Inverse Convolution Approach to Importance Sampling in Monte Carlo Simulation of Loss Systems

P. E. Lassila and J. T. Virtamo
Laboratory of Telecommunications Technology
Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
Email: {Pasi.Lassila, Jorma.Virtamo}@hut.fi

July 2000

Abstract

In this paper we consider the problem of estimating blocking probabilities in the multiservice loss system via simulation, applying the static Monte Carlo method with importance sampling. Earlier approaches to this problem include the use of either a single exponentially twisted version of the steady state distribution of the system or a composite of individual exponentially twisted distributions. Here, a different approach is introduced, where the original estimation problem is first decomposed into independent simpler sub-problems, each roughly corresponding to estimating the blocking probability contribution from a single link. An importance sampling distribution is presented, which very closely approximates the ideal importance sampling distribution for each sub-problem. The distribution is a conditional distribution and the samples generated from it are directly in the blocking state region in each sub-problem. We show how samples from this conditional distribution can be generated effectively by the so called inverse convolution method. Finally, a dynamic control algorithm is given for optimally allocating the samples between different sub-problems. The numerical results demonstrate that the variance reduction obtained with the inverse convolution method is truly remarkable, between 670 and 1 000 000 in the considered examples.

1 Introduction

Modern broadband networks have been designed to integrate several service types into the same network. On the call scale, the process describing the number of calls present in the network can be modeled by a loss system, see e.g. [3]. One of the basic tasks is to calculate the steady state blocking probability for each traffic class in the system. The steady state distribution of the system is of the well known product form, from which it is easy to write down analytic expressions also for the blocking probabilities. A problem with the exact solution, however, is that it cannot be computed for realistic size networks due to the prohibitive size of the state space. Recursive methods can be used to alleviate the problem, but they are applicable only in the case of a small number of links.

In such a situation there are two alternatives: to use analytical approximations or to simulate the problem to a desired level of accuracy. In this paper we will be dealing with the latter approach. Then, as the form of the stationary distribution is known, the static Monte Carlo (MC) method can be used to perform the simulation. In order to make the simulation more efficient, it is possible to use importance sampling (IS), where one uses an alternative sampling distribution, which makes the interesting samples more likely than under the original distribution. The twist in the distribution is then corrected for by weighting the samples with the so called likelihood ratio.

In this paper an efficient IS distribution is derived aiming at approximating the properties of the ideal IS distribution as closely as possible. Previous work on estimating the blocking probabilities via the static Monte Carlo method includes the works of Ross [3, chap. 6] and Mandjes [2]. Ross has presented heuristics which attempt to increase the likelihood of the blocking states while, at the same time, trying to limit the likelihood of generating misses from the allowed state space, resulting in a rather conservative twist. Mandjes has proposed to use an importance sampling distribution which is an exponentially twisted version of the stationary distribution of the system that shifts the mean of the sampling distribution to match the most probable blocking state in the network. In [1], we presented an approach based on using a similar technique with exponentially twisted distributions, but we extended the approach with ideas suggested by the large deviation results obtained by Sadowsky and Bucklew in [4]. They have shown that for estimating the probability of sets having a similar shape as the set of the blocking states, the asymptotically optimal IS distribution is of a composite form.

Here a slightly different approach is adopted. The basic idea is the same as in [1], to effectively sample the blocking states lying on the boundary of each active link constraint. Instead of using a composite form distribution for this, the problem is first decomposed into separate sub-problems. The decomposition corresponds to breaking the blocking probability down to components each of which essentially gives the blocking probability contribution from a single link. Then an effective IS method to solve each subproblem is given. In this method the earlier used exponentially twisted distributions are replaced with a more accurate approximation of the ideal IS distribution. The idea is to generate samples directly into the set of blocking states of a given link, assuming solely that link to have a finite capacity. This is achieved by using a certain conditional distribution as the IS distribution. Samples from this distribution can be generated by a method we call the inverse convolution method. The method drastically improves the performance of the IS sampling. In the examples considered, the reduction of the standard deviation obtained by the inverse convolution method varied from 26 to 1000, using the direct Monte Carlo method as a reference. In terms of the required number of samples for a given accuracy this translates to a reduction by a factor of the order from 670 to 1 000 000.

