A Dynamic Important Methodology for the Efficient Estimation of Rare Event Probabilities in Regenerative Simulations of Queueing Systems

Michael Devetsikiotis

J. Keith Townsend

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

TR-91/14
September 1991

A Dynamic Importance Sampling Methodology for the Efficient Estimation of Rare Event Probabilities in Regenerative Simulations of Queueing Systems

Michael Devetsikiotis **J. Keith Townsend**

Center for Communications and Signal Processing,
Department of Electrical & Computer Engineering,
North Carolina State University, Raleigh, NC 27695-7914
Tel: (919)515-5200
Fax: (919)515-5523

Abstract

Importance sampling (IS) is recognized as a potentially powerful method for reducing simulation runtimes when estimating the probabilities of rare events in communication systems using Monte Carlo simulation. When simulating networks of queues, regenerative techniques must be used in order to make the application of IS feasible and efficient. The application of regenerative techniques is also crucial in obtaining correct confidence intervals for the estimates involved. However, using the most favorable IS settings very often makes the length of regeneration cycles infinite or impractically long. We present here a methodology that uses IS dynamically, within each regeneration cycle, in order to drive the system back to the regeneration state, after an accurate estimate has been obtained.

We also extend a technique we developed for finding near-optimal biasing parameters for link simulations to discrete-event simulations of queueing systems.

We demonstrate the combination of these techniques by estimating blocking probabilities for the M/M/1/K, M/D/1/K, and GI/D/1/K queues. Improvement factors of thirteen to fourteen orders of magnitude are obtained for these examples.

1 Introduction

Performance analysis of communications networks (e.g., asynchronous transfer mode, ATM) requires the estimate of cell or packet loss probabilities. The (potentially) non-Poisson statistics in these networks make pure analytical techniques difficult, and very low packet loss probabilities ($\leq 10^{-9}$) renders brute-force Monte Carlo simulation impractical. We present in this paper a methodology which uses Importance Sampling (IS) to significantly speedup simulations of queueing systems.

To obtain large improvement factors in simulation runtime using IS, the modification, or bias of the underlying probability measures must be carefully chosen, else the runtimes may increase. The most promising IS biasing schemes are parametric. In the case of bit error rate estimation, increasing the noise variance [1], and more recently, translating (shifting) the distribution of the noise probability density function (pdf) have been used [2]. The translation biasing scheme resulted in large improvement factors. Similar results have been obtained for single-sided pdf's using a scheme resembling translation or "quasi-translation" [3, 4]. In the case of queueing networks, an exponential change of measure has been shown to be optimal under certain conditions [5, 6, 7].

Analytically minimizing the variance of the importance sampling estimator with respect to the biasing parameters [2, 8], or analytically finding the optimal exponential change of measure [6, 7], has typically yielded results for systems which could be solved analytically (e.g., linear system with additive Gaussian noise or simple queues with Poisson traffic). Previously we have presented a technique for finding a near-optimal set of biasing parameters for the translation and quasi-translation biasing schemes [3, 4]. Rather than use analytical methods, the basic feature was that repetitive, very short simulation runs were used to determine near-optimal translation values. Our method exploited a theoretically justifiable relationship, for small sample sizes, between the probability estimate and the amount of translation.

There are two significant contributions in this paper. First, we extend our method of estimating optimal IS biasing parameter values presented in [3, 4] to queueing system

simulation. Second, we present a method that uses IS *dynamically* in order to allow maximum improvement while still maintaining an efficient regenerative evolution of the system. Using optimal IS when simulating queueing systems virtually requires regenerative methods. Three major advantages of using regenerative simulations are: Overcoming the deleterious effects of system memory on the efficiency of IS, no need for a warm-up period, and improved accuracy of confidence interval calculations [9].

An important issue we address in this paper is that near-optimal IS parameter settings and regenerative simulation are in conflict — near-optimal IS parameter settings typically result in impractically long or even infinitely long (for systems with infinite queueing capacity) regenerative cycles.

The idea behind *dynamic* IS is to use initially, in each regeneration cycle, the IS settings that will lead to an accurate estimate with maximum efficiency and then *change* IS values during the simulation so that the system will be driven to regeneration as quickly as possible. This change occurs after the point of diminishing returns in efficiency of the IS estimate is reached within a regenerative cycle. Thus the benefits of optimal IS and of short regeneration periods are achieved simultaneously.

In contrast, when IS is used in the customary, static way, regeneration cycles will most likely be impractically, or even infinitely long. Using static IS, the only techniques to circumvent this are either to *force* regeneration at chosen instants — which may not always be theoretically correct, or choose IS parameter values under the constraint that regeneration cycles be of manageable length — which may decrease the efficiency dramatically.

Section 2 gives a brief background of IS and regenerative techniques with respect to estimating blocking probabilities in queueing systems. Our dynamic IS methodology is presented in Section 3. Section 4 discusses the extension of our algorithm from [3, 4] to simulations of queueing systems. Section 5 presents experimental results using a combination of our dynamic IS technique and our method of finding near-optimal IS parameter values, to estimate the blocking probability of M/M/1/K, M/D/1/K, and GI/D/1/K systems. Improvement factors in runtime over conventional Monte Carlo simulation from thirteen to fourteen orders of magnitude are thus obtained. Concluding remarks are given in Section 6.

2 IS and Regenerative Simulations

2.1 Variance Reduction

Our goal is to estimate the expectation $E[Y]$ of a random variable (r.v.) Y that may in general be a function of a random vector $\mathbf{X}$, with a probability density function (pdf) $p_X(\mathbf{x})$. Then

$$E[Y] = \int y p_X(\mathbf{x})\, d\mathbf{x} \tag{1}$$

From a simulation standpoint, this would lead to the maximum likelihood empirical estimator $\widehat{E[Y]} = \frac{1}{N}\sum_{i=1}^{N} y_i$ which, depending on $p_X(\mathbf{x})$ and $Y(\mathbf{x})$, may require a very large number of samples N to yield sufficient accuracy.

