_B = 1,000$ RC's per simulation run. A similar procedure was used for system 3, however the progression of increasing buffer sizes was K = 500, K = 1100, and finally K = 1700. In each case, the algorithm required approximately $I_B = 1,000$ to "scan" the search space. The step size $h = 5 \times 10^{-4}$ was obtained by trial-and-error.

Tables 5 and 6 describe the parameter set-up, the near-optimal IS settings $(\theta_{op1}, \theta_{op2}, \theta_{op3})$ found by the SGD and SFA algorithms, respectively, the estimated loss probabilities, and the speed-up factors over conventional MC simulation. The same assumptions stated for the IBP/Geo/1/K case were used while calculating speed-up factors.

Finally, Figure 8 illustrates the results of applying the same IS setting chosen by SGD for the simulation of systems 2 and 3 in Table 5, as the queue size varies.

	System	$\theta_{op1}, \theta_{op2}, \theta_{op3}$	$\widehat{\Pr}$[loss]	95% Interval	R_{net}
1.	M-IBP+MMBBP/D/1/K $p = 0.45$, $q = 0.89$, $C_E^2 = 1.68$ $q_T = 0.95$, $K_P = 5$, K = 100	1.0128 0.9217 0.9296	1.097×10^{-7}	$(7.747 \times 10^{-8}$, $1.419 \times 10^{-7})$	8.9×10
2.	M-IBP+MMBBP/D/1/K $p = 0.3$, $q = 0.825$, $C_E^2 = 1.1$ $q_T = 0.97778$, $K_P = 5$, K = 100	1.0073 0.9382 0.8951	1.024×10^{-9}	$(4.675 \times 10^{-10}$, $1.581 \times 10^{-9})$	1.3×10^2
3.	M-IBP+MMBBP/D/1/K $p = 0.95$, $q = 0.99$, $C_E^2 = 26.9$ $q_T = 0.95$, $K_P = 5$, K = 1700	1.0068 0.9839 0.9993	8.921×10^{-10}	$(3.298 \times 10^{-10}$, $1.454 \times 10^{-9})$	1.9×10^2

Table 6: Estimated cell loss probabilities and speed-up factors using the SFA algorithm for the M-IBP+MMBBP/D/1/K queue. For these estimates: $N_R = 100$, $N_{RC} = 1,000$, except for system 2, where $N_R = 500$, $N_{RC} = 10,000$ were used.

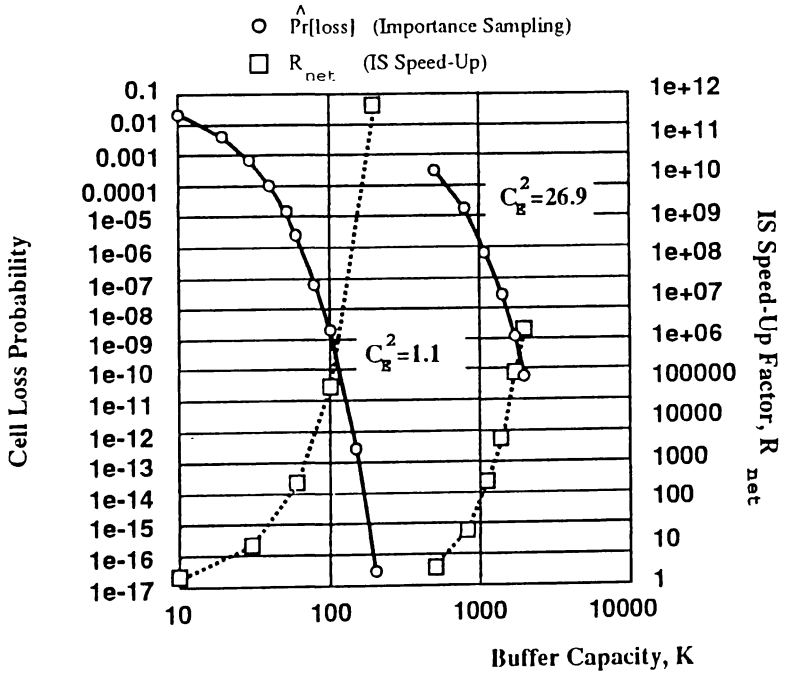

Figure 8: Estimated cell loss probabilities and IS speed-up factors, R_{net}, as a function of the queue capacity, for two example M-IBP+MMBBP/D/1/K queues (systems 2 and 3). For both cases, the IS settings were taken from Table 5, and remained fixed as K varied.