The paper is organized as follows. Section 2 presents briefly the multiservice loss system. The simulation of the blocking probabilities and the IS method together with the properties of a proper IS distribution for estimating the blocking probabilities are discussed in section 3. Sections 4 contain the main results of the paper and describes the inverse convolution method. In section 5 we describe the dynamic method for optimally allocating the number of samples to be used for each sub-problem and give some numerical examples demonstrating the effectiveness of the two methods. Section 6 contains our conclusions.

2 The multiservice loss system

Consider a network consisting of J links, indexed with $j = 1, \ldots, J$, link j having a capacity of C_j resource units. The network supports K classes of calls. Associated with a class-k call, $k = 1, \ldots, K$, is an offered load ρ_k and a bandwidth requirement of b_k^j units on link j. Note that $b_k^j = 0$ when class-k call does not use link j. Let the vector $\mathbf{b}^j = (b_1^j, \ldots, b_K^j)$ denote the required bandwidths of different classes on link j. Further, we denote by $\mathcal{R}_k$ the set of links used by the traffic class k, i.e.

$$\mathcal{R}_k = \{j \in \mathcal{J} \mid b_k^j > 0\},$$

where $\mathcal{J} = \{1, 2, \ldots, J\}$ denotes the set of link indexes.

We assume that the calls in each class arrive according to a Poisson process, a call is always accepted if there is enough capacity available, and that the blocked calls are cleared. Let $\mathbf{X} = (X_1, \ldots, X_K)$ denote the state of the system, with X_k giving the number of class-k calls in progress. Consider first the case where the capacities of the links are infinite. The system behaves as K independent Poisson processes. The state space is then

$$\mathcal{I} = \{\mathbf{x} \mid \mathbf{x} \geq \mathbf{0}\},$$

where $x_k \in \mathbb{N}$ with $\mathbb{N}$ denoting the set of natural numbers $\{0, 1, 2, \ldots\}$. The steady state distribution, P, of $\mathbf{X}$ is of the product form

$$f(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x}\} = \prod_{k=1}^{K} f_k(x_k), \quad \mathbf{x} \in \mathcal{I}, \tag{1}$$

where $f_k(x) = (\rho_k^x / x!)\, e^{-\rho_k}$ is the one-dimensional Poisson distribution.

Also, let Y_k^j denote the random variable for the occupancy of link j due to the traffic of class k, i.e $Y_k^j = b_k^j X_k$. The distribution of Y_k^j is then

$$m_k^j(y) = \mathrm{P}\{Y_k^j = y\} = \begin{cases} f_k(x), & \exists\, x \in \mathbb{N} : y = b_k^j x, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

For the finite capacity system, the set of allowed states, $\mathcal{S}$, can be described as

$$\mathcal{S} = \left\{\mathbf{x} \in \mathcal{I} \mid \forall\, j : \mathbf{b}^j \cdot \mathbf{x} \leq C_j \right\},$$

where the scalar product is defined as $\mathbf{b}^j \cdot \mathbf{x} = \sum_k b_k^j x_k$. The steady state distribution, π, is given by the truncation of (1) to the allowed state space, $\mathcal{S}$,

$$\pi(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{S}\} = \begin{cases} \dfrac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}}, & \mathbf{x} \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases}$$

The set of blocking states for a class-k call, $\mathcal{B}_k$, is

$$\mathcal{B}_k = \left\{ \mathbf{x} \in \mathcal{S} \mid \exists\, j : \mathbf{b}^j \cdot (\mathbf{x} + \mathbf{e}_k) > C_j \right\},$$

where $\mathbf{e}_k$ is a K-component vector with 1 in the k^{th} component and zeros elsewhere. The blocking probability of a class-k call, B_k, can then be expressed in the form of a ratio of two state sums, denoted by β_k and γ,

$$B_k = \sum_{\mathbf{x} \in \mathcal{B}_k} \pi(\mathbf{x}) = \frac{\sum_{\mathbf{x} \in \mathcal{B}_k} f(\mathbf{x})}{\sum_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x})} = \frac{\mathrm{P}\{\mathbf{X} \in \mathcal{B}_k\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}} = \frac{\beta_k}{\gamma}. \tag{3}$$

We can note here that, instead of having the state space $\mathcal{I}$ for $\mathbf{X}$, we could consider any Cartesian product space enclosing $\mathcal{S}$.