Under IS, we use a modified ("biased") version, $p_X^*(\mathbf{x})$, of $p_X(\mathbf{x})$. Then, the expectation

$$E^*[Y] = \int y w(\mathbf{x}) p_X^*(\mathbf{x})\, d\mathbf{x} \tag{2}$$

is equal to $E[Y]$ when $w(\mathbf{x}) = p_X(\mathbf{x})/p_X^*(\mathbf{x})$, and $p_X^*(\mathbf{x}) \neq 0$ when $Y \cdot p_X(\mathbf{x}) \neq 0$. Therefore, $E[Y] = E^*[Y]$ can be estimated by

$$\widehat{E^*[Y]} = \frac{1}{N^*}\sum_{i=1}^{N^*} y_i\, w(\mathbf{x}_i) \tag{3}$$

where $p_X^*(\mathbf{x})$ is used in the simulation and N^* samples are observed. The potential for efficiency improvement offered by drawing samples from $p_X^*(\mathbf{x})$ is indicated by the difference between the variance σ^2 of Y (under p_X) and σ_*^2 of $Y \cdot w(\mathbf{x})$ (under p_X^*): $\Delta\sigma^2 = \sigma^2 - \sigma_*^2 = \int y^2 (1 - w(\mathbf{x})) p_X(\mathbf{x})\, d\mathbf{x}$. Clearly, if $w(\mathbf{x}) < 1$ for all $\mathbf{x}$ such that $Y(\mathbf{x}) \neq 0$ then $\Delta\sigma^2 > 0$. The key issue when using IS is the choice of $p_X^*(\mathbf{x})$ — a good choice can lead to significant improvement when compared to brute-force Monte Carlo, a poor choice can reduce the efficiency of the simulation. Ideally, when $p_X^*(\mathbf{x}) = Y(\mathbf{x})\, p_X(\mathbf{x})/E[Y]$ the variance under p_X^* becomes $\sigma_*^2 = 0$ and $\Delta\sigma^2 = \sigma^2$ [1]. However, this is merely a tautology since $E[Y]$ is the unknown quantity we need to estimate in the first place [1, 10]. Moreover, in realistic situations a closed-form expression for $Y(\mathbf{x})$ is *not* known, neither is an explicit description of the "important region", i.e., the region of the sample space that contributes to the integral in (1).

A more practical approach has been to choose $p_X^*(\mathbf{x})$ from a parametric family of distributions, usually related to the original pdf. *Variance modification* [1] and *translation* [2], as well as variations thereof [11, 3, 4] have been reported. Principles of *large deviation theory* have also been applied [5], especially to Markov process and queueing systems simulation.

When $p_X^*(\mathbf{x})$ is chosen from a parametric family of distributions, the chief issue is determining IS parameter values which yield good improvement. Analytical methods [1, 2], extrapolation techniques [12, 13], and numerical approximation [7] have been used with varying degrees of success. In this paper, we extend our statistically-based methodology [3, 4] for choosing good IS parameter settings to queueing system simulation.

2.2 Estimation of Blocking Probabilities Using Regeneration

We discuss here how blocking probabilities can be estimated using regenerative methods [9, 13]. We borrow most of the notation from [14]. The analysis that follows is applicable to a general GI/GI/1/K model.

The probability of blocking, P_B, will be defined here as the probability of an arriving customer being blocked (lost) after finding the queue already full. The empirical estimator we will use is: $\widehat{P_B} = \frac{\text{number of blocked arrivals}}{\text{total number of arrivals}}$ (i.e., "call congestion"). Let the simulated system include a single server and a single queue of length $K-1$, corresponding to a maximum number in the system of K. Assume that discrete-event simulation is being used [15] and let the events that drive the simulation belong to a finite set $\mathcal{E}$. Let $V(t_k)$ denote the type of event that occurred at instant t_k, and let $\{t_1, \ldots, t_k, \ldots, t_N\}$ be the sequence of event epochs observed in a simulation, with $V(t_k) \in \mathcal{E}, 1 \leq k \leq N$. Let the number of customers in the system just before the j-th event be X_j. Define the *arrival indicator function* I_j^A as

$$I_j^A = \begin{cases} 1, & j\text{-th event is an arrival} \\ 0, & \text{otherwise} \end{cases}$$

and the *blocked arrival indicator function* I_j^B as

$$I_j^B = \begin{cases} 1, & j\text{-th event is an arrival that is blocked, } X_j = K \\ 0, & \text{otherwise} \end{cases}$$

An event such that $I_j^B = 1$ will be called hereafter an "important event" to distinguish it from the ordinary events that drive the simulation. It is a crucial assumption that the system displays *regenerative* behavior. That is, there exists a state that is visited infinitely often, such that all new random events are scheduled *independently* of the past each time this state is visited. We can then define *stopping times* B_i and *regenerative cycles* (RC's) with length $L_i = B_i - B_{i-1}$. The important result that follows is that r.v.'s defined on disjoint RC's are statistically independent and identically distributed (i.i.d.).

Let the number of blocked arrivals, after N events have been observed, be denoted by $n_b = \sum_{j=1}^{N} I_j^B$. If we break the summation over L disjoint sets of events, the number of blocked arrivals in set i will be $n_b(i) = \sum_{j=1}^{M_i} I_{ij}^B$, where the second index, i, denotes set i, M_i is the number of events in set i and $N = \sum_{i=1}^{L} M_i$. Then $n_b = \sum_{i=1}^{L} n_b(i)$. In a similar way, we can group event epochs into disjoint sets $\{t_{11}, \ldots, t_{1M_1}\}$, $\{t_{i1}, \ldots, t_{iM_i}\}$, $\{t_{L1}, \ldots, t_{LM_L}\}$, with $t_{11} = t_1$ and $t_{LM_L} = t_N$.

