Submitted to the *International Conference on Communications, ICC '96*

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

James A. Freebersyser Michael Devetsikiotis[†] Wael A. Al-Qaq^{††} J. Keith Townsend

[†] Dept. of Systems & Computer Engineering Carleton University Ottawa, Ontario K1S 5B6, Canada Tel: (613) 788-2600 ×1952, Fax: (613) 788-5727 mike@sce.carleton.ca	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 jkt@eos.ncsu.edu
--	---

^{††}Compact Software
201 McLean Boulevard
Paterson, New Jersey 07504, U.S.A.
Tel: (201) 881-1200, Fax: (201) 881-8361
al-qaq@jupiter.comsoft.com

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 utilize the Stochastic Gradient Descent (SGD) algorithm, which uses stochastic gradient optimization techniques, to arrive at favorable IS bias parameter settings for the simulation of tandem queues with bursty traffic, geometric service times and a finite buffer. We describe in detail the experimental method associated with applying the SGD algorithm. Speed-up factors of 1 to 8 orders of magnitude over conventional Monte Carlo estimation of the cell loss probability are achieved for the examples presented.¹

¹This work was supported in part by the Center for Communications & Signal Processing, North Carolina State University, and by the Telecommunications Research Institute of Ontario. James A. Freebersyser is supported in part by the U.S. Air Force Palace Knight Program.

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 in network (queueing) systems [1]-[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 bias parameter settings for complex and/or large networks with bursty traffic.

Fast simulation methods based on Large Deviation Theory (LDT) [1, 3] have been successfully applied in many cases (recently, most notably in [5]). When feasible, LDT-like solutions provide asymptotically optimal results, leading to estimators that exhibit asymptotic efficiency, i.e., a variance reduction of the same order as the estimate itself. This implies that no matter how rare the event becomes, the simulation length required for a fixed relative error (confidence interval) remains bounded. However, LDT-based methods utilize asymptotical analytical knowledge and analytical/numerical manipulations which are not feasible yet for many realistic systems.

An alternative 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 in [6, 7]. Stochastic optimization techniques trade-off “true optimality” for conceptual simplicity and general applicability. At the expense of set-up and simulation overhead, such techniques are not restricted to a specific type of random process, and do not require knowledge of the internal workings of the system being simulated. Therefore, stochastic optimization techniques are not only appealing, but the only alternative when the system complexity precludes the application of LDT-like methods.

The Mean-Field Annealing (MFA) global optimization algorithm, which is a form of Simulated Annealing (SA), has also been used for finding near-optimal bias parameter values for queueing system simulation [7]. Although very general, the MFA approach is potentially

slow, especially as the dimensionality of the search (i.e., the number of bias parameters) increases. Therefore, techniques that perform a directed search through the bias parameter space (e.g., using derivative information) while requiring a smaller number of repetitive simulations can provide attractive alternatives to stochastic annealing techniques.

Previously, we have presented a stochastic gradient technique for the near-minimization of IS-variance in the simulation of queueing systems [8], including tandem networks [9]. The Stochastic Gradient Descent (SGD) algorithm uses MC estimates of the gradient of a cost function using likelihood ratio methods, as described in [10]-[13]. The SGD algorithm uses estimates of the IS-variance and its gradient with respect to the bias parameters, and follows a stochastic steepest descent path to near-optimal bias parameter settings.

In this paper, we consider tandem queueing networks with bursty traffic, geometric service times and finite buffers. Such queueing networks can not be solved by methods such as those in [5], which can be used only on queueing networks with deterministic service times, which we also considered in [9]. We describe the experimental method associated with applying the SGD algorithm and illustrate the effectiveness of this technique with numerical examples. Speed-up factors of 1 to 8 orders of magnitude over conventional MC simulation, at low cell loss probabilities (e.g., 10^{-7} to 10^{-14}), are obtained for tandem M-IBP+MMBBP/Geo/1/ K queues.

2 Efficient Simulation of Communication Networks

2.1 MC and IS Estimation in Network Analysis

Let $\mathbf{X}_i$ be the vector of observations relevant to a slotted-time communications network at time i . Let $U(\mathbf{s})$ be given by $U(\mathbf{s}) = \sum_{i=0}^{\tau} u(\mathbf{X}_i)$, where τ is the termination time, and $\mathbf{s}$ denotes a sample path in the evolution of the system under study. Let $E_P[U(\mathbf{s})]$ denote the expectation of $U(\mathbf{s})$ with respect to a probability measure $P(\mathbf{s})$. A MC estimator for $E_P[U(\mathbf{s})]$ is given by $1/N \sum_{k=1}^N \sum_{i=0}^{\tau-1} u(\mathbf{X}_{ik})$, where $\mathbf{X}_{ik}$ is the vector of observations at time i of sample path k .

