Date: January 17, 2000 COST257TD(00)
Issue: 1

Nearly Optimal Importance Sampling for Monte Carlo Simulation of Loss Systems

Pasi Lassila [1] and Jorma Virtamo [2]

Helsinki University of Technology

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. Then two importance sampling distributions are presented, which very closely approximate the ideal importance sampling distribution for each sub-problem. In both methods, the idea is to try to generate samples directly into the blocking state region. The difference between the methods is that the first method, the inverse convolution method, achieves this exactly, while the second one, using a fitted Gaussian distribution, only approximately. The inverse convolution algorithm, however, has a higher memory requirement. 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 methods, especially with the inverse convolution method, is truly remarkable, between 400 and 500 000 in the considered examples.

Keywords: Loss systems, simulation, Monte Carlo methods, variance reduction, importance sampling.

[1] Pasi.Lassila@hut.fi
[2] Jorma.Virtamo@hut.fi

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. [4]. 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 has 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 two efficient IS distributions are 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 [4, chap. 6] and Mandjes [3]. Ross has presented heuristics which attempt to increase the likelihood of the blocking states while, at the same, 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 [2], 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 [6]. 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 we take on a slightly different approach. The basic idea is the same as in [2], to effectively sample the blocking states lying on the boundary of each active link contraint. 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 we give two effective IS methods to solve each sub-problem. In these methods the earlier used exponentially twisted distributions are replaced with a more accurate approximation of the ideal IS distribution. In both methods 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. The first method, based on using an inverse convolution, achieves this objective exactly. The second one is an approximation of the first method, where a Gaussian approximation of the original distribution is used. The trade-off in the two methods is between the better performance of the first method and the lower memory consumption of the second method. The two methods drastically improve the performance of the IS sampling. In the examples considered, the reduction of the standard deviation obtained by the inverse convolution method varied from 20 to 700, 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 400 to 500 000.

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 and 5 contain the main results of the paper and describe the inverse convolution method and the Gaussian IS method, respectively. In section 6 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 7 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. Also, 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, $\mathcal{S}$,

$$\pi(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} | \mathbf{X} \in \mathcal{S}\} = \frac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{S}\}}, \quad \mathbf{x} \in \mathcal{S}.$$

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

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

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 some additional notation. Let $\mathcal{D}_k^j$ denote the set of blocking states for link j consisting of the points

$$\mathcal{D}_k^j = \left\{ \mathbf{x} \in \mathcal{I} \mid C_j - b_k^j + 1 \leq \mathbf{b}^j \cdot \mathbf{x} \leq C_j \right\}.$$

Also, we denote by $\mathcal{R}_k$ the set of links that the traffic class k uses, 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.

3 Efficient importance sampling for loss systems

In what follows we discuss the estimation of the blocking probabilities via the importance sampling simulation method. Then, as the form of the stationary distribution $\pi(\mathbf{x})$ 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 consists of generating samples from the distribution $f(\cdot)$, which is easy to do, and for each sample checking whether it is in the allowed state space or not. The estimate for γ corresponds then simply to the probability of generating hits into $\mathcal{S}$. This probability is usually close to 1 and is, therefore, easy to estimate. It is only in the rather unrealistic case where the traffic in the system is extremely heavy such that the main mass of the distribution $f(\mathbf{x})$ lies far outside $\mathcal{S}$, when the direct estimation of γ may become a problem and importance sampling may be needed for that, too. Therefore, in the rest of this paper we concentrate on efficient methods for estimating β_k.

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. Let $\mathbf{X}^* \in \mathcal{I}$ be another random variable with distribution

$$P^* : p^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X}^* = \mathbf{x}\} > 0 \quad \forall\ \mathbf{X}^* \in \mathcal{B}. \tag{4}$$

Note that the above requirement allows $p^*(\mathbf{x})$ to be positive also outside $\mathcal{B}$. Then β can be written as an expectation

$$\beta = \sum_{\mathbf{x}\in\mathcal{B}} \frac{f(\mathbf{x})}{p^*(\mathbf{x})} p^*(\mathbf{x}) = \mathrm{E}[1_{\mathbf{X}^*\in\mathcal{B}}\ w(\mathbf{X}^*)], \tag{5}$$

where $w(\cdot) = f(\cdot)/p^*(\cdot)$ is the so called likelihood ratio. Thus, we have the following estimator

$$\widehat{\beta} = \frac{1}{N} \sum_{n=1}^{N} 1_{\mathbf{X}^*_n\in\mathcal{B}}\ w(\mathbf{X}^*_n), \tag{6}$$

where N is the number of generated samples of $\mathbf{X}^*$.