For later purposes, we introduce the set $\mathcal{D}_k^j$ of blocking states for link j,

$$\mathcal{D}_k^j = \left\{ \mathbf{x} \in \mathcal{I} \mid C_j - b_k^j < \mathbf{b}^j \cdot \mathbf{x} \le C_j \right\}.$$

Thus the set $\mathcal{D}_k^j$ consists of the blocking states in a system where only link j has a finite capacity and all other links have an infinite capacity.

3 Decomposition and importance sampling for loss systems

In what follows we discuss the estimation of the blocking probabilities via the importance sampling simulation method. As the form of the stationary distribution $f(\mathbf{x})$ is known, a natural choice for the simulation method is the static Monte Carlo method. The main problem in the simulation is to quickly get a good estimate for β_k, i.e. the numerator in (3), especially in the case, when the B_k are very small. For completeness, we note that the blocking probability B_k does not only depend on β_k, but also on the state sum γ given by the denominator of (3). The direct Monte Carlo method for estimating γ corresponding to the probability $\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}$. This probability is usually close to 1 and is therefore easy to estimate using the standard MC method. Therefore, in the rest of this paper we concentrate on efficient methods for estimating β_k. Furthermore, in the following, we suppress from the notation the index k of the class for which the state sum β_k (and the blocking probability) is to be estimated.

In our case, we can apply IS very effectively by first decomposing the problem into independent sub-problems and then using IS for each subproblem. The decomposition is based on the following observation. The set of blocking states (for traffic class k) can be expressed as

$$\mathcal{B} = \mathcal{S} \cap \bigcup_{j \in \mathcal{R}} \mathcal{D}^j .$$

This is illustrated in Figure 1 on the left hand side, which shows a two traffic class example with three links. The grey areas represent the blocking state regions $\mathcal{D}^j$ of some traffic class for each link. The whole set of blocking states $\mathcal{B}$ is then the area between the continuous black lines. Now, β is an expectation of the form $\mathrm{E}[h(\mathbf{X})]$ where $h(\mathbf{x}) = 1_{\mathbf{x} \in \mathcal{B}}$. Thus $h(\mathbf{x})$ can be decomposed as

$$h(\mathbf{x}) = 1_{\mathbf{x} \in \mathcal{B}} = \sum_{j \in \mathcal{R}} \frac{1}{\nu(\mathbf{x})} 1_{\mathbf{x} \in \mathcal{S}} \, 1_{\mathbf{x} \in \mathcal{D}^j} ,$$

where $\nu(\mathbf{x})$ is a function giving the number of sets $\mathcal{D}^j$ that the point $\mathbf{x}$ belongs to, i.e. it takes care of weighting those points appropriately that lie in the intersection of two or more $\mathcal{D}^j$ sets. Then, also the computation of the original expectation decomposes into independent sub-problems, i.e.

$$\mathrm{E}[h(\mathbf{X})] = \sum_{j \in \mathcal{R}} \frac{1}{\nu(\mathbf{X})} 1_{\mathbf{X} \in \mathcal{S}} \, 1_{\mathbf{X} \in \mathcal{D}^j} .$$

Now, let $h^j(\mathbf{x}) = 1_{\mathbf{x} \in \mathcal{S}} \, 1_{\mathbf{x} \in \mathcal{D}^j} / \nu(\mathbf{x})$. The value of one of the $h^j(\cdot)$ functions is illustrated in Figure 1 on the right hand side. Note that with slight modification we could also decompose the set $\mathcal{B}$ into non-overlapping regions, whence there would not appear any $1/\nu(\mathbf{x})$ term in the $h^j(\mathbf{x})$ function.