5.2.3 The SGD Algorithm for Tandem M-IBP+MMBBP/D/1/K Queues

Here, the estimate of the cell loss probability at the input of the S-th stage in the tandem network is obtained by using the SGD algorithm to minimize the estimate of the variance of the average number of tagged cells blocked per RC at the S-th stage with respect to the IS bias parameters. This requires that S stages be used to estimate the cell loss probability at the input of the S-th stage. The average number of arrivals per RC is estimated using conventional MC simulation since arrivals are not rare events.

Regeneration epochs are defined as the instants where each queue in the network is empty, the tagged traffic stream is going active and generating a cell, and all the external traffic streams in the network are idle. In each RC, q_T, p_s, and q_s are initially biased to $q_{T,1}^{*(S)}$, $p_{s,1}^{*(S)}$, $q_{s,1}^{*(S)}$ respectively, where s indexes the stage in the tandem network and S indexes the position in the tandem network which is being optimized, until one customer is blocked, then the IS parameters are changed to $q_{T,2}^{*(S)}$, $p_{s,2}^{*(S)}$, and $q_{s,2}^{*(S)}$ in order to allow fast regeneration.

In these simulations, $q_{T,2}^{*(S)} = q_T$, $p_{s,2}^{*(S)} = p_s$, and $q_{s,2}^{*(S)} = q_s$, and the optimization was performed with respect to the settings of $\theta_1 = q_{T,1}^{*(S)}/q_T$, $\theta_{2s} = p_{s,1}^{*(S)}/p_s$, and $\theta_{2s+1} = q_{s,1}^{*(S)}/q_s$ for $s = 1, \ldots, S$ using the SGD algorithm. In addition, each stage in the network was assumed to have identical parameters, $p = p_s$, $q = q_s$, and $K = K_s$. The external traffic was not allowed to propagate through more than one stage in the tandem network. For the example cases, the total offered tagged traffic load at each node was held fixed at 0.7, with the offered external traffic load ranging from 0.5 to 0.6. Table 7 describes the system set-up that was optimized for the example cases, referred to as systems 2 and 3 (consistently with Section 5.2.2.) for 1, 2, and 3-stage tandem networks.

As with the single queue case, the SGD algorithm was applied by using as the starting point the near-optimal biasing for a smaller queue capacity. Obtaining the near-optimal IS biasing for shorter buffers was not difficult, since the corresponding cell loss probabilities were high and the space could be searched efficiently with the SGD algorithm starting from the brute-force MC point. Initially, this was done for the tandem configurations as well until it was determined that the near-optimal bias parameters for a single stage could be

System	M-IBP+MMBBP/D/1/K	$\theta_{opt}^{(1)}$	$\theta_{opt}^{(2)}$	$\theta_{opt}^{(3)}$
2	$q_T = 0.97778$, $K_P = 5$ $K = 100$ $p = 0.3$, $q = 0.825$ $C_E^2 = 1.1$, $n_{max} = 5$	(0.9745, 1.4693, 0.8593)	(0.9794, 1.0009, 0.9987 1.4551, 0.8991)	(0.9808 1.0004, 0.9980 1.0042, 0.9944 1.4427, 0.9046)
3	$q_T = 0.95$, $K_P = 5$ $K = 2000$ $p = 0.95$, $q = 0.99$ $C_E^2 = 26.9$, $n_{max} = 5$	(0.9978, 1.0234, 0.9919)	(0.9979, 1.0028, 0.9992 1.0220, 0.9929)	(0.9981 1.0013, 0.9999 1.0018, 0.9998 1.0211, 0.9936)