In general, n_b is a function of the sequence of event epochs $\{t_1, \ldots, t_N\}$: $n_b = g(t_1, \ldots, t_N)$. Thus $E[n_b] = \int_{N-fold} \cdots \int g(t_1, \ldots, t_N)\, f_T(t_1, \ldots, t_N)\, dt_1 \ldots dt_N$, where $f_T(t_1, \ldots, t_N)$ is the joint probability distribution (pdf) of the event processes. Using our summation over the event sets $E[n_b] = \sum_{i=1}^{L} \int_{N-fold} \cdots \int\ g(t_1, \ldots, t_{iM_i})\, f_T(t_1, \ldots, t_N)\, dt_1 \ldots dt_N$, since $n_b(i)$ will, in general, be a function of all the events in its past, i.e., $n_b(i) = g(t_1, \ldots, t_{iM_i})$. Now, if we define the event sets as RC's, random variables defined in different RC's will be statistically independent, yielding $n_b(i) = g(t_{i1}, \ldots, t_{iM_i})$, $f_T(t_1, \ldots, t_N) = \prod_{i=1}^{L} f_T(t_{i1}, \ldots, t_{iM_i})$, and $E[n_b] = \sum_{i=1}^{L} E[n_b(i)] = \sum_{i=1}^{L} \int_{M_i-fold} \cdots \int g(t_{i1}, \ldots, t_{iM_i})\, f_T(t_{i1}, \ldots, t_{iM_i})\, dt_{i1} \ldots dt_{iM_i}$. Under our assumption of independent generation of intervals between events $E[n_b] = \sum_{i=1}^{L}$ $\int_{M_i-fold} \cdots \int g(t_{i1}, \ldots, t_{iM_i}) \prod_{j=1}^{M_i} f_T(t_{j1})\, dt_{i1} \ldots dt_{iM_i}$.

Similarly, the total number of arrivals can be written as $n_a = \sum_{j=1}^{N} I_j^A$, or as $n_a = \sum_{i=1}^{L} n_a(i)$, where $n_a(i) = \sum_{j=1}^{M_i} I_{ij}^A$. Since $n_a(i)$ will be a function of $\{t_{i1}, \ldots, t_{iM_i}\}$, $n_a(i) = q(t_{i1}, \ldots, t_{iM_i})$, it follows that, the expected number of arrivals will be given by $E[n_a] = \sum_{i=1}^{L} \int_{M_i-fold} \cdots \int q(t_{i1}, \ldots, t_{iM_i}) \prod_{j=1}^{M_i} f_T(t_{j1})\, dt_{i1} \ldots dt_{iM_i}$.

Using the regenerative method from [9], the blocking probability P_B can be expressed as $P_B = \frac{N_b}{N_a}$, where $N_b = E$ [number of blocked arrivals in a RC] and $N_a = E$ [total number

of arrivals in a RC]. Then an estimator for P_B is given by $\widehat{P_B} = \frac{\widehat{N_b}}{\widehat{N_a}} = \frac{\frac{1}{L_b}\sum_{i=1}^{L_b} n_b(i)}{\frac{1}{L_a}\sum_{i=1}^{L_a} n_a(i)}$, where $n_a(i)$, $n_b(i)$ are, respectively, the number of customers arrived and the number of customers blocked, during the i-th RC. The numbers L_a and L_b of RC's observed for the estimation of $\widehat{N_a}$ and $\widehat{N_b}$ can, in general, be different. An accurate estimate of N_a, can be obtained for single queue systems without using IS, since it does not involve the simulation of rare events [14]. Estimates of N_a are thus obtained in *separate* runs without IS. From now on, we restrict our attention to the efficient estimation of N_b.

Note that although $\widehat{N_a}$ and $\widehat{N_b}$ are statistically unbiased estimators of N_a and N_b, their ratio yields a *biased* estimate of P_B (in general, $E[X/Y] \neq E[X]/E[Y]$). However, $\widehat{P_B}$ is strongly consistent and asymptotically unbiased [9]. Also from [9], we can calculate asymptotically valid confidence intervals for $\widehat{P_B}$, which is one of the major motivations for using regenerative methods.

2.3 Application of IS to Regenerative Simulation

Let the pdf for the random intervals τ between events of type $e \in \mathcal{E}$ be given by $p(\tau, e)$. In applying IS, we will assume that the pdf's $p(\tau, e)$ are modified separately to become $p^*(\tau, e)$, for all $e \in \mathcal{E}$. The corresponding *weight functions* (equivalent to *likelihood ratios*) are defined as $W(\tau_{ij}, e) = \frac{p(\tau_{ij}, e)}{p^*(\tau_{ij}, e)}$ for the j-th inter-event time in the i-th RC. Using the regenerative method, the expected number of blocked arrivals under the original event processes is

$$E[n_b(i)] = \sum_{j=1}^{M_i} \int I_{ij}^B \, p(\tau_{i1}, V(i1)) \cdots p(\tau_{ij}, V(ij)) \, d\tau_{i1} \ldots d\tau_{ij} \tag{4}$$

where $V(ik)$ returns the type of the event that occurred at instant k. Under the modified processes (4) becomes

$$E[n_b(i)] = \sum_{j=1}^{M_i} \int I_{ij}^B p^*(\tau_{i1}, V(i1)) \cdots p^*(\tau_{ij}, V(ij)) \, W(\tau_{i1}, V(i1)) \cdots W(\tau_{ij}, V(ij)) \, d\tau_{i1} \ldots d\tau_{ij} \tag{5}$$

The empirical estimate for N_b resulting from (5) is

$$\widehat{N_b} = \frac{1}{L_b} \sum_{i=1}^{L_b} n_b(i) = \frac{1}{L_b} \sum_{i=1}^{L_b} \sum_{j=1}^{M_i} I_{ij}^B \, W(\tau_{i1}, V(i1)) \cdots W(\tau_{ij}, V(ij)) \tag{6}$$

Since our simulation goal is to estimate the average number of blocked arrivals in a RC, N_b, and since blocked arrivals occur rarely under $p(\tau, e)$, we should choose $p^*(\tau, e)$ such that the frequency of blocked arrivals increases. Using IS, we generate inter-event times from $p^*(\tau, e)$, and "unbias" appropriately when we compute the estimate using the weight functions.

Clearly, in (6) above, the weight function for the event epoch t_{ij} depends on all random inter-event times (e.g., interarrival or service times) previously drawn in the same RC. The memory of the system is increasing within each RC. We use regenerative techniques to avoid the deleterious effects of large system memory on the efficiency of IS.