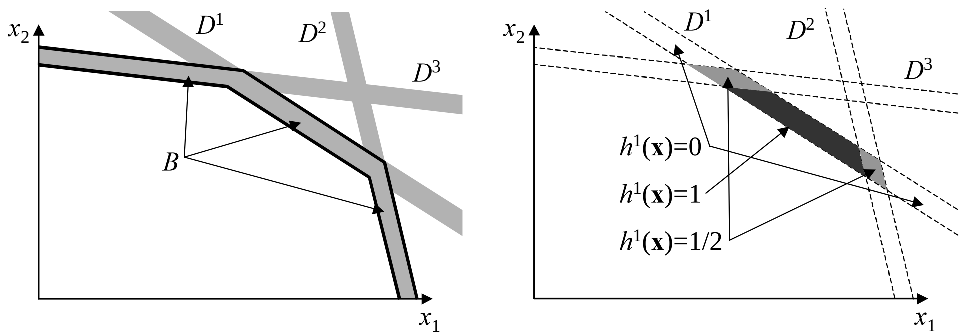

Figure 1: Decomposition of the set $\mathcal{B}$ into three subsets in a network with two traffic classes and three link constraints (left figure) and the values of one of the $h^j(\cdot)$ functions in different parts of $\mathcal{D}^j$ (right figure).

To estimate each $\eta^j = \mathrm{E}[h^j(\mathbf{X})]$ efficiently we apply importance sampling. To this end, let $\mathbf{X}^* \in \mathcal{I}$ be another random variable with distribution $p_j^*(\mathbf{x})$ and let $w(\mathbf{x}) = f(\mathbf{x})/p_j^*(\mathbf{x})$ denote the so called likelihood ratio. As is well known, the general

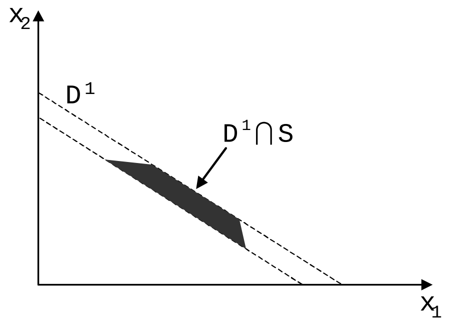

Figure 2: Estimation of η^j.

idea in IS is to find a distribution $p_j^*(\mathbf{x})$ such that the variance of the IS estimator,

$$\widehat{\eta}^j = \frac{1}{N_j} \sum_{n=1}^{N_j} \frac{1}{\nu(\mathbf{X}_n^*)} 1_{\mathbf{X}_n^* \in \mathcal{S}} \, 1_{\mathbf{X}_n^* \in \mathcal{D}^j} \, w(\mathbf{X}_n^*), \tag{4}$$

is minimized (N_j denotes the number of samples). Obviously, in this case, the ideal IS distribution would always generate points that lie in $\mathcal{D}^j$ and are always inside the allowed state space $\mathcal{S}$, i.e. points that are in $\mathcal{B}$, with a distribution proportional to $f(\mathbf{x})/\nu(\mathbf{x})$. Consequently, the value of the observed variable $w(\cdot)/\nu(\cdot)$ would be a constant. This conditional distribution is unrealizable but we approximate it by another conditional distribution

$$p_j^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}^j\} = \begin{cases} \dfrac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{D}^j\}} = \dfrac{f(\mathbf{x})}{v_j}, & \mathbf{x} \in \mathcal{D}^j, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where v_j is the probability mass of the set $\mathcal{D}^j$. This is illustrated in Figure 2. With IS distribution (5) we are generating points in the set $\mathcal{D}^j$ (area between the dashed lines) and simulation is needed essentially only to determine which part of the probability mass of $\mathcal{D}$ is actually inside $\mathcal{S}$ (factor $1_{\mathbf{X}^* \in \mathcal{S}}$, grey area in the figure). Additionally we have the factor $1/\nu(\mathbf{X}^*)$ to compensate for double (or multiple) counting for such points $\mathbf{x}$ that belong to more than one of the sets $\mathcal{D}^j$. However, the effect of this is of minor importance as, in practice, most of the points belong to only one set $\mathcal{D}^j$.