Table 7: Optimal bias parameters using the SGD algorithm for the 1, 2, and 3 stage tandem M-IBP+MMBBP/D/1/K queues.

used as the starting point for the 2-stage tandem network by using the translation $\theta_{initial}^{(2)} = (\theta_{1,opt}^{(1)}, 1.0, 1.0, \theta_{2,opt}^{(1)}, \theta_{3,opt}^{(1)})$. Thus, the near-optimal bias parameters at the s-th stage can be used as a starting point for the optimization runs for the $(s+1)$-th stage. In fact, the near-optimal bias parameters for a single stage can be used as a starting point for the optimization runs for any multiple stage system by using the near-optimal bias parameters for the external traffic of the first stage as the initial bias parameters for the external traffic at the stage of interest. For $N_A = 1,000$ RC's per simulation iteration, the algorithm converged in all trials after $I_A < 7,000$ iterations for the first stage (as in Section 5.2.2), and $I_A < 300$ iterations for the subsequent stages. The step size h was obtained by trial-and-error, with a typical value of $h = 1 \times 10^{-3}$.

Table 7 shows the near-optimal bias found using the SGD algorithm for 1, 2, and 3-stage tandem networks for systems 2 and 3. Notice that the amount of biasing required for the near-optimal parameters at each stage decreases as the number of stages increases. This phenomenon has been seen previously in [8] when the SGD algorithm was applied to the area of wireless communications links with diversity reception. As the amount of diversity increased, the amount of biasing required for speed-up decreased. The technique of using the parameters from one stage as a starting point for an increased number of stages was also incorporated in [8] with increasing amounts of diversity instead of queueing stages.

Tables 8 – 13 illustrate the results of applying the IS setting chosen by the SGD algorithm in the simulation used to estimate the cell loss probability of systems 2 and 3 in Table 7

K	$\widehat{\Pr[\text{block}]}_1$	95% Interval	R_{net}
10	2.38×10^{-2}	$(2.26 \times 10^{-2}, 2.50 \times 10^{-2})$	1.1
30	5.67×10^{-4}	$(5.30 \times 10^{-4}, 6.03 \times 10^{-4})$	10
60	2.81×10^{-6}	$(2.31 \times 10^{-6}, 3.31 \times 10^{-6})$	140
100	1.38×10^{-9}	$(1.18 \times 10^{-9}, 1.59 \times 10^{-9})$	2.5×10^{5}
150	2.68×10^{-13}	$(2.28 \times 10^{-14}, 5.12 \times 10^{-13})$	2.3×10^{7}

Table 8: Estimated cell loss probabilities and speed-up factors for the simulation of one stage of system 2 in Table 7, as the queue size varies for $q_T = 0.9778$, $K_P = 5$, $p = 0.3$, $q = 0.825$, $C_E^2 = 1.1$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 10 to 150. For these estimates: $N_R = 20$, $N_{RC} = 1,000$.

as the queue size and number of stages in the tandem network varies. As in [22, 21], the cell loss probability for the tagged traffic is of interest. Speed-up factors are calculated as in Section 5.1. Unlike the first stage, there are no known results for the second and third stage cell loss probability. The information in Tables 8 – 13 is also plotted in Figures 9 and 10 for systems 2 and 3 respectively. In order to determine confidence intervals and speed-up factors, N_R runs of $N_{RC} = 1000$ RC's for each run were performed for varying buffer sizes using the near-optimal IS biasing values in Table 7. The estimate of the end-to-end cell loss probability is obtained from the estimates of the individual stage cell loss probabilities, given that the probability of a cell block at any one stage is mutually exclusive from the other stages.