IS is based on the observation that the expectation $E[U(\mathbf{s})]$ under measure P can be written as $E_P[U(\mathbf{s})] = E_{P^*}[U(\mathbf{s})L^*(\mathbf{s})]$, where $L^*(\mathbf{s}) = P(\mathbf{s})/P^*(\mathbf{s})$ and provided that $P^*(\mathbf{s}) \neq$

0 whenever $U(\mathbf{s})P(\mathbf{s}) \neq 0$. L^* is a likelihood ratio or, in the language of IS, a weight function.

If $\{\mathbf{X}_i\}_{i \geq 0}$ converges in distribution to $\mathbf{X}$ regardless of initial conditions, we are typically interested in steady-state performance measures such as the expectation $E[f(\mathbf{X})]$ of some function $f(\mathbf{X}) = u(\mathbf{X})/v(\mathbf{X})$. Steady-state estimators are commonly approximated by allowing τ above to become large and providing for a warm-up period before collecting statistics. An alternative, precise, approach involves the regenerative method [14]. Following the formulation in [15], let $\mathbf{r}$ be a regeneration state. Then, the expectation of f can be written as

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

where $\mathbf{X}_0 = \mathbf{r}$, and τ_1 is the first time greater than zero that $\mathbf{X}_i = \mathbf{r}$. Using IS, $E[f]$ can be estimated by

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

where $L_{ik}^* = P(\mathbf{X}_{0k}, \dots, \mathbf{X}_{ik})/P^*(\mathbf{X}_{0k}, \dots, \mathbf{X}_{ik})$, $k = 1, 2, \dots$. In general, the numerator and denominator of (2) can be estimated separately, with $M \neq N$ and different IS distributions [4, 15].

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 regeneration cycle (RC). For cases where true regenerations are rare, techniques based on approximate regeneration [16], batch means [17], or A-cycles [5], can be used to obtain approximately independent trials.

The case of Markov chains is especially important since many practical traffic models (e.g., ON-OFF, MMPP, batch-geometric, autoregressive, Transform-Expand-Sample) fall under this broad class of random processes (see [18]). If $\{\mathbf{X}_i\}_{i \geq 0}$ is a discrete-time Markov chain with transition matrix $\mathbf{P}$, and transition probabilities $p(\mathbf{X}_j, \mathbf{X}_{j+1})$, then within each RC, $k = 1, 2, \dots$,

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

In (2) above, the likelihood ratio at time i during the simulation depends on all random transitions which previously occurred in the same RC. Also 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 [15]. 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].

2.2 Statistical Optimization of the IS Estimator

It is well known that the general, non-parametric, globally optimal IS measure is 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 bias parameter settings can be posed as a multidimensional, nonlinear optimization problem, where the values of the bias parameters must be set to optimize some measure of performance, usually the estimator variance, $\sigma_{IS}^2(P, P^*)$.

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

2.3 Stochastic Gradient Techniques

Performance measures of communication systems and networks often take the form of an expectation $a(\boldsymbol{\theta})$ that depends on a vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$ 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 $\boldsymbol{\theta}$. As in the case of the original expectation $a(\boldsymbol{\theta})$, analytical calculation of $\nabla a(\boldsymbol{\theta})$ is often intractable. The Monte Carlo estimation of derivatives and gradients of expectations, based on likelihood ratios has been previously investigated in [10]-[13], and the complete treatment of the SGD algorithm with importance sampling in a communications network context is given in [8].

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 [11, 19],

$$\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n - h(n)\widehat{\nabla}a(\boldsymbol{\theta}_n) \quad (4)$$

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

2.4 The SGD Algorithm for IS in Tandem Networks

We observe now that the variance $\sigma_{IS}^2(P, P^*)$ of the IS estimator in (2) is also an expectation parameterized by the bias parameter settings. Recall that we wish to choose bias parameter values in a way that minimizes this IS-variance. Therefore, by letting $a(\boldsymbol{\theta}) = \sigma_{IS}^2(P, P^*)$, where $\boldsymbol{\theta}$ is the vector of bias parameters in (2) that determine P^* , we can formulate the choice of bias parameter values as a minimization problem (i.e., $\min_{\boldsymbol{\theta}} a(\boldsymbol{\theta})$) that can be tackled according to (4).

Such an approach to optimizing the choice of bias parameter settings is a natural complement to our previous statistical optimization techniques [6, 7], where we determine near-optimal bias parameter values by observing estimates of a 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 bias parameter settings faster than global search techniques like MFA and similar annealing methods. The essential difference from annealing techniques is that gradient-based techniques are local optimization methods.