The efficiency gain obtained with the above can be shown as follows (where, for clarity, we omit the effect of the factor $1/\nu(\cdot)$). When using the original distribution $f(\cdot)$ as the IS distribution in (4), we need to estimate the probability $p = \mathrm{E}[1_{\mathbf{X} \in \mathcal{S}} \, 1_{\mathbf{X} \in \mathcal{D}^j}]$. Then each sample generated from $f(\cdot)$ is an independent Bernoulli variable and the relative error (or relative deviation) of the estimate, given by the ratio of the standard deviation and the mean of the estimate, after N samples have drawn is $\sqrt{(1-p)/(pN)}$. Thus, assuming for instance $p = 0.005$, we need almost 80 000 samples to have a relative error of 5% for the estimate. On the other hand, when using (5) as our IS distribution, we only need to estimate the conditional probability $p' = \mathrm{E}\left[1_{\mathbf{X}\in\mathcal{S}} \,|\, \mathbf{X} \in \mathcal{D}^j\right]$. This probability is typically much greater than p. Thus, assuming for example that $p' = 0.9$ we only need about 45 samples to reach the same 5% relative error level, giving us a decrease by a factor of almost 2000 in the required sample size. Our numerical experiments verify that efficiency gains of this order or even greater can be obtained in practice, as well.

When using (5) as our IS distribution, the likelihood ratio is a constant,

$$w(\mathbf{x}) = \frac{f(\mathbf{x})}{f(\mathbf{x})/v_j} = v_j,$$

and the estimator for η^j becomes

$$\widehat{\eta}^j \quad = \quad \frac{v_j}{N} \sum_{n=1}^{N} \frac{1}{\nu(\mathbf{X}_n^*)} \, 1_{\mathbf{X}_n^* \in \mathcal{S}}.$$

Then the final estimator for β is simply

$$\widehat{\beta} = \sum_{j\in\mathcal{R}} \widehat{\eta}^j.$$

Now, given the total number of samples N to be used for the estimator, the number of samples N_j allocated to each subproblem is a free parameter. In section 5 we show how to choose each N_j to minimize the variance of $\widehat{\beta}$.

In the next section we describe how the probability $\mathrm{P}\{\mathbf{X} \in \mathcal{D}^j\}$ of our conditioning event is computed, and how samples can be generated directly into the set $\mathcal{D}^j$.

4 The inverse convolution method

As we are now only considering the estimation of η^j for a fixed $j \in \mathcal{R}$ we omit the link index j from the notation. This implies that C_j, b_k^j and $\mathcal{D}_k^j$ are denoted here by C, b and $\mathcal{D}$, respectively (remember that dependence on the traffic class k being under inspection was suppressed earlier). To further simplify the notation, we also assume, without loss of generality, that the traffic classes which use link j have the indexes $1, \ldots, L$. The following method is based on the observation that it is relatively easy to generate points into the set $\mathcal{D}$ from the conditional IS distribution (5), i.e. $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$, by reversing the steps used to calculate the occupancy distribution of the considered link by convolutions.

Recall that the occupancy due to the traffic of class-k calls on the link under consideration is denoted by Y_k with the distribution $m_k(\cdot)$ as defined in (2). Let S_l, with $l = 1, \ldots, L$, denote the occupancy distribution on the considered link caused by the superposition of the first l classes, i.e.

$$S_l = \sum_{l'\leq l} Y_{l'}, \quad l = 1, \ldots, L.$$

We can also express $S_l = S_{l-1} + Y_l$, where both S_{l-1} and Y_l are independent. The distribution of S_l, $q_l(x) = \mathrm{P}\{S_l = x\}$, can be obtained recursively from the convolution

$$q_l(x) = \sum_{y=0}^{x} q_{l-1}(x-y)m_l(y). \tag{6}$$

Thus we also obtain the probability mass of the set $\mathcal{D}$, v, from

$$v = \mathrm{P}\{\mathbf{X} \in \mathcal{D}\} = \mathrm{P}\{C - b < S_L \leq C\} = \sum_{i=C-b+1}^{C} q_L(i).$$

The event $S_l = x$ is the union of the events $\{Y_l = y, S_{l-1} = x-y\}$, $y = 0, \dots, x$ with the probabilities $m_l(y)q_{l-1}(x-y)$. Conversely, given $S_l = x$, the conditional probability of the event $Y_l = y$ is $m_l(y)q_{l-1}(x-y)/q_l(x)$, for $y = 0, 1, \dots, x$. These probabilities can be precomputed and stored. Then, given $S_l = x$, using these probabilities it is easy to draw a value, say y, for Y_l and consequently for $S_{l-1} = x - y$. In fact, it is advantageous to store directly the values of the cdf

$$\mathrm{P}\{Y_l \leq y \,|\, S_l = x\} = \sum_{y'=0}^{y} m_l(y')q_{l-1}(x-y')/q_l(x). \tag{7}$$

Then the value of $Y_l \leq y$ can be drawn by finding the smallest y such that $\mathrm{P}\{Y_l \leq y \,|\, S_l = x\} \geq U$, where U is a random variable drawn from the uniform distribution in $(0, 1)$.