The burstiness characteristic of the external traffic in system 2 is nearly Poisson, compared to the mildly bursty external traffic in system 3. For system 2, the cell loss probability decreases as the tagged traffic propagates through the tandem network because of the low external burstiness. This behavior is in contrast to system 3, where the cell loss probability of the tagged traffic stays relatively constant as it propagates through the tandem network because of the higher external burstiness. The change in the burstiness of the tagged traffic, shown in Table 14, is due to the fact that the external traffic is mixed into the tagged traffic stream by the random queue-filling discipline, and changes the variability of the interarrival

K	$\widehat{\Pr[\text{block}]}_2$	95% Interval	R_{net}	$\widehat{\Pr[\text{block}]}_{\text{system}}$
10	2.14×10^{-2}	$(2.07 \times 10^{-2}, 2.22 \times 10^{-2})$	MC*	4.47×10^{-2}
30	5.21×10^{-4}	$(4.71 \times 10^{-4}, 5.71 \times 10^{-4})$	MC*	1.09×10^{-3}
60	8.41×10^{-7}	$(7.12 \times 10^{-7}, 9.71 \times 10^{-7})$	97	3.65×10^{-6}
100	5.19×10^{-10}	$(3.67 \times 10^{-10}, 6.71 \times 10^{-10})$	3.1×10^{4}	1.90×10^{-9}
150	5.27×10^{-14}	$(3.20 \times 10^{-14}, 7.35 \times 10^{-14})$	1.2×10^{8}	3.20×10^{-13}

Table 9: Estimated cell loss probabilities and speed-up factors for the simulation of two stages of system 2 in Table 7, as the queue size varies for $q_T = 0.9778$, $K_P = 5$, $p = 0.3$, $q = 0.825$, $C_E^2 = 1.1$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 10 to 150. For these estimates: $N_R = 20$, $N_{RC} = 1,000$. The asterisk (*) is used to denote points where the use of IS did not result in speed-up over MC simulation, hence the point used is that found by MC simulation.

K	$\widehat{\Pr[\text{block}]}_3$	95% Interval	R_{net}	$\widehat{\Pr[\text{block}]}_{\text{system}}$
10	2.00×10^{-2}	$(1.98 \times 10^{-2}, 2.03 \times 10^{-2})$	MC*	6.38×10^{-2}
30	5.29×10^{-5}	$(4.89 \times 10^{-5}, 5.68 \times 10^{-4})$	MC*	1.62×10^{-3}
60	1.05×10^{-6}	$(1.74 \times 10^{-7}, 1.93 \times 10^{-6})$	1.3	4.70×10^{-6}
100	2.28×10^{-10}	$(1.73 \times 10^{-10}, 2.83 \times 10^{-10})$	5.8×10^{4}	2.13×10^{-9}
150	2.82×10^{-14}	$(3.17 \times 10^{-15}, 5.33 \times 10^{-14})$	2.8×10^{7}	3.48×10^{-13}

Table 10: Estimated cell loss probabilities and speed-up factors for the simulation of three stages of system 2 in Table 7, as the queue size varies for $q_T = 0.9778$, $K_P = 5$, $p = 0.3$, $q = 0.825$, $C_E^2 = 1.1$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 10 to 150. For these estimates: $N_R = 20$, $N_{RC} = 1,000$. The asterisk (*) is used to denote points where the use of IS did not result in speed-up over MC simulation, hence the point used is that found by MC simulation.

K	$\widehat{\Pr[\text{block}]}_1$	95% Interval	R_{net}
500	3.11×10^{-4}	$(3.04 \times 10^{-4}, 3.18 \times 10^{-4})$	4.3
800	1.39×10^{-5}	$(1.35 \times 10^{-5}, 1.44 \times 10^{-5})$	28
1100	6.26×10^{-7}	$(6.05 \times 10^{-7}, 6.47 \times 10^{-7})$	430
1400	2.89×10^{-8}	$(2.77 \times 10^{-8}, 3.01 \times 10^{-8})$	5000
1700	1.21×10^{-9}	$(1.17 \times 10^{-9}, 1.24 \times 10^{-9})$	2.3×10^{5}
2000	5.29×10^{-11}	$(5.08 \times 10^{-11}, 5.49 \times 10^{-11})$	2.3×10^{6}

Table 11: Estimated cell loss probabilities and speed-up factors for the simulation of one stage of system 3 in Table 7, as the queue size varies for $q_T = 0.95$, $K_P = 5$, $p = 0.95$, $q = 0.99$, $C_E^2 = 26.9$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 500 to 2,000. For these estimates: $N_R = 100$, $N_{RC} = 1,000$.