For a given point θ in the bias parameter space, the random numbers used to estimate $\nabla_{\theta} \sigma_{IS}^2(P, P^*)$ 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 bias parameters are found.

In [9], we developed the SGD algorithm for using IS in tandem networks. The SGD algorithm for determining the near-optimal bias parameters for estimating the cell loss probability in tandem queues is outlined in Figure 1. Previously, the SGD algorithm was applied to tandem networks with deterministic service times [9]. Here, we apply the SGD algorithm to tandem networks with geometric service times, systems which can not be solved with techniques such as those in [5].

Several heuristic arguments can be used to identify a starting point for the search in (4) when important events are rare. For example, near-optimal bias parameters for the case where the important events are not rare (e.g., smaller buffer size for cell loss probability) can be found first and used as a starting point for the rare event case. We comment more on our experimental method and the selection of the parameters used in the SGD algorithm in section 3.2.

3 Tandem M-IBP+MMBBP/Geo/1/ K Queues

3.1 System Description

The recent attention paid to ATM technology has made simulation of tandem networks of great interest. Tandem networks of M-IBP+MMBBP/1/Geo/ 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 geometric server models the link carrying the traffic to the next node in the network.

As described in [20] and shown in Figure 2, a single stage of this slotted time queueing model has one server with a buffer that holds K cells. There are two independent traffic

```

 $I_A \leftarrow 0$  /* Initialize total iteration count */
 $i \leftarrow 1$  /* Initialize buffer size search point count */
 $h_i \leftarrow h$  /* Initialize step size */
 $n_i \leftarrow -1$  /* Initialize buffer size iteration count */
 $\alpha_i \leftarrow \alpha$  /* Initialize stopping parameter */
 $\theta_0^{(S)} \leftarrow \theta_{\text{CMC}}^{(S)}$  /* Initialize bias parameter values to conventional MC point */
 $K_i \leftarrow K_{\min}$  /* Initialize buffer size */
 $N_{A_i} \leftarrow N_A$  /* Initialize number of RC per buffer size search point */

/* Perform SGD on  $\hat{\sigma}^2(\hat{P}(\theta_{n_i}^{(S)}))$  */

do { /* loop over buffer size */

  do {

     $n_i = n_i + 1$  /* Increment iteration count */
    Determine  $\hat{P}(\theta_{n_i}^{(S)})$ ,  $\hat{\sigma}^2(\hat{P}(\theta_{n_i}^{(S)}))$ , and  $\hat{\nabla} \hat{\sigma}^2(\hat{P}(\theta_{n_i}^{(S)}))$  using  $N_{A_i}$  RC
     $\theta_{n_i+1}^{(S)} \leftarrow \theta_{n_i}^{(S)} - h_i \hat{\nabla} \hat{\sigma}^2(\hat{P}(\theta_{n_i}^{(S)})) / \max_j \frac{\partial}{\partial \theta_j} \hat{\sigma}^2(\hat{P}(\theta_{n_i}^{(S)}))$  /* Perform descent step */
     $\Delta_{n_i+1}^{(S)} = \|\theta_{n_i+1}^{(S)} - \theta_{n_i}^{(S)}\|$  /* Compute norm */

  } while {  $\Delta_{n_i+1}^{(S)} > \alpha_i$  }

   $I_{A_i} \leftarrow n_i$  /* Store buffer size iteration count */
   $I_A = I_A + n_i$  /* Add to total iteration count */
   $N_{A_{i+1}} \leftarrow N_{A_i} + \Delta N_A$  /* Increase number of RC per buffer size search point */
   $K_{i+1} \leftarrow K_i + \Delta K$  /* Increase buffer size */
   $\alpha_{i+1} \leftarrow \alpha_i - \Delta \alpha$  /* Reduce stopping parameter */
   $h_{i+1} \leftarrow h_i - \Delta h$  /* Reduce step size */
   $i = i + 1$  /* Increase buffer size search point count */
   $n_i \leftarrow -1$  /* Initialize buffer size iteration count */
   $\theta_{n_i+1}^{(S)} \leftarrow \theta_{n_{i-1}}^{(S)}$  /* First iteration of next buffer size point is final iteration of previous buffer size point */

} while {  $K_i < K_{\max}$  }

```

Figure 1: Pseudo-code describing the SGD algorithm for determining near-optimal bias parameters for estimating the cell loss probability in tandem networks.