Now, S_L is the occupancy of the link, and the set $\mathcal{D}$ corresponds to $C - b < S_L \leq C$. A point in $\mathcal{D}$ can be generated by first drawing a value for S_L using the distribution $q_L(\cdot)$ conditioned on $C - b < S_L \leq C$, which is also precomputed and stored. This is shown in Figure 3 on the left hand side. Then, as described above, (Y_L, S_{L-1}) can be drawn. This is shown in Figure 3 in the middle. Once the value of S_{L-1} is fixed, we can draw (Y_{L-1}, S_{L-2}). This process is continued until the value of the last component Y_1 has been drawn. The most important thing here is to note that the distributions of the conditional sets (Y_l, S_{l-1}) for a fixed value of S_l^j are easily precomputed and, hence, each component Y_l is generated as an outcome from a simple table lookup. The other classes not using the link, i.e. classes $L + 1$ to K, are independent from classes $1, \dots, L$ and from each other. Hence, their values are drawn independently from the distributions $f_k(\cdot)$, $k = L + 1, \dots, K$.

The generation of samples is as fast as in a standard MC method, once the conditional distributions have been computed. Furthermore, the memory requirements of the algorithm, i.e. the number of elements in the arrays, are not prohibitive. The number of array elements to be stored can be seen to be $\frac{1}{2}KC(C + 1)$. It should be noted that the dependence on K is only linear whereas the size of the state space grows exponentially with K. However, if this memory requirement grows too large, the minimum requirement is that the q_l and m_l distributions have been precomputed. Then the conditional distribution $\mathrm{P}\{Y_l \leq y \,|\, S_l = x\}$, given by (7), must be constructed on the fly, making the sample generation somewhat slower.

In summary the procedure for generating samples from (5) with the inverse convolution method can be described as follows. First we have the preparatory steps:

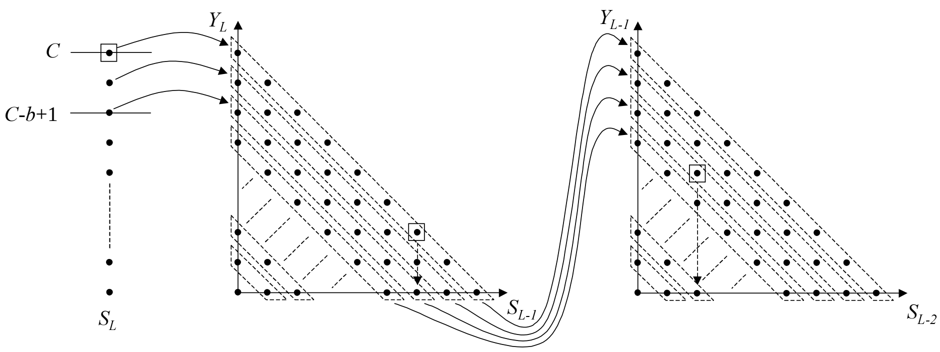

Figure 3: Sample generation into the set $\mathcal{D}$ with the inverse convolution method.

1. Compute the distribution of S_L recursively from (6).
2. Compute the conditional distributions given by (7), for $l = 1, \dots, L$.

To generate a sample we perform the following

1. Generate a value, say s, for link occupancy S_L from $\mathrm{P}\{S_L = x \,|\, C - b_L < S_L \le C\}$.
2. For i from L downto 2
   - Generate the value of Y_i from the distribution $\mathrm{P}\{Y_i \le y \,|\, S_i = s\}$ (eq. (7)).
   - Set $X_i \leftarrow Y_i / b_i$.
   - Set $s \leftarrow s - b_i X_i$.
3. Set $X_1 \leftarrow s/b_1$.
4. For i from $L+1$ to K, generate X_i from $f_i(x) = (\rho_i^x / x!)\, e^{-\rho_i}$.