The relation (5) and the resulting estimator show the basic principle of a simulation method known as importance sampling, where the idea is to choose the sampling distribution P^* satisfying (4) such that the variance of (6) is minimized. In [2] it was shown that the variance of the observed variable $1_{\mathbf{X}^*\in\mathcal{B}}\ w(\mathbf{X}^*)$ under the distribution P^* can be expressed as

$$\mathrm{V}\left[1_{\mathbf{X}^*\in\mathcal{B}}\ w(\mathbf{X}^*)\right] = \frac{\beta^2}{\beta^*} - \beta^2 + \beta^*\sigma^{*2}, \tag{7}$$

where β^* is the blocking probability under P^*, $\beta^* = \mathrm{E}[1_{\mathbf{X}^*\in\mathcal{B}}]$, and σ^{*2} is the variance of the observed variable in the set of the blocking states under P^*, $\sigma^{*2} = \mathrm{V}[w(\mathbf{X}^*) \mid \mathbf{X}^* \in \mathcal{B}]$. From this we can note that the ideal IS distribution has the following properties: each generated sample is in the set $\mathcal{B}$ and the likelihood ratio $w(\mathbf{x})$ has a constant value in the set $\mathcal{B}$. However, the ideal IS distribution implies knowledge of the estimated quantity itself and is, hence, impractical. An efficient realizable IS distribution, however, tries to approximate it as closely as possible.

Earlier approaches to obtaining an efficient IS distribution for estimating the blocking probabilities, see [3] or [4], suggested the use of an exponentially twisted IS distribution that moves the main probability mass closer to one of the link constraints or, as in [3] to be centered around the most probable blocking state. However, in a well engineered loss system, the blocking probability of class-k calls, is not dominated by a single bottleneck link. Instead, on the boundaries of all the link constraints, there are states that contribute significantly to the blocking probability, implying that an efficient IS distribution must be capable of producing samples lying on the boundaries of all the links that the traffic class uses.

In [2] we approximated this by using a composite distribution, consisting of a weighted combination of several exponentially twisted distributions, one for each link in $\mathcal{R}$. Each distribution was centered around the most probable blocking state on link j and could, hence, be used to sample predominantly the blocking states of link constraint j. These exponentially twisted distributions are also Poisson distributions and thus the generation of samples is very easy. Then we only needed one parameter to completely define each twisted distribution making the control over where the samples get generated and the variance of $w(\cdot)$ for each link j in the set $\mathcal{B}$ somewhat limited. Here our aim is again to be able to sample the blocking states on each link constraint, but we will not use a composite distribution for this. Instead, the problem will be decomposed into separate simpler sub-problems. Then, we can easily derive an IS distribution very closely approximating the properties of the ideal IS distribution for each particular sub-problem.

3.1 Decomposition and importance sampling

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 $\mathrm{E}[h(\mathbf{X})]$ with $h(\cdot)$ being the indicator function of the set $\mathcal{B}$. Based on the above we can decompose $h(\cdot)$ as

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

where

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

and $\nu(\mathbf{x})$ is a function giving the number of sets $\mathcal{D}^j$ 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. Thus, the computation of the original expectation decomposes into independent sub-problems, i.e $\mathrm{E}[h(\mathbf{X})] = \sum_{j\in\mathcal{R}} \mathrm{E}[h^j(\mathbf{X})]$. 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 and, whence there would not appear any $1/\nu(\mathbf{x})$ term in the $h^j(\mathbf{x})$ function.

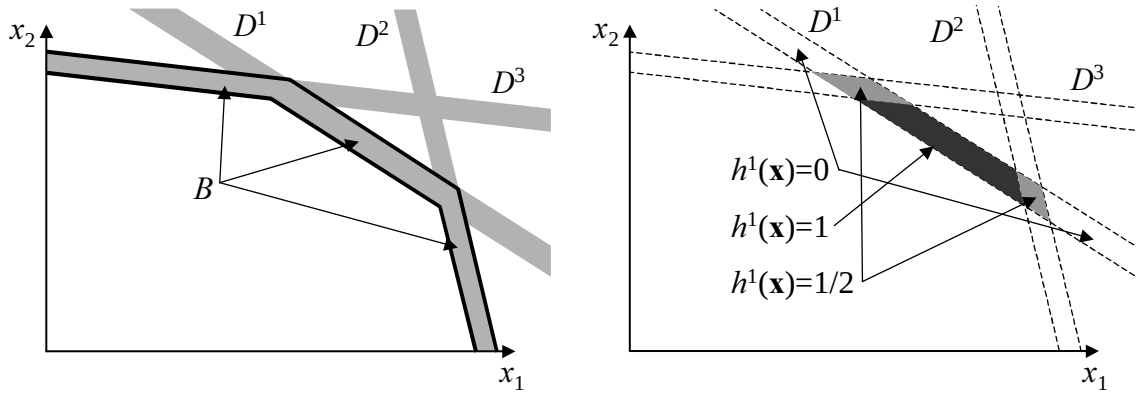

Figure 1: Decomposition of the set $\mathcal{B}$ into three subsets in a network with two traffic classes and three link constraints (left figure) and the values of one of the $h^j(\cdot)$ functions in different parts of $\mathcal{D}^j$ (right figure).