Figure 2: M-IBP+MMBBP/Geo/1/K tandem queues.

streams entering the first stage. The first stream, called the tagged traffic [20], 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 N_P slots, where N_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 tagged traffic remains idle in the next slot, and a probability $1 - q_t$ that the tagged traffic transitions to the active state in the next slot.

The second stream, called the external traffic [20], is modeled by a Markov Modulated Bernoulli Batch Process (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 cell arrivals. The number of cells m arriving in a busy slot is described by some distribution $b_i(m)$ for each state $i = 1, \dots, N_S$ of the MMBP. We assume the MMBBP has two states, $N_S = 2$, which we call the active and idle states, resulting in an Interrupted Bernoulli Batch Process (IBBP). When the external traffic is active, cell arrivals occur and there is a probability p that the external traffic remains active in the next slot, and a probability $1 - p$ that the external traffic transitions to the idle state in the next slot. In the active state, $b_1(0) = 0$ and

arrivals occur with a uniform batch-size distribution, $b_1(m) = 1/n_{max}$ for $m = 1, \dots, n_{max}$. When the external traffic is idle, there are no cell arrivals, $b_2(0) = 1$, and $b_2(m) = 0$ for $m = 1, \dots, n_{max}$, and there is a probability q that the external traffic remains idle in the next slot, and a probability $1 - q$ that the external traffic transitions to the active state in the next slot.

When tagged and external cell arrivals occur in the same slot, they are inserted into the buffer in a random order. A cell loss occurs when there is no room in the buffer for the arriving cells. According to the geometric service, if the buffer is non-empty, the probability that the cell at the head of the buffer receives service in a slot is $1 - \sigma$.

For an S -stage tandem network of M-IBP+MMBBP/Geo/1/ K queues, p , q and σ are indexed for each stage as p_s , q_s and σ_s for $s = 1, \dots, S$. Similarly, each stage in the tandem network has a finite buffer size of K_s cells. The tagged traffic always continues from one stage in the tandem network to the next stage in the tandem network, while the external traffic exits the system after the stage in which it enters 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 MMBBP modeling the external traffic. The burstiness of the external traffic entering the network at each stage is denoted by C_E^2 .

3.2 Experimental Method

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 cell losses per RC at the S -th stage with respect to the 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 (corresponding to the denominator in (2)) is estimated using conventional MC simulation since arrivals are not rare events [4].

Regeneration epochs are defined as the instants where each buffer 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 , q_s and σ_s are initially biased to $q_{t,1}^{*(S)}$,

System	M-IBP+MMBBP/Geo/1/ K	$\theta_{opt}^{(1)}$	$\theta_{opt}^{(2)}$	$\theta_{opt}^{(3)}$
G1	$q_t = 0.97778, N_P = 5$ $K = 150, \sigma = 0.01$ $p = 0.3, q = 0.825$ $C_E^2 = 1.1, n_{max} = 5$	(0.9779, 1.4828, 0.8708, 1.0822)	(0.9812, 0.9995, 0.9998, 1.0004, 1.4798, 0.8885, 1.0829)	(0.9791, 0.9996, 1.0057, 0.9999, 1.0016, 1.0045, 1.0005, 1.4761, 0.8855, 1.0840)
G2	$q_t = 0.95, N_P = 5$ $K = 2000, \sigma = 0.01$ $p = 0.95, q = 0.99$ $C_E^2 = 26.9, n_{max} = 5$	(0.9983, 1.0228, 0.9920, 1.0001)	(0.9985, 1.0028, 0.9998, 0.9998, 1.0211, 0.9927, 1.0012)	(0.9986, 1.0024, 0.9998, 1.0003, 1.0026, 0.9997, 0.9999, 1.0185, 0.9942, 1.0005)

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

$p_{s,1}^{*(S)}$, $q_{s,1}^{*(S)}$ and $\sigma_{s,1}^{*(S)}$ 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 cell is lost, then the bias parameters are changed to $q_{t,2}^{*(S)}$, $p_{s,2}^{*(S)}$, $q_{s,2}^{*(S)}$ and $\sigma_{s,2}^{*(S)}$ in order to allow fast regeneration.

In these simulations, $q_{t,2}^{*(S)} = q_t$, $p_{s,2}^{*(S)} = p_s$, $q_{s,2}^{*(S)} = q_s$ and $\sigma_{s,2}^{*(S)} = \sigma_s$ and the optimization was performed with respect to the bias parameter settings of $\theta_1 = q_{t,1}^{*(S)}/q_t$, $\theta_{3s-1} = p_{s,1}^{*(S)}/p_s$, $\theta_{3s} = q_{s,1}^{*(S)}/q_s$ and $\theta_{3s+1} = \sigma_{s,1}^{*(S)}/\sigma_s$ for $s = 1, \dots, S$ using the SGD algorithm. In addition, each stage in the network was assumed to have identical parameters, $p = p_s$, $q = q_s$, $\sigma = \sigma_s$ and $K = K_s$. 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 1 describes the system set-up that was optimized for the example cases, referred to as systems G1 and G2 for 1, 2 and 3-stage tandem networks.