5 Numerical results

5.1 Allocation of the sample points

Here we reintroduce the dependence on the link index j explicitly in the notation. Above we have decomposed the problem of estimating the expectation $\beta = \mathrm{E}[h(\mathbf{X})]$ into J independent problems of estimating the expectations $\eta^{(j)} = \mathrm{E}[h^j(\mathbf{X})],\ j = 1, \dots, J$, with $\beta = \sum_j \eta^{(j)}$, and correspondingly $\hat{\beta} = \sum_j \hat{\eta}^{(j)}$. Each of the estimators $\hat{\eta}^{(j)}$,

$$\hat{\eta}^{(j)} = \frac{1}{N_j} \sum_{n=1}^{N_j} h^{(j)}(\mathbf{X}_n^{(j)}),$$

where $\mathbf{X}^{(j)}$ is a random vector obeying the distribution $p_j^*(\cdot)$, gives an unbiased estimate for $\eta^{(j)}$, irrespective of the number of samples N_j used. The allocation of the total number of samples N between different subproblems, $N = N_1 + \cdots + N_J$, should be made based on the minimization of the variance of the final estimator $\hat{\beta}$. Because the estimators $\hat{\beta}_j$ are independent we have

$$\mathrm{V}[\hat{\beta}] = \sum_j \mathrm{V}[\hat{\eta}^{(j)}] = \sum_j \frac{s_j^2}{N_j},$$

where we have denoted $s_j^2 = \mathrm{V}[h^j(\mathbf{X}^{(j)})]$. Now the minimization of this expression with respect to the N_j under the constraint $\sum_j N_j = N$ readily leads to the optimal allocation

$$N_j = \frac{s_j}{\sum_{i=1}^J s_i} N, \quad j = 1, \ldots, J. \tag{8}$$

Of course, the s_j are not known before the simulation. Therefore a dynamic sample allocation scheme is needed. One practical solution is to make the simulation in batches, using $J * M$ samples per batch, where M is a suitable integer, for instance $M = 100$. In the first batch, all the samples are distributed evenly for different links, i.e., M samples are used per link. Then initial estimates for the s_j are obtained. Using these estimates, the optimal sample sizes after the second batch, i.e. for $N = 2J * M$, can be calculated from (8). If the calculated N_j is less than the number of samples already used (M samples in the first batch) no samples of the new batch are allocated for that link. Otherwise, the available $J * M$ new samples are distributed between the links in proportion to the deficiencies (deficiency being the difference between the calculated optimal value after the new batch and the actual number of samples used so far). Real numbers are appropriately rounded to integers. After the new batch, new estimators are calculated for the s_j and the procedure is repeated.

5.2 Numerical examples

Here some numerical examples are presented in order to illustrate the efficiency of the presented method in Monte Carlo simulation of the blocking probabilities. First we consider a simple two traffic class network with three links. The parameters of the network are: $C_j = [100, 120, 170], \mathbf{b}^{(1)} = (2, 0), \mathbf{b}^{(2)} = (0, 3)$ and $\mathbf{b}^{(3)} = (2, 3)$. We consider the blocking probability of traffic class 1 with two different loads such that the blocking probabilities are of the order $1.03 \cdot 10^{-2}$ and $1.22 \cdot 10^{-4}$ (Cases 1 and 2 in Table 1, respectively). The offered loads were $\boldsymbol{\rho} = (35, 22)$ (Case 1) and $\boldsymbol{\rho} = (27, 18)$ (Case 2). The inverse convolution method (labeled with "Convolution" in the table) is compared against results obtained with the composite method ("Composite") from [1], the standard MC and the methods proposed by Mandjes ("Single twist" in Table 1) in [2] and Ross in [3, chap. 6], which both correspond to the use of a single twisted IS distribution. To this end, we estimated the relative deviation of the estimator, given by $(\mathrm{V}[\widehat{\beta}_k])^{1/2}/\widehat{\beta}_k$, for 10^4 samples (Case 1) and 10^5 samples (Case 2). Our second example is the large network example from [6] for the scaling factor $N = 25$. The example network is a lightly loaded network with blocking probabilities of the order 10^{-3} for each traffic class. There are 10 traffic classes and 13 links with large capacities (several hundreds of capacity units). Again, we estimated the relative deviation of $\widehat{\beta}_k$ for traffic classes 6 (Case 3) and 8 (Case 4) with 10^5 samples.