To estimate each $\eta^j = \mathrm{E}[h^j(\mathbf{x})]$ efficiently we apply importance sampling. Let $p_j^*(\cdot)$ denote the corresponding IS distribution. Then we have the IS estimator

$$\widehat{\eta}^j = \frac{1}{N_j} \sum_{n=1}^{N_j} \frac{1}{\nu(\mathbf{X}_n^*)} 1_{\mathbf{X}_n^* \in \mathcal{S}}\, 1_{\mathbf{X}_n^* \in \mathcal{D}^j}\, w(\mathbf{X}_n^*), \tag{8}$$

where $w(\mathbf{x}) = f(\mathbf{x})/p_j^*(\mathbf{x})$. 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 sub problem are free parameters. In section 6 we show how to choose the N_j to minimize the variance of $\widehat{\beta}$.

We will next present two new methods that try to approximate the ideal IS distribution for estimating η^j as closely as possible without making the computational effort of generating samples excessive. For estimator (8) 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. In both new methods we approximate this by deriving a

distribution for which the value of $w(\cdot)$ is (almost) constant and from which we can generate samples (almost) directly into $\mathcal{D}^j$, i.e. the region of states corresponding to blocking states on link j. Then the only sources of variance in the estimator (8) are due to the fact that some of the samples in $\mathcal{D}^j$ are not in the allowed state space $\mathcal{S}$ and due to the inverse of the multiplicity factor $1/\nu(\cdot)$. With the exponentially twisted distributions used earlier such control was not possible. Hence, we get a much better approximation of the ideal distribution with the new methods.

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 $\mathbf{X} \in \mathcal{D}$ exactly obeying the distribution P, i.e. from conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$ 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.

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

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

$$q_l(x) = \sum_{y=0}^{x} q_{l-1}(x-y) m_l(y). \tag{9}$$

Note that the event $S_l = x$ is the union of the events $\{Y_l = y, S_{l-1} = x - y\}$, $y = 0, \ldots, x$ 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 $y = 0, 1, \ldots, x$. 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

$$\mathrm{P}\{Y_l \le y \,|\, S_l = x\} = \sum_{y'=0}^{y} m_l(y') q_{l-1}(x-y')/q_l(x). \tag{10}$$

Then the value of $Y_l \le y$ can be drawn by finding the smallest y such that $\mathrm{P}\{Y_l \le y \,|\, S_l = x\} \ge U$, 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 $C-b+1 \leq S_L \leq C$. A point in $\mathcal{D}$ can be generated by first drawing a value for S_L using the distribution $q_L(\cdot)$ conditioned on $C-b+1 \leq S_L \leq C$, which is also precomputed and stored. This is shown in Figure 2 on the left hand side. Then, as described above, (Y_L, S_{L-1}) can be drawn. This is shown in Figure 2 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 $1, \ldots, L$ and from each other. Hence, their values are drawn independently from the distributions $f_k(\cdot)$, $k =, L+1, \ldots, K$.

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 (10), must be constructed on the fly, making the sample generation somewhat slower.

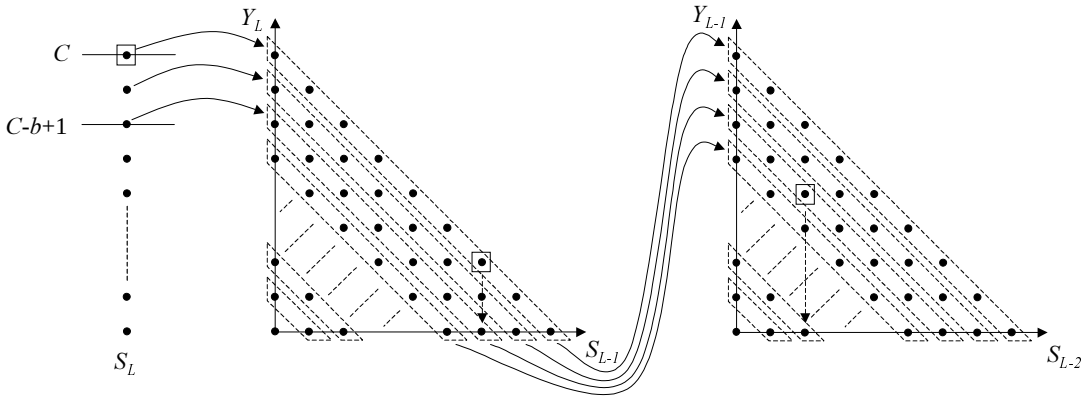

Figure 2: Sample generation into the set $\mathcal{D}$ with the inverse convolution method.