The SGD algorithm was applied by using as the starting point the near-optimal IS bias parameters for a smaller buffer capacity. We examine in detail the search for the near-optimal IS bias parameters for one stage of system G2. As shown in Table 2, the optimization runs consisted of four stages, where N_s is the expected number of slots per stage. The search began at a buffer size of $K = 500$ where the cell loss probability was high and the bias parameter space could be searched efficiently with the SGD algorithm starting from the conventional MC point. The value of the step size for $K = 500$ was $h = 1 \times 10^{-2}$. The actual step size in the SGD algorithm in Figure 1 is normalized to the largest partial derivative so that the search takes the largest step in the direction of greatest change. This means that the bias parameter with the largest partial derivative can change by an amount of h from one

	1		2			3	4
K	500	800	1100	1400	1700	2000	2000
h	1×10^{-2}		2×10^{-3}			1×10^{-3}	2×10^{-4}
α							4×10^{-4}
N_s	482	778	1085	1406	1750	2013	2013

Table 2: Parameters used in the SGD algorithm in the search for the near-optimal bias parameters for one stage of system G2.

Figure 3: Normalized estimator variance for $K = 500$ for run 0 of one stage of system G2.

iteration to the next. The value of α is not yet critical since we only need an estimate of the optimal for $K = 500$ to begin the search for the next buffer size. The normalized estimator variance for a search run for $K = 500$ shown in Figure 3 demonstrates how the estimator variance is reduced as the SGD algorithm steers the search toward the near-optimal bias parameters as a result of more efficient simulation using IS.

The near-optimal bias parameters at each buffer size were used as the starting point for the optimization run at the next longer buffer size that was considered. In this case, the near-optimal bias parameters for $K = 500$ were used as the starting point for the $K = 1100$ search, and the near-optimal bias parameters for $K = 1100$ were used as the starting point for the $K = 2000$ search. There are several heuristics that allow us to know when the vicinity of the optimal has been reached: the estimates of the partial derivatives change sign near the optimal, and the normalized estimator variance reaches a minima and begins to oscillate

Run	$\theta_{opt}^{(1)}$	I_{A_1}	I_{A_2}	I_{A_3}	I_{A_4}	Search Overhead	R_{net} Reduction
0	(0.9983, 1.0228, 0.9920, 1.0001)	92	75	10	21	1.88×10^8	2.25
1	(0.9980, 1.0225, 0.9916, 1.0001)	96	75	9	18	1.82×10^8	2.21
2	(0.9983, 1.0228, 0.9917, 1.0002)	107	38	32	6	1.69×10^8	2.13
3	(0.9981, 1.0226, 0.9917, 1.0001)	185	55	7	12	1.87×10^8	2.24
4	(0.9981, 1.0230, 0.9918, 1.0001)	99	48	13	7	1.40×10^8	1.93
5	(0.9982, 1.0230, 0.9917, 1.0002)	103	27	20	18	1.55×10^8	2.03
6	(0.9980, 1.0233, 0.9919, 1.0000)	123	22	7	14	1.25×10^8	1.83
7	(0.9980, 1.0228, 0.9918, 1.0000)	126	51	16	19	1.87×10^8	2.24
8	(0.9977, 1.0228, 0.9918, 1.0001)	109	24	21	6	1.33×10^8	1.88
9	(0.9980, 1.0230, 0.9919, 1.0001)	160	61	14	4	1.80×10^8	2.19

Table 3: Near-optimal bias parameters found for 10 searches for one stage of system G2, the number of runs for each buffer size of the search the overall cost of the search and the resulting reduction in R_{net} .

as the SGD algorithm gets near the optimal then moves away only to return again. The SGD algorithm is robust in that the partial derivative estimates can be rough, although IS improves the estimates of the partial derivatives as well as the estimate of the cell loss itself. The stopping criteria is related to the norm of the distance of the step through the bias parameter space. Since the SGD algorithm always moves at the least a normalized distance h through the parameter space, we make h smaller once the vicinity of the optimal is reached, which is why two different values of h were used for the $K = 2000$ buffer size. As given in Table 2, we choose $\alpha = 4 \times 10^{-4}$ using $\alpha = 2h$ to end the search for the near-optimal bias parameters for one stage of system G2.