3 A Dynamic IS Methodology: "Throttling"

3.1 Motivation

Discrete-event simulations of queueing systems can be modeled appropriately by a *generalized semi-Markov process*, as discussed in [13]. The *state* of the Markov chain involved can be defined at the instants when simulation *events* occur, that is when arrivals, service completions or other events such as state changes in an arrival process occur. Regenerative cycles (RC's) can then be defined as the periods between stopping times (instants that the system visits the regenerative state) so that IS-related r.v.'s and estimates defined in different RC's are *statistically independent.*

To illustrate, consider the following situation. Let the number of customers in the system, at instant k be denoted by X_k, and assume that a regenerative state is chosen such that $X_k = 0$. Denote the utilization factor by $\rho = \lambda/\mu$, where λ is the arrival rate and μ the service rate, and let ρ^* be the utilization factor when IS is used. For the original system (no IS), under light traffic conditions (e.g., $\rho \ll 1.0$) and with a low P_B, it is clear that the system is relatively empty most of the time, regeneration occurs rather often but blocked arrivals are rare events.

Naturally, any IS modification of the probability measures involved should tend to make "important events" (blocked arrivals) occur more often. This implies *increasing* the effective arrival rate and *decreasing* the effective service rate of the system. This immediately raises

the issue of the frequency of occurrence of regenerative states under the modified measures. We observe that, when IS settings are chosen so that the system traffic load is light ($\rho^* < 1.0$), the system still visits $X_k = 0$ although less often now, because of the higher utilization factor. On the other hand, when IS modifies the probability measures so that the system traffic load becomes excessively large ($\rho^* > 1.0$) the average length of RC's, which is customarily at least as long as the mean recurrence time of the condition $X_k = 0$, grows to an impractical size. As an example, the mean recurrence time M_0 of $X_k = 0$ would be *infinite* for practically any GI/GI/1 queue with $\rho \geq 1.0$ (unstable system). Furthermore, for the M/M/1/K system $M_0 = O(\rho^K)$, which clearly shows the exponential increase of M_0 when $\rho > 1.0$. Other queueing systems behave similarly, demonstrating the requirement for low ρ^*'s, even when "stability" is not an issue in its formal sense.

Clearly, unless restrictive assumptions on the traffic type allow regeneration to be *forced* after the first blocked arrival (as in [6]), we are required to maintain at least moderate load conditions, even under IS. This can limit dramatically the potential improvement that can be realized with IS; analytical results for simple systems have shown that the optimal biasing typically corresponds to $\rho^* > 1.0$ [6], a fact that is supported by our empirical findings.

3.2 Dynamic application of IS

To circumvent these difficulties, and combine IS and regenerative simulation we propose a technique in which IS is implemented *dynamically*. Thus, IS parameter settings are varied during each RC to initially allow important events (i.e., blocked arrivals) to occur frequently then changed in a cycle to facilitate driving the system back to regeneration. Hence the idea of "slowing down" or "throttling" the simulation after the high utilization IS settings have been used long enough within a RC to yield accurate estimates, so that the regenerative state will be reached, at which time the whole procedure will be repeated again. At the beginning of a RC, a very high utilization factor is used (the search for optimal IS values is further discussed in the next section) causing blocked arrivals to occur frequently. It will be shown below that $n_b(i)$ in eq. (6) converges within each cycle (given enough time). Thus, after a finite number of blocked arrivals have been observed (< 50), IS settings can be

changed to now favor the re-occurrence of the regenerative state. That is, once the goal of the first phase of the cycle (the phase we call "Efficient Estimation" or EE phase) has been achieved, namely to observe enough important events (blocked arrivals), the second phase (the "Accelerated Regeneration" or AR phase) regards the achievement of the regenerative state as the important event and modifies the probability measures in order to accelerate the return to such a state. In fact, as our experimental results also verify, the use of IS parameters that speed-up the return to regenerative state, e.g. $\rho^* \ll \rho$ or $\rho^* \ll 1.0$, for M/M/1/K, M/D/1/K, or GI/D/1/K queues, can induce substantial runtime savings compared with using the original, unmodified parameters in the AR phase of the RC.

3.3 Justification and Discussion

From the general IS formulation in (5) we can easily see that changing parameter values during the simulation does not violate the basic IS rules, and as long as the appropriate weight function is always used, the estimate obtained will be correct. Under the scheme described in the previous part, the empirical estimate in (6) becomes

$$\widehat{N}_b = \frac{1}{L_b} \sum_{i=1}^{L_b} \sum_{j=1}^{M_i} I_{ij}^B \prod_{k=1}^{j} W_k\left(\tau_{ik}, V(ik)\right) \tag{7}$$

where $W_k(\tau_{ik}, e) = \frac{p_k(\tau_{ik},e)}{p_k^*(\tau_{ik},e)}$ explicitly denotes the dependence of the modified pdf's (and hence the weight function) on the instant k, reflecting the dynamic variation of the IS parameters.

During the EE phase of the RC, while a high utilization factor has been artificially imposed due to IS and as the simulation progresses, the likelihood of the observed system trajectory becomes smaller and smaller with respect to the trajectory under the unmodified measures. Therefore, the weight function decreases within each RC since weights smaller than 1.0 dominate, and the effect of successive important events on the cumulative estimate decreases as well. Eventually, the weight function becomes so small that blocked arrivals contribute insignificant amounts to the summation in (7). As Glynn and Iglehart show in [13], in both the cases of a discrete-time Markov chain and the previously mentioned generalized semi-Markov process, the cumulative weight (likelihood ratio) $L_j = \prod_{k=1}^{j} W_k(\tau_{ik}, V(ik))$ goes

to zero (a.s.), as $j \to \infty$. It is straightforward to extend their approach to show that not only L_j but also $j^2 L_j$ goes to zero (a.s.). It follows then, that $\sum_{j=1}^{\infty} L_j$ *converges* (a.s.), and therefore also $\sum_{j=1}^{M_i} I_{ij}^B L_j$ converges a.s. as $M_i \to \infty$. We can, therefore choose to switch to the EE ("throttling") phase after the difference between the summation value at two successive blocked arrival instants becomes smaller than a prespecified tolerance ϵ. In practice, this usually occurred only after 10 or 20 blocked arrivals had been collected. This behavior has been consistently verified in our experimental observations.