K	$\widehat{\Pr[\text{block}]}_2$	95% Interval	R_{net}	$\widehat{\Pr[\text{block}]}_{\text{system}}$
500	3.01×10^{-4}	$(2.74 \times 10^{-4}, 3.28 \times 10^{-4})$	1.0	6.12×10^{-4}
800	1.35×10^{-5}	$(1.28 \times 10^{-5}, 1.41 \times 10^{-5})$	20	2.74×10^{-5}
1100	6.72×10^{-7}	$(6.14 \times 10^{-7}, 7.29 \times 10^{-7})$	100	1.30×10^{-6}
1400	2.52×10^{-8}	$(2.32 \times 10^{-8}, 2.72 \times 10^{-8})$	2500	5.41×10^{-8}
1700	1.22×10^{-9}	$(1.12 \times 10^{-9}, 1.31 \times 10^{-9})$	4.4×10^{4}	2.43×10^{-9}
2000	5.30×10^{-11}	$(4.90 \times 10^{-11}, 5.70 \times 10^{-11})$	9.2×10^{5}	1.06×10^{-10}

Table 12: Estimated cell loss probabilities and speed-up factors for the simulation of two stages of system 3 in Table 7, as the queue size varies for $q_T = 0.95$, $K_P = 5$, $p = 0.95$, $q = 0.99$, $C_E^2 = 26.9$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 500 to 2,000. For these estimates: $N_R = 20$, $N_{RC} = 1,000$.

K	$\widehat{\Pr[\text{block}]}_3$	95% Interval	R_{net}	$\widehat{\Pr[\text{block}]}_{\text{system}}$
500	2.74×10^{-4}	$(2.46 \times 10^{-4}, 3.02 \times 10^{-4})$	1.0	8.86×10^{-4}
800	1.15×10^{-5}	$(1.03 \times 10^{-5}, 1.26 \times 10^{-5})$	3.2	3.89×10^{-5}
1100	4.97×10^{-7}	$(4.29 \times 10^{-7}, 5.66 \times 10^{-7})$	30	1.79×10^{-6}
1400	2.56×10^{-8}	$(2.24 \times 10^{-8}, 2.88 \times 10^{-8})$	580	7.97×10^{-8}
1700	1.14×10^{-9}	$(9.28 \times 10^{-10}, 1.34 \times 10^{-9})$	5100	3.56×10^{-9}
2000	4.58×10^{-11}	$(4.05 \times 10^{-11}, 5.11 \times 10^{-11})$	2.8×10^{5}	1.52×10^{-10}

Table 13: Estimated cell loss probabilities and speed-up factors for the simulation of three stages of system 3 in Table 7, as the queue size varies for $q_T = 0.95$, $K_P = 5$, $p = 0.95$, $q = 0.99$, $C_E^2 = 26.9$, and $n_{max} = 5$. IS settings were taken from Table 7, and remained fixed as K varied from 500 to 2,000. For these estimates: $N_R = 20$, $N_{RC} = 1,000$.