Table 1: The relative deviation of the estimates $\widehat{\beta}_k$ for the examples.

Case	Convolution	Composite	Single twist	Ross	MC
1	0.0023	0.051	0.060	0.066	0.099
2	0.0003	0.017	0.027	0.076	0.302
3	0.0007	0.031	0.031	0.071	0.095
4	0.0014	0.017	0.020	0.029	0.037

As can be seen, the variance reductions obtained with the inverse convolution method are remarkable. For example, in Case 2, the ratio between the deviations of the standard MC and the inverse convolution method is about 1000 and even in the large network examples the ratio is about 135 in Case 3 and 26 in Case 4. These ratios of the deviations correspond to ratios of 1 000 000 (Case 2), 18 000 (Case 3) and 670 (Case 4) in terms of the required number of samples.

Also, we can note here that with the inverse convolution method the estimation of the variance of the estimates is guaranteed to be reliable. In rare event simulation (which is not the main interest here), a problem is that one can get results that appear to be very accurate judging by the estimated variance, but the results can, in reality, be far from the correct value. This can happen e.g. when using a single heavily twisted IS distribution and the reason is that the likelihood ratio $w(\cdot)$ can have a huge value at some point in the state space, but under the twisted distribution these points are very rare and are never encountered during the course of a simulation run. Hence, the estimates, especially for the variance or other higher moments, can be heavily under estimated, as has been rigorously shown in [5]. However, with the inverse convolution method the estimation is always reliable, since the observed values of the samples are bounded within the interval $[0, 1]$. Thus, the problem of the occurrence of events with a very small probability under the IS distribution but having a significant contribution to the estimate does not occur with the inverse convolution method.

6 Conclusions

In this paper we have presented a new approach to the problem of estimating blocking probabilities in a multiservice loss system by using the static Monte Carlo simulation method and importance sampling. First we observed that the estimation problem can be decomposed into separate simpler sub-problems each roughly corresponding to the estimation of the blocking probability contribution from a single link. For the solution of the sub-problems, we presented the inverse convolution method, which very closely approximates the generation of samples with the ideal IS distribution. The idea is to generate samples directly into the set of blocking states of a given link in the system, where all the other links are assumed to have an infinite capacity. This set of course extends beyond the allowed state space of the system. Then, simulation is essentially only needed to determine which part of this set is actually inside the allowed state space. In terms of the obtained variance reduction, the inverse convolution method by far surpasses all previously reported results. The excellent results of the inverse convolution method, however, are obtained at the cost of high, though manageable, memory requirements. However, it can be noted that the memory requirements of the inverse convolution algorithm can be significantly reduced by constructing the conditional distributions on the fly for each sample with the trade-off of making the sample generation process somewhat more time consuming.

Acknowledgement

The authors thank Jouni Karvo for useful discussions which led them to consider IS distributions outside the family of exponentially twisted Poisson distributions. This research has been funded by the Finnish Academy. The work of the first author has also been financially supported by the Nokia Foundation.

References

[1] P. E. Lassila and J. T. Virtamo, "Efficient Importance Sampling for Monte Carlo Simulation of Loss Systems", Proceedings of the ITC-16, Edinburgh, 7-11 June, 1999, Teletraffic Engineering in a Competitive World, Elsevier, 1999, pp. 787-796.

[2] M. Mandjes, “Fast simulation of blocking probabilities in loss networks”, European Journal of Operations Research, Vol. 101, 1997, pp. 393-405.

[3] K. W. Ross, “Multiservice Loss Models for Broadband Telecommunication Networks”, Springer-Verlag, London, 1995.

[4] J. S. Sadowsky, J. A. Bucklew, “On Large Deviations Theory and Asymptotically Efficient Monte Carlo Estimation”, IEEE Transactions on Information Theory, vol. 36, 1990, pp. 579–588.

[5] J. S. Sadowsky, “On the Optimality and Stability of Exponential Twisting in Monte Carlo Estimation”, IEEE Transactions on Information Theory, vol. 39, 1993, pp. 119–128.

[6] A. Simonian, J. W. Roberts, F. Theberge, R. Mazumdar, “Asymptotic Estimates for Blocking Probabilities in a Large Multi-rate Loss Network”, Advances in Applied Probability, vol. 29, 1997, pp. 806–829.