4 Efficient IS Parameter Settings

In the context of bit error rate estimation for communication links [3, 4], we proved that when the method known as *translation* is used for biasing, and under certain conditions, *over-translation* results in *under-estimation* of the expectation in (2). Over-translation means shifting the pdf beyond the point of optimal IS improvement. In effect, we proved that, for any fixed number of samples N, the estimate in (3), denoted here as $\widehat{P(C)}$, goes to 0 almost surely (a.s.) as the translation parameter C goes to infinity. At the other end (the lower end) of the range of C, with $C = 0$ corresponding to no IS biasing, the behavior of the simulation will resemble closely that of brute-force MC simulation (i.e., no IS). Therefore, most of the time no important events will occur in this region of operation. For the range of C-values between these two extremes we note the following: As was demonstrated in [4], a robust indicator for the performance of the IS parameter settings used, can be based on the sample variance $\widehat{V(C)}$ of the important weights (i.e., those that correspond to important events) collected during the simulation. Such a performance measure is equivalent to an estimate of the *estimator variance* σ_*^2 discussed in Section 2. Furthermore, under the assumption that the variance reduction induced by IS is a well-behaved function of the parameter C, the local scatter $\widehat{S(C)}$ of the estimate in a small neighborhood $(C - \Delta C, C + \Delta C)$, compared with such scatter elsewhere, can be thought of as a measure of the performance at C. We concluded that, under certain assumptions, the estimates $\widehat{P(C)}$ and $\widehat{V(C)}$, plotted against the IS parameter values C, would appear as in Figure 1. A composite cost function (i.e.,

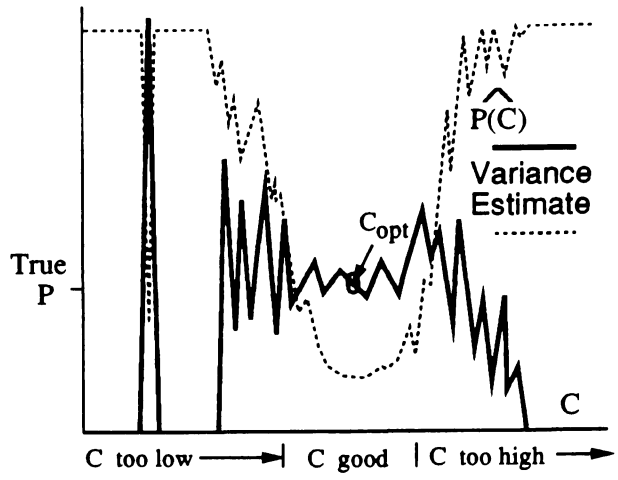

Figure 1: Probability Estimate $\widehat{P(C)}$ and estimated sample variance $\widehat{V(C)}$ vs. parameter C.

performance measure), combining the *local scatter* mentioned above and the *sample variance* estimate, was given in [4]. A near-optimal value for C can then be chosen as the value where the composite cost function is minimum.

Extensive experimental results verified our theoretical observations and indicated that large speedup factors over conventional Monte Carlo simulation could be achieved using this method. Furthermore, we applied successfully the same method for locating favorable IS settings, to a biasing scheme that resembled translation but was more appropriate for single-sided pdf's ("pure" translation applied to single-sided pdf's violates IS rules and is not permissible). This supported the expectation that our under-estimation theorem and IS optimization algorithm could be extended to other IS schemes that involved more general forms of uni-directional probability mass transfer (i.e., change of mean). Since inter-event pdf's in discrete-event simulations are single-sided (time intervals are non-negative), such an extension has particular practical significance.

Returning our attention to discrete-event simulations of queueing systems, let $p(\tau, e)$ be the pdf of the time $\mathcal{T}_e$ between events of type e. Define $\bar{\tau}_e$ as the expected value of $\mathcal{T}_e$ and $\sigma^2_{\tau_e}$ as its variance. The method of translation implies that the biased random intervals be generated as $\tau^* = \tau + C$, leading to $p^*(\tau, e) = p(\tau - C, e)$, with $\bar{\tau}^*_e = \bar{\tau}_e + C$ and $\sigma^{*2}_{\tau_e} = \sigma^2_{\tau_e}$. A second method implies that $\tau^* = \tau \cdot C$, leading to $p^*(\tau, e) = 1/|C| p(\tau/C, e)$, with $\bar{\tau}^*_e = \bar{\tau}_e \cdot C$

and $\sigma^{*2}_{\tau_e} = \sigma^2_{\tau_e} \cdot C$. In the case of the second biasing method, namely variance modification, it easily seen that there is indeed a probability mass transfer involved when $p(\tau, e)$ is single-sided and $C \geq 0$. This is indicated by the fact that the mean of the biased pdf becomes $\bar{\tau}^*_e = \bar{\tau}_e \cdot C$. Yet another biasing method is the "exponential twist" whereby $p^*(\tau, e) = e^{C\tau} p(\tau, e)/M_e(C)$, where $M_e(C)$ is the moment generating function of $\mathcal{T}_e$ calculated at C [5]. The exponential twist also generally involves probability mass transfer, and coincides with translation for a Gaussian pdf or with variance modification for an exponential pdf.

Clearly, our observations concerning the behavior of $\widehat{P(C)}$, $\widehat{V(C)}$, and $\widehat{S(C)}$ in link simulations can be directly extended to the case of queueing simulations, by letting the densities $p(\tau, e)$ or their biased counterparts replace the noise densities, and the i.i.d. estimates $n_b(i)$ in eq. (6) replace the i.i.d. observations (decisions) in link simulations. Empirical results included in Section 5 further indicate that the above described method for selection of IS settings, can be applied successfully to the simulation of queueing systems.

5 Experimental Results

In this section we use the techniques discussed earlier in this paper to estimate the average probability of blocked arrival for M/M/1/K, M/D/1/K, and GI/D/1/K systems.