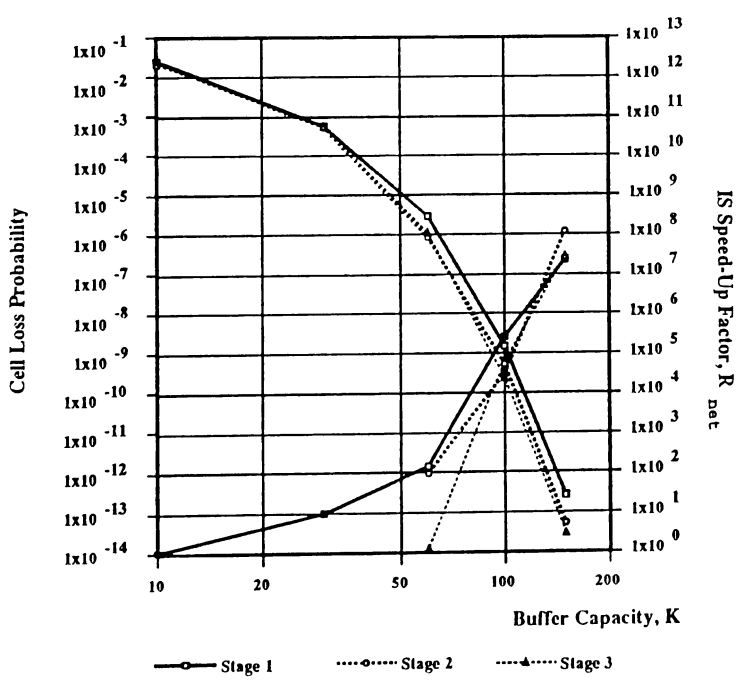

Figure 9: Cell loss probability (decreasing curves) and speed-up factors (increasing curves) for system 2.

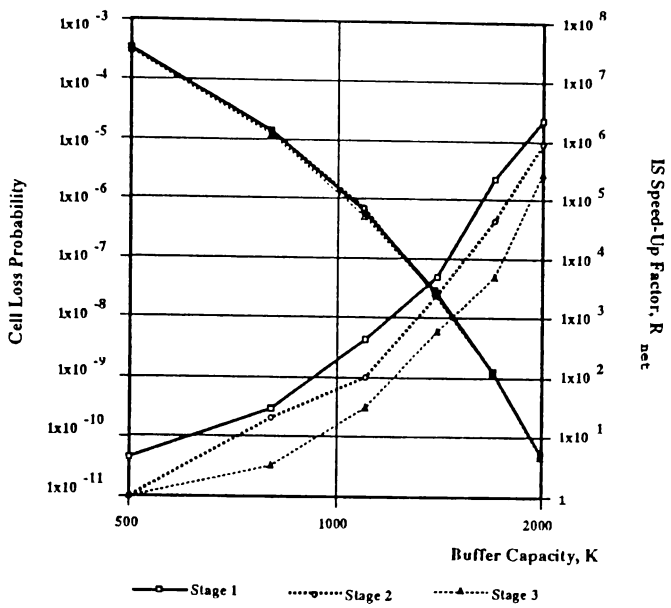

Figure 10: Cell loss probability (decreasing curves) and speed-up factors (increasing curves) for system 3.

System	External C_E^2	Tagged C_1^2	Tagged C_2^2	Tagged C_3^2	Tagged C_4^2	Tagged C_5^2
2	1.1	7.2	6.7	6.3	6.0	5.7
3	26.9	5.6	7.6	9.4	10.8	12.0

Table 14: Burstiness parameter of the external traffic, C_E^2, and the tagged traffic, C_s^2, at the input of the 1st through 5th stages for tandem M-IBP+MMBBP/D/1/K queues (systems 2 and 3 from Table 7).