In order to verify the near-optimal bias parameters found by the SGD algorithm, we performed 10 searches for one stage of system G2 with the identical parameters given in Table 2 for the SGD algorithm. The only difference in the searches was the seed used by the random number generator. As shown in Table 3, the number of iterations at each stage in the algorithm varied because of the stochastic nature of the search using the SGD algorithm, yet the final bias parameters for each search are within 5×10^{-4} of the average bias parameter value. This demonstrates that the SGD algorithm arrives at bias parameters that are near-optimal because of the stochastic nature of the search, but also that the SGD algorithm consistently arrives in the vicinity of the optimal bias parameters.

Table 3 also computes the cost of the search in terms of the number of slots that were simulated. We assume on average that N_s slots in Table 2 are simulated for each RC, which is a conservative assumption since the near-optimal bias parameters used to determine N_s

are more biased than the bias parameters used during the search, resulting in a longer RC in terms of the number of slots. The cost of the MC simulations was 1.50×10^8 slots. Thus, from Table 3, the overhead involved with the search for the near-optimal bias parameters reduces R_{net} by a factor of roughly 2. Since R_{net} is not effected by minor perturbations in the bias parameters, the near-optimal bias parameters offer a speed-up that is close enough to the optimal for efficient simulation.

3.3 Numerical Examples

It was determined that the near-optimal bias parameters for a single stage could be used as the starting point for the 2-stage tandem network by using the translation $\theta_{initial}^{(2)} = (\theta_{1,opt}^{(1)}, 1.0, 1.0, 1.0, \theta_{2,opt}^{(1)}, \theta_{3,opt}^{(1)}, \theta_{4,opt}^{(1)})$. 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. For $N_A = 1,000$ RC per simulation iteration, the algorithm converged for both systems after $I_A < 7,000$ iterations for the first stage and $I_A < 2,000$ iterations for the subsequent stages.

Table 1 shows the near-optimal bias parameter values found using the SGD algorithm for 1, 2 and 3-stage tandem networks for systems G1 and G2. In order to determine confidence intervals and speed-up factors over conventional MC simulation, $N_R = 20$ sets of $N_{RC} = 1000$ RC per set were performed for varying buffer sizes using the near-optimal IS bias parameter values in Table 1. The confidence intervals of the estimate of the cell loss probability and the speed-up factors over conventional MC simulation provided by simulation using IS are determined using the method in [7]. Tables 4 and 5 give the estimates of the cell loss probability, the 95% confidence intervals and the R_{net} that result from applying the bias parameter settings chosen by the SGD algorithm in the simulation of systems G1 and G2 in Table 1 as the buffer size and number of stages in the tandem network varies. The results in Tables 4 and 5 are also plotted in Figures 4 and 5 for systems G1 and G2, respectively.

The burstiness of the external traffic in system G1 is very low, compared to the mildly bursty external traffic in system G2. For system G1, the cell loss probability decreases as the tagged traffic propagates through the tandem network because of the low external traffic burstiness. This behavior is in contrast to system G2, where the cell loss probability of the

K	$\hat{P}_{CL}^{(1)}$	95% Interval	R_{net}	$\hat{P}_{CL}^{(2)}$	95% Interval	R_{net}	$\hat{P}_{CL}^{(3)}$	95% Interval	R_{net}
10	2.54×10^{-2}	$(2.47 \times 10^{-2}, 2.61 \times 10^{-2})$	CMC	2.28×10^{-2}	$(2.23 \times 10^{-2}, 2.32 \times 10^{-2})$	CMC	2.14×10^{-2}	$(2.10 \times 10^{-2}, 2.18 \times 10^{-2})$	CMC
30	6.04×10^{-4}	$(5.06 \times 10^{-4}, 7.03 \times 10^{-4})$	CMC	5.73×10^{-4}	$(5.08 \times 10^{-4}, 6.38 \times 10^{-4})$	CMC	5.58×10^{-4}	$(5.18 \times 10^{-4}, 5.98 \times 10^{-4})$	CMC
60	3.13×10^{-6}	$(2.06 \times 10^{-6}, 4.20 \times 10^{-6})$	10	1.60×10^{-6}	$(9.68 \times 10^{-7}, 2.23 \times 10^{-6})$	8.1	5.91×10^{-7}	$(2.78 \times 10^{-7}, 9.04 \times 10^{-7})$	6.1
100	1.75×10^{-9}	$(1.43 \times 10^{-9}, 2.06 \times 10^{-9})$	4.2×10^4	1.03×10^{-9}	$(7.16 \times 10^{-10}, 1.35 \times 10^{-9})$	1.5×10^4	4.67×10^{-10}	$(2.76 \times 10^{-10}, 6.58 \times 10^{-10})$	1.1×10^4
150	2.94×10^{-13}	$(1.60 \times 10^{-13}, 4.27 \times 10^{-13})$	2.8×10^7	1.01×10^{-13}	$(7.26 \times 10^{-14}, 1.30 \times 10^{-13})$	1.4×10^8	3.73×10^{-14}	$(2.60 \times 10^{-14}, 4.85 \times 10^{-14})$	2.1×10^8