The M/M/1/K system had an average arrival rate $\lambda = 1.0$, an average service rate $\mu = 1.333$, and a system capacity $K = 101$. The probability of an arrival being blocked could be calculated analytically and was found to be 6.01×10^{-14}. For this system RC's coincided with *busy cycles*. The average number of arrivals per RC was calculated analytically to be $N_a = 4.0$. As discussed earlier, we only needed to use IS to estimate the average number blocked per RC, N_b. For this example, $N_b = 2.41 \times 10^{-13}$, and this is the number that we estimated using our technique.

Under IS, the inter-arrival and service times were still exponential, but the rates λ and μ were independently multiplied by biasing parameters. Figure 2 shows a 3-dimensional plot of the average blocked per RC for the M/M/1/K system. Each point on this plot represents one simulation run of 100 RC's. The independent axes of the plot are the settings for the two

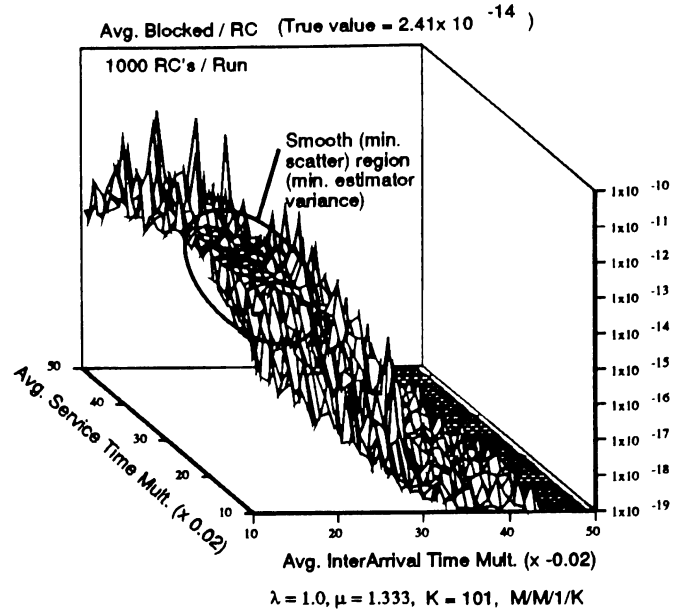

Figure 2: Plot of average number blocked per RC for an M/M/1/K queueing system as a function of the IS parameters, namely average service time multiplier and average interarrival time multiplier.

IS parameters, namely service time multiplier, and interarrival time multiplier. The point (0, 0) (front corner) corresponds to brute-force Monte Carlo. Note that this part of the plot is not shown to provide a better view of the smooth region.

The opposite ends of the axes (near the 50) correspond to over-biasing of the probability densities. Note that there is under-estimation of the average blocked per RC in the region near these two extremes. The region of minimal local scatter (circled) corresponds to the set of IS parameter settings which yield near-optimal improvement. The local smoothness of this plot is indicative of small estimator variance.

Using our algorithm in [4], optimal parameter values are estimated to be 0.73 and 1.36 for the interarrival time multiplier and service time multipliers respectively. Shown in Figures 3 and 4 are cuts of the 3-D figure through this optimal point. Also shown in these figures are plots of the corresponding variance estimators. As discussed in [4], the algorithm minimizes a cost function which combines the local scatter of the (estimated) average-blocked-per-RC curve and the sample variance for each estimate. The effectiveness of the algorithm is clearly evident in the figures.

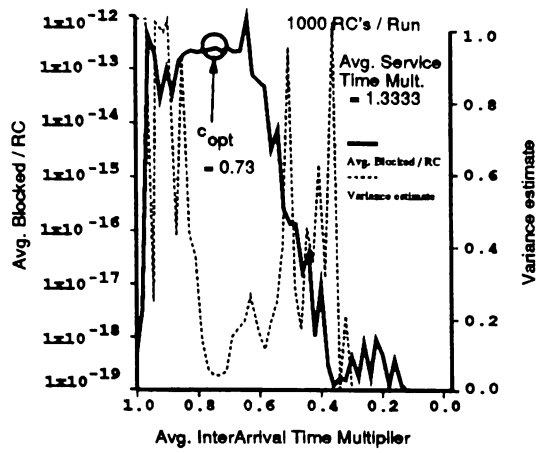

Figure 3: Cut of the 3-D plot of average blocked per RC overlaid with estimated sample variance, as a function of the interarrival time multiplier at the near-optimal value for the service time multiplier.

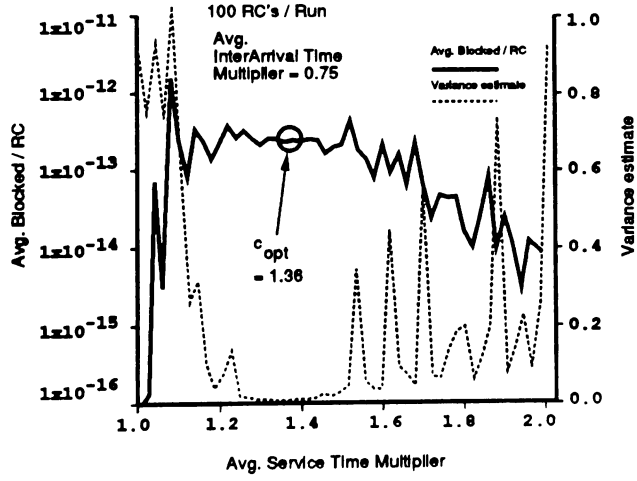

Figure 4: Cut of the 3-D plot of average blocked per RC overlaid with estimated sample variance, as a function of the service time multiplier at the near-optimal value for the interarrival time multiplier.

These optimal IS parameter values were used in a set of longer simulation runs to estimate the improvement over brute-force Monte Carlo simulation. For 100 runs with 1000 RC's per run, the IS simulation estimate for $N_b = 2.4 \times 10^{-13}$. The 95% confidence coefficient for this set of runs (1000 RC's per estimate) was calculated to be 2.89×10^{-15}.

To estimate the improvement factor in simulation run time over brute-force MC, we used the results in [16] to calculate the number of RC's required for a brute-force run to yield the same confidence coefficient. Using this procedure we calculated an improvement factor of 4×10^{13}, i.e., our simulation estimated the average number blocked in a RC with a factor of 4×10^{13} fewer RC's than would have been required by brute-force Monte Carlo simulation.