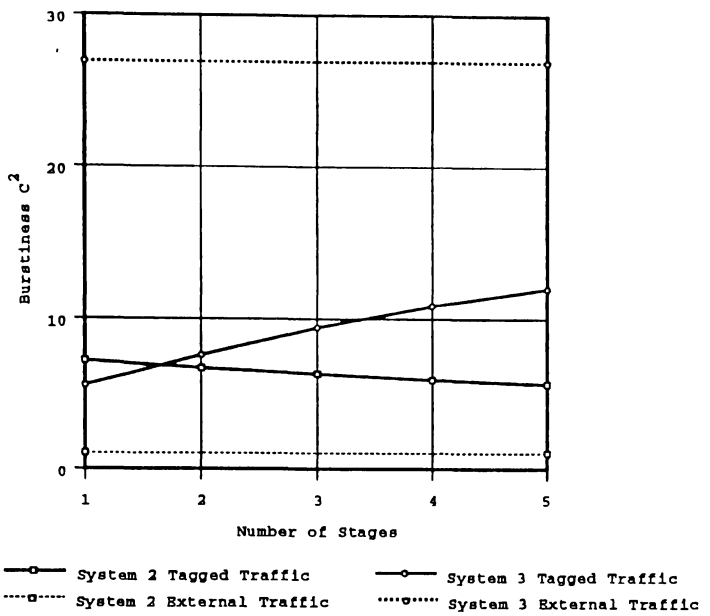

Figure 11: Tagged traffic burstiness for the first five stages of a tandem network for systems 2 and 3.

time of the tagged traffic arrivals at each stage. The change in the tagged traffic burstiness is plotted for the two systems for a five stage tandem network in Figure 11.

As with the single queue case, the statistical accuracy of the tandem cell loss estimates indicates a significant robustness of the IS speed-up factor with respect to the queue capacity, when all other system parameters remain fixed. This can be very useful in increasing the efficiency of the simulation, since the search for near-optimal IS values needs to be performed only once for the largest buffer size at each stage. Thus, when cell loss probabilities are required for several buffer sizes and stages in the network, the search overhead is divided among all cases.

The simulation time required for 1,000 RC's on a DECStation 5000/25 when no IS was applied for the two different systems and the three different tandem queue network configurations is given in Table 15. The increase in simulation time for system 2 from 1-stage to 2-stages is due to to a change in the RC conditions and the addition of the code required to handle multiple stages.

System	1-Stage CPU Time, seconds	2-Stage CPU Time, seconds	3-Stage CPU Time, seconds
2	0.342	54.025	232.175
3	20.772	56.514	256.401

Table 15: CPU Time for 1,000 RC's of tandem M-IBP+MMBBP/D/1/K queues on a DECStation 5000/25 when no IS is applied (systems 2 and 3 from Table 7).

6 Conclusions

Monte Carlo simulation using importance sampling (IS) can obtain large speed-up factors if the modification or bias of the underlying probability measures is properly chosen. In this paper, we presented the *Stochastic Gradient Descent* (SGD) algorithm and the *Stochastic (Important Event) Frequency Ascent* (SFA) algorithm, which used stochastic gradient optimization techniques to arrive at favorable IS parameter settings that increase the efficiency of the simulation of queueing networks, including queues with bursty traffic.

We demonstrated the effectiveness of our algorithms by applying them to the problem of estimating the cell loss probability of the IBP/Geo/1/K queue and tandem M-IBP+MMBBP/D/1/K queues. These queueing systems are useful building blocks in performance models for ATM switches and networks. For the examples presented, our methods achieve speed-up factors of 1 to 8 orders of magnitude over conventional Monte Carlo simulation of the estimation of the cell loss probability.

References

[1] S. Parekh and J. Walrand. A Quick Simulation Method for Excessive Backlogs in Networks of Queues. *IEEE Trans. Automat. Contr.*, AC-34(1):54–66, Jan. 1989.

[2] P. W. Glynn and D. L. Iglehart. Importance Sampling for Stochastic Simulations. *Management Science*, 35(11):1367–1392, Nov. 1989.

[3] J. S. Sadowsky and J. A. Bucklew. On Large Deviation Theory and Asymptotically Efficient Monte Carlo Estimation. *IEEE Trans. Inform. Theory*, IT-36(3):579–588, May 1990.

[4] Q. Wang and V. S. Frost. Efficient Estimation of Cell Blocking Probability for ATM Systems. *IEEE/ACM Trans. on Networking*, 1(2):230–235, 1993.