Table 4: Estimated cell loss probabilities, 95% confidence intervals and speed-up factors for the simulation of 1, 2 and 3-stages of system G1 in Table 1 as the buffer size varies. Bias parameter settings were taken from Table 1 and remained fixed as K varied from 10 to 150. For these estimates $N_R = 20$ and $N_{RC} = 1,000$. CMC denotes points where the use of IS did not result in speed-up over conventional MC simulation, hence the estimate given is that found by conventional MC simulation.

K	$\hat{P}_{CL}^{(1)}$	95% Interval	R_{net}	$\hat{P}_{CL}^{(2)}$	95% Interval	R_{net}	$\hat{P}_{CL}^{(3)}$	95% Interval	R_{net}
500	3.44×10^{-4}	$(3.22 \times 10^{-4}, 3.65 \times 10^{-4})$	1.5	3.85×10^{-4}	$(3.44 \times 10^{-4}, 4.27 \times 10^{-4})$	*	3.45×10^{-4}	$(2.97 \times 10^{-4}, 3.93 \times 10^{-4})$	*
800	1.70×10^{-5}	$(1.60 \times 10^{-5}, 1.79 \times 10^{-5})$	21	1.71×10^{-5}	$(1.55 \times 10^{-5}, 1.87 \times 10^{-5})$	4.5	2.42×10^{-5}	$(1.38 \times 10^{-5}, 3.46 \times 10^{-5})$	*
1100	7.98×10^{-7}	$(7.55 \times 10^{-7}, 8.41 \times 10^{-7})$	390	8.76×10^{-7}	$(7.90 \times 10^{-7}, 9.61 \times 10^{-7})$	57	7.37×10^{-7}	$(6.37 \times 10^{-7}, 8.36 \times 10^{-7})$	19
1400	4.00×10^{-8}	$(3.81 \times 10^{-8}, 4.20 \times 10^{-8})$	7.0×10^3	4.68×10^{-8}	$(3.87 \times 10^{-8}, 5.49 \times 10^{-8})$	280	3.63×10^{-8}	$(3.12 \times 10^{-8}, 4.13 \times 10^{-8})$	300
1700	1.94×10^{-9}	$(1.87 \times 10^{-9}, 2.01 \times 10^{-9})$	2.0×10^5	1.92×10^{-9}	$(1.54 \times 10^{-9}, 2.30 \times 10^{-9})$	4.2×10^3	1.96×10^{-9}	$(1.61 \times 10^{-9}, 2.31 \times 10^{-9})$	2.8×10^3
2000	9.04×10^{-11}	$(8.64 \times 10^{-11}, 9.44 \times 10^{-11})$	2.7×10^6	1.12×10^{-10}	$(7.90 \times 10^{-11}, 1.46 \times 10^{-10})$	2.8×10^4	9.12×10^{-11}	$(6.68 \times 10^{-11}, 1.16 \times 10^{-10})$	2.3×10^4

Table 5: Estimated cell loss probabilities, 95% confidence intervals and speed-up factors for the simulation of 1, 2 and 3-stages of system G2 in Table 1 as the buffer size varies. Bias parameter settings were taken from Table 1 and remained fixed as K varied from 500 to 2,000. For these estimates $N_R = 20$ and $N_{RC} = 1,000$. The asterisk (*) denotes points where $R_{net} < 1$, yet the estimate found by simulation using IS was still more accurate than the estimate found using conventional MC simulation.

Figure 4: Estimated cell loss probabilities (decreasing curves) and speed-up factors (increasing curves) for system G1.

Figure 5: Estimated cell loss probabilities (decreasing curves) and speed-up factors (increasing curves) for system G2.

System	1-stage G1	2-stages G1	3-stages G1	1-stage G2	2-stages G2	3-stages G2
CPU time, seconds	11.7	64.7	275.9	11.3	60.2	271.0

Table 6: CPU time for 1,000 RC of tandem M-IBP+MMBBP/Geo/1/ K queues on a DECStation 5000/25 when no IS is applied for systems G1 and G2 from Table 1.

tagged traffic stays relatively constant as it propagates through the tandem network because of the higher external traffic burstiness. The estimate of the end-to-end cell loss probability can be obtained from the estimates of the individual stage cell loss probabilities.