Note that this measure of comparison is based on the *number* of RC's, not the actual simulation time that would have to include the *length* of RC's as well. Assuming the computational effort required to complete the simulation of the i-th RC to be equal to its length L_i, the total runtime required to obtain an estimate based on N RC's is $L = \sum_{i=1}^{N} L_i$. Then, $\bar{L} = \mathrm{E}\{L\} = N\,\mathrm{E}\{L_i\}$. A fair comparison of simulation efficiency can then be based on the *time-reliability product* $\bar{L}\,\sigma_*^2$, where σ_*^2 is the variance of $n_b(i)$, the number of blocked customers during the i-th RC. In the case of brute-force MC, since the queue rarely fills up, RC's are extremely short at the cost of a very large σ^2. Under favorable IS settings, the length of RC's increases but σ_*^2 decreases so dramatically that the resulting time-reliability product is orders of magnitude smaller than that of brute-force MC. The choice of "throttling" parameters should make $\mathrm{E}\{L_i\}$ as short as possible, without increasing σ_*^2. Hence, while the purpose of the EE phase is mainly to make σ_*^2 low, the AR phase ensures that regeneration will occur and keeps L_i's short.

To see the effect of "throttling" on the simulation length, refer to Figure 5. In this figure, which also corresponds to the same M/M/1/K system above, note that the leftmost point corresponds to brute-force MC simulation. As we increase the amount of "throttling", the RC length is shorter, as would be expected. The values of interarrival time multiplier and service time multiplier shown on the rightmost point are the values used in all simulation results reported in this section. The advantage of throttling is most evident when the system

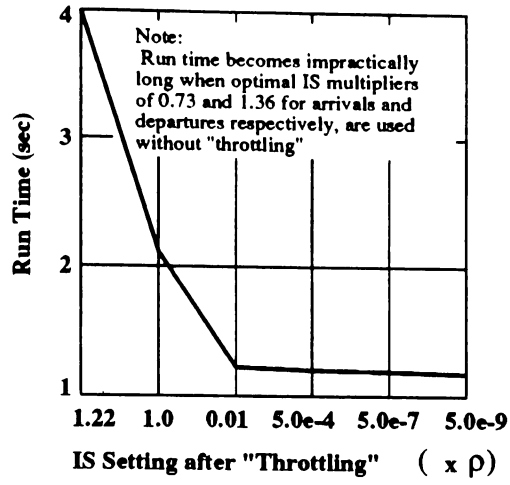

Figure 5: Plot of the average simulation run time in seconds versus the "throttling" (AR) parameters expressed as a fraction of the original utilization ρ. Runtimes are based on starting with the optimal IS settings and "throttling" after 20 lost customers, 100 RC's per run.

is biased for maximum improvement (i.e., the point (0.73, 1.36) from above). In this case, RC's would be impractically long, as discussed earlier.

One other characteristic alluded to earlier was that as the trajectory in an RC evolves under IS, $\sum_{j=1}^{M_i} I_{ij}^{B} \prod_{k=1}^{j} W_k(\tau_{ik}, V(ik))$ *converges* (a.s.) as M_i increases. This can be seen in Figure 6, which shows the estimated number blocked per RC for the same M/M/1/K system above as a function of the number of blocked arrivals observed in one RC before throttling. After 10 to 20 blocked arrivals have been observed in an RC, the estimate has converged to a desired level of precision.

The next example is an M/D/1/K queue, where $\lambda = 1.0$, the deterministic service rate is fixed at 1.333, and the system capacity was $K = 59$. Again for this system, RC's coincided with busy cycles. Under IS, the inter-arrival times were still exponential, but the rate λ was multiplied by a biasing parameter. A plot of the average blocked per RC and the sample variance estimate as a function of the interarrival time multiplier is shown in Figure 7. As before, the optimal IS biasing parameter value was estimated using our algorithm. It is clear from Fig. 7 that the chosen value $c_{opt} = 0.55$ corresponds to the minimum scatter and minimum variance point. Repeating the same procedure as above in the M/M/1/K

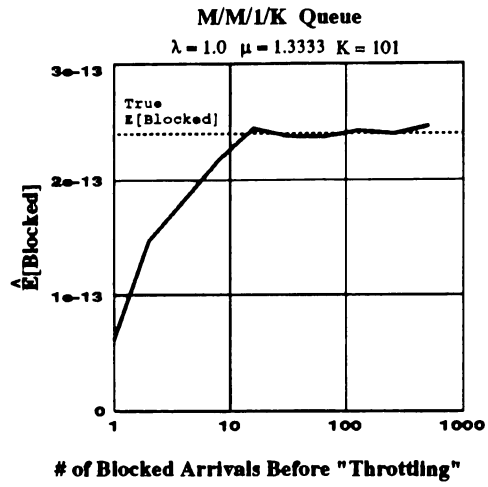

Figure 6: Plot of the estimated number blocked per RC as a function of the number of blocked arrivals observed before "throttling".

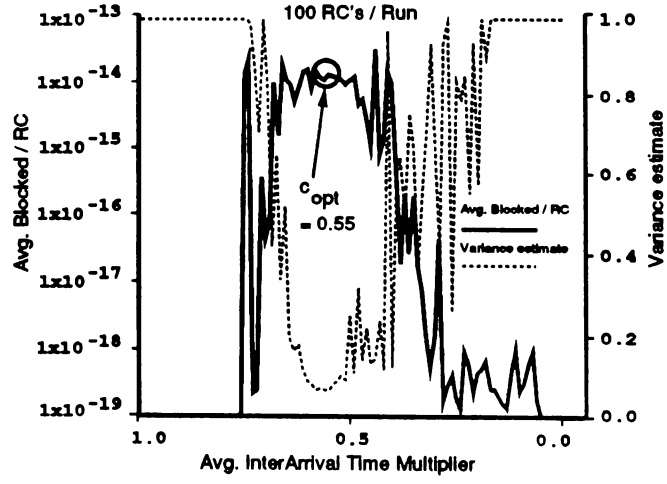

Figure 7: Plot of estimated average number lost per RC overlaid with estimated sample variance, as a function of the interarrival time multiplier for an M/D/1/K system.