[5] C. S. Chang, P. Heidelberger, S. Juneja, and P. Shahabuddin. Effective Bandwidth and Fast Simulation of ATM Intree Networks. In *Proc. of Performance '93*, Rome, Italy, October 1993.

[6] M. Devetsikiotis and J. K. Townsend. An Algorithmic Approach to the Optimization of Importance Sampling Parameters in Digital Communication System Simulation. *IEEE Trans. Commun.*, 41(10), Oct. 1993.

[7] M. Devetsikiotis and J. K. Townsend. Statistical Optimization of Dynamic Importance Sampling Parameters for Efficient Simulation of Communication Networks. *IEEE/ACM Trans. Networking*, 1(3), June 1993.

[8] W. Al-Qaq, M. Devetsikiotis, and J. K. Townsend. Simulation of Digital Communication Systems Using a Stochastically Optimized Importance Sampling Technique. In *Proc. IEEE Global Telecom. Conf., GLOBECOM '93*, Houston, Dec. 1993.

[9] R. Y. Rubinstein. Sensitivity Analysis and Performance Extrapolation for Computer Simulation Models. *Operns. Res.*, 37:72–81, 1989.

[10] M. I. Reiman and A. Weiss. Sensitivity Analysis for Simulations Via Likelihood Ratios. *Operns. Res.*, 37:830–844, 1989.

[11] P. W. Glynn. Stochastic Approximation for Monte Carlo Optimization. In *Proc. of the Winter Simulation Conference*, Wilson, J., Henriksen, J. and Roberts, S. (eds), IEEE Press, 1986.

[12] P. W. Glynn. Likelihood Ratio Gradient Estimation: An Overview. In *Proc. of the Winter Simulation Conference*, Thesen, A., Grant, H., Kelton, W. D., (eds), IEEE Press, 1987.

[13] M. K. Nakayama, A. Goyal, and P. W. Glynn. Likelihood Ratio Sensitivity Analysis for Markovian Models of Highly Dependable Systems. Technical Report RC 15400 (#68500), IBM Research Division, T. J. Watson Research Center, Jan. 1990.

[14] A. Goyal, P. Shahabuddin an P. Heidelberger, V. F. Nicola, and P. W. Glynn. A Unified Framework for Simulating Markovian Models of Highly Dependable Systems. *IEEE Trans. Computers*, 41(1):36–51, Jan. 1992.

[15] F. L. Gunther and R. W. Wolff. The Almost Regenerative Method for Stochastic System Simulations. *Operations Research*, 28(2):375–386, March-April 1980.

[16] Y. C. Ho and C. Cassandras. A New Approach to the Analysis of Discrete Event Dynamic Systems. *Automatica*, 19:149–167, 1983.

[17] M. Metivier and P. Priouret. Applications of a Kushner and Clark Lemma to General Classes of Stochastic Algorithms. *IEEE Trans. Inform. Theory*, IT-30(2):140–151, March 1984.

[18] R. J. Wolfe, M. C. Jeruchim, and P. M. Hahn. On Optimum and Suboptimum Biasing Procedures for Importance Sampling in Communication Simulation. *IEEE Trans. Commun.*, COM-38(5):639–647, May 1990.

[19] A. A. Nilsson, F. Lai, and H. G. Perros. An Approximate Analysis of a Bufferless N x N Synchronous Clos ATM Switch. In *Proc. 13th Int. Teletraffic Congress, ITC 13*, Copenhagen, Denmark, June 1991.

[20] M. C. Jeruchim. Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems. *IEEE J. Select. Areas Commun.*, SAC-2(1):153–170, Jan. 1984.

[21] H. Yamashita and R. O. Onvural. On the Cell Loss Distribution of Protocol Data Units in ATM Networks. In *Proc. 5th Int. Conf. on Data Comm. Systems and their Performance: High-Speed Networks*, Raleigh, NC, Oct. 1993.

[22] 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.