The statistical accuracy of the cell loss estimates indicates a significant robustness of the speed-up factor with respect to the buffer 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 bias parameter 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 overhead associated with the search for the near-optimal bias parameters distributed over all points and all stages reduces R_{net} by a factor of roughly 25 for system G1 and by a factor of roughly 3 for system G2. The simulation time required for 1,000 RC on a DECStation 5000/25 when no IS was applied for 1, 2 and 3-stages of systems G1 and G2 in Table 1 is given in Table 6.

We demonstrate the conservative nature of our method of computing R_{net} , due to the independence assumption between cell losses, by performing equal length simulations using $N_R = 20$ sets of $N_{RC} = 1000$ RC for both conventional MC simulations and simulation using the near-optimal bias parameters. We do this for both systems G1 and G2 for a point for which the buffer overflow event was not rare, so that the probability of cell loss and its variance could be estimated to a sufficient degree of accuracy using conventional MC simulation. We define $\tilde{R}_{net}$ as the speed-up found using the estimate of the variance resulting from the respective equal length simulations and take into account the increase in the length of the RC when IS is used:

$$\tilde{R}_{net} = \left(\frac{\hat{\sigma}_{MC}^2}{\hat{\sigma}_{IS}^2} \right) \left(\frac{N_{S,MC}}{N_S} \right) \quad (5)$$

System	K	$\hat{\sigma}_{MC}^2$	$N_{S,MC}$	$\hat{\sigma}_{IS}^2$	N_S	R_{net}	$\tilde{R}_{net}$
G1	30	5.62×10^{-7}	150	3.85×10^{-7}	322	0.42	0.45
G2	500	6.63×10^{-6}	69	2.16×10^{-8}	482	1.56	43.9

Table 7: Comparison of estimated and actual R_{net} for one stage of systems G1 and G2.

As given in Table 7, the speed-up values are comparable for system G1, but for system G2 R_{net} is a factor of roughly 25 below the value of $\tilde{R}_{net}$. This happens because as the buffer size increases, cell loss events become more bursty and hence more correlated and our independence assumption becomes more conservative. We could argue that for the longer buffer sizes the conservative nature of our method of calculating R_{net} more than compensates for the overhead incurred, thus making the R_{net} value given in the tables conservative even after factoring in the overhead.

4 Conclusion

Monte Carlo simulation using importance sampling can obtain large speed-up factors if the modification or bias of the underlying probability measures is properly chosen. In this paper, we utilized the Stochastic Gradient Descent algorithm to arrive at favorable bias parameter settings that increased the efficiency of the simulation used to estimate the cell loss probability of tandem M-IBP+MMBBP/Geo/1/ K queues. Such systems can not be solved using the methods in [5]. These queueing systems are useful building blocks in performance models for ATM switches and networks. For the examples presented, we achieved speed-up factors of 1 to 8 orders of magnitude over conventional Monte Carlo simulation of the estimation of the cell loss probability. We also described in detail the experimental method associated with the application of the SGD algorithm.

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):1464–1473, 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):293–305, June 1993.
- [8] 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*, pages 751–756, Houston, Dec. 1993.
- [9] J. A. Freebersyser, M. Devetsikiotis, and J. K. Townsend. Efficient Simulation of High-Speed Networks Using Importance Sampling and Stochastic Gradient Techniques. In *Proc. IEEE Global Telecom. Conf., GLOBECOM '94*, pages 1095–1099, San Francisco, Nov. 1994.
- [10] R. Y. Rubinstein. Sensitivity Analysis and Performance Extrapolation for Computer Simulation Models. *Operns. Res.*, 37:72–81, 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. *Operations Research*, 42(1):137–157, 1994.
- [14] M. A. Crane and A. J. Lemoine. *An Introduction to the Regenerative Method for Simulation Analysis*. Berlin: Springer-Verlag, 1977.
- [15] A. Goyal, P. Shahabuddin, 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.
- [16] F. L. Gunther and R. W. Wolff. The Almost Regenerative Method for Stochastic System Simulations. *Operations Research*, 28(2):375–386, March-April 1980.
- [17] A. M. Law and W. D. Kelton. *Simulation Modeling & Analysis*. New York: McGraw-Hill, 1991.
- [18] V. S. Frost and B. Melamed. Overview of Simulation and Traffic Modeling for Telecommunications Networks. *IEEE Comm. Magazine*, 32(3), Mar. 1994.
- [19] 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.
- [20] 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.