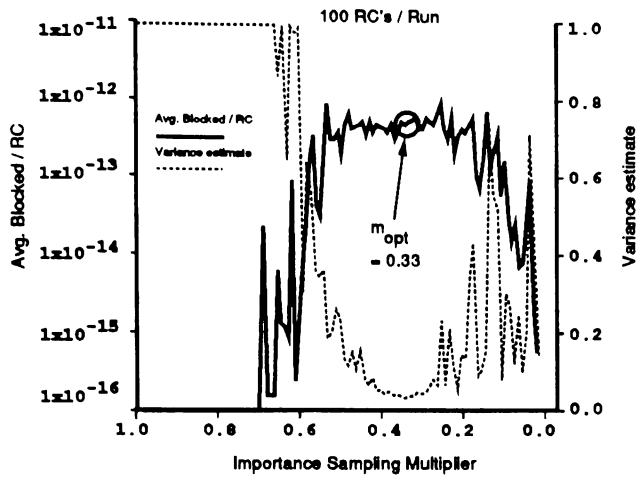

Figure 8: Plot of estimated average number lost per RC overlaid with estimated sample variance, as a function of the multiplier m for an GI/D/1/K system.

case, we obtained an estimate of the average blocked per RC $\hat{N}_b = 1.14 \times 10^{-14}$. The 95% confidence coefficient for 100 runs at 1000 RC's per run was 4.46×10^{-16}. The improvement over brute-force MC simulation was found to be 1.75×10^{14}.

The last example we present is an GI/D/1/K queue, where Pr[interarrival time $= \alpha_1$] $= p$, and Pr[interarrival time $= \alpha_2$] $= 1 - p$, $0 < p < 1$. In this example, $\alpha_1 = 2.1$, $\alpha_2 = 0.7$, $p = 0.6$, the deterministic service rate is fixed at 1.25, and the system capacity is $K = 19$. RC's coincided with busy cycles.

Under IS, p was multiplied by an IS multiplier m to obtain the biased distribution Pr*[interarrival time $= \alpha_1$] $= p^* = pm$, and Pr*[interarrival time $= \alpha_2$] $= 1 - pm$, $0 < m < 1/p$. A plot of the average blocked per RC and the sample variance estimate as a function of the multiplier m is shown in Figure 8. The optimal IS biasing parameter setting was found to be $m_{opt} = 0.33$. The estimate of the average blocked per RC, 95% confidence coefficient, and improvement over brute-force MC simulation were found to be $\hat{N}_b = 3.88 \times 10^{-13}$, 2.68×10^{-15}, and 2.1×10^{14}, respectively.

6 Conclusions

We have presented a methodology that uses IS dynamically, within each regeneration cycle, in order to drive the system back to the regeneration state, after an accurate estimate has been obtained. Using this methodology, the benefits of optimal IS and of short regeneration periods can be achieved simultaneously.

We also extended a technique we developed for finding near-optimal biasing parameters for link simulations to discrete-event simulations of queueing systems.

We demonstrated the combination of these techniques by estimating blocking probabilities for the M/M/1/K, M/D/1/K, and GI/D/1/K queues. Improvement factors of thirteen to fourteen orders of magnitude were obtained for these examples.

References

[1] K. S. Shanmugan and P. Balaban. A Modified Monte-Carlo Simulation Technique for the Evaluation of Error Rate in Digital Communication Systems. *IEEE Transactions on Communications*, COM-28(11):1916–1924, November 1980.

[2] D. Lu and K. Yao. Improved Importance Sampling Technique for Efficient Simulation of Digital Communication Systems. *IEEE Journal on Selected Areas in Communications*, 6(1), January 1988.

[3] M. Devetsikiotis and J. K. Townsend. A Useful and General Technique for Improving the Efficiency of Monte Carlo Simulation of Digital Communication Systems. In *Proceedings of GLOBECOM '90, San Diego, CA*, 1990.

[4] M. Devetsikiotis and J. K. Townsend. An Algorithmic Approach to the Optimization of Importance Sampling Parameters in Digital Communication System Simulation. Submitted for publication in the *IEEE Transactions on Communications*.

[5] M. Cottrell, J.-C. Fort, and G. Malgouyres. Large Deviations and Rare Events in the Study of Stochastic Algorithms. *IEEE Transactions on Automatic Control*, AC-28:907–920, September 1983.

[6] S. Parekh and J. Walrand. A Quick Simulation Method for Excessive Backlogs in Networks of Queues. *IEEE Transactions on Automatic Control*, AC-34(1):54–66, January 1989.

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

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

[9] M. A. Crane and A. J. Lemoine. *An Introduction to the Regenerative Method for Simulation Analysis.* Berlin: Springer-Verlag, 1977.

[10] Q. Wang and V. K. Bhargava. On the Application of Importance Sampling to BER Estimation in the Simulation of Digital Communication Systems. *IEEE Transactions on Communications*, COM-35(11):1231–1233, November 1987.

[11] H. J. Schlebusch. Nonlinear Importance Sampling Techniques for Efficient Simulation of Communication Systems. In *Proceedings of IEEE International Communications Conference, ICC '90*, 1990.

[12] W. W. LaRue and V. S. Frost. A Technique for Extrapolating the End-to-End Performance of HDLC Links for a Range of Lost Packet Rates. *IEEE Transactions on Communications*, COM-38(4):461–466, April 1990.

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

[14] V. S. Frost and Q. Wang. Efficient Estimation of Cell Blocking Probability for ATM Systems. In *Proceedings of IEEE International Communications Conference, ICC '91*, 1991.

[15] A. M. Law and W. D. Kelton. *Simulation Modeling & Analysis*. New York: McGraw-Hill, 1991.

[16] M. C. Jeruchim. Techniques for Estimating the Bit Error Rate in the Simulation of Digital Communication Systems. *IEEE Journal on Selected Areas in Communications*, SAC-2(1):153–170, January 1984.