The samples $\mathbf{X}_n^*$ generated with the above method and to be used in the IS estimator (8) obey the conditional distribution

$$p^*(\mathbf{x}) = \mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\} = \frac{\mathrm{P}\{\mathbf{X} = \mathbf{x}\}}{\mathrm{P}\{\mathbf{X} \in \mathcal{D}\}} = \frac{f(\mathbf{x})}{v},$$

where v is the probability mass of the set $\mathcal{D}$, i.e.

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

Then the estimator for η becomes

$$\begin{aligned}\widehat{\eta} &= \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\nu(\mathbf{X}_n^*)}\,1_{\mathbf{X}_n^*\in\mathcal{S}}\,1_{\mathbf{X}_n^*\in\mathcal{D}}\,\frac{f(\mathbf{X}_n^*)}{f(\mathbf{X}_n^*)/v}\\ &= \frac{v}{N}\sum_{n=1}^{N}\frac{1}{\nu(\mathbf{X}_n^*)}\,1_{\mathbf{X}_n^*\in\mathcal{S}},\end{aligned}$$

where we have omitted the indicator $1_{\mathbf{X}_n^*\in\mathcal{D}}$ since with the inverse convolution method the generated samples are always inside $\mathcal{D}$. In practice the samples are not generated in the infinite space $\mathcal{I}$ but in some smaller Cartesian product space enclosing $\mathcal{S}$, which further increases the hit ratio of the method. Also, note that with this method simulation is needed only to determine which part of the probability mass of $\mathcal{D}$ is actually inside $\mathcal{S}$ (factor $1_{\mathbf{X}^*\in\mathcal{S}}$) and to compensate for double (or multiple) counting for such points $\mathbf{x}$ that belong to more than one of the sets $\mathcal{D}$ (factor $1/\nu(\mathbf{X}^*)$).

5 Gaussian IS distribution for loss systems

In this section we present another IS distribution for estimating η_k^j. However, now the $p_j^*(\mathbf{x})$ will only approximately represent the conditional distribution $\mathrm{P}\{\mathbf{X}=\mathbf{x}\mid\mathbf{X}\in\mathcal{D}_j^k\}$. On the other hand, the generation of samples from this distribution can be done without the precomputation and storage of a large number of probability tables as required in the inverse convolution method. Again, for ease of notation, we omit the dependence on the traffic class k for which we are estimating η_k^j and the link $j\in\mathcal{R}_k$ under consideration.

The idea of the method is briefly as follows. First we find the point $\mathbf{x}^*$ maximizing $f(\mathbf{x})$ on the constraining hyperplane $\mathbf{b}\cdot\mathbf{x}=C$. Then, at that most important point $\mathbf{x}^*$ we fit a Gaussian function $g(\mathbf{x})$ to $f(\mathbf{x})$ (considered as a continuous function of $\mathbf{x}$). This Gaussian function is used as an approximation to $f(\mathbf{x})$. The distribution $\mathrm{P}\{\mathbf{X}=\mathbf{x}\mid\mathbf{X}\in\mathcal{D}\}$ can now be approximated by a conditional multinormal distribution. Sample points in $\mathcal{D}$ can be generated by first generating a value for the link occupancy from its marginal distribution in the strip $C-b+1\le\mathbf{b}\cdot\mathbf{X}\le C$ and then generating the other coordinates from a conditional multinormal distribution. As the normal distribution is a continuous distribution, we finally have to discretize the values by rounding them to the closest integers.

There is, however, a small technical problem in the method. As will be explained, in order to be able to make the calculation of the likelihood ratio practical, we have to enlarge the strip $C-b+1\le\mathbf{b}\cdot\mathbf{X}\le C$ somewhat, i.e. we use limits $r\le\mathbf{b}\cdot\mathbf{X}\le s$, where $r<C-b+1$ and $s>C$. Unfortunately, this small problem turns out to have a rather big impact on the performance of the method in terms of the miss ratio.

So, we start by considering the fitting of a Gaussian function $g(\mathbf{x})$ to $f(\mathbf{x})$ at $\mathbf{x}^*$. The fitting procedure is described in more detail in the Appendix. Since $f(\mathbf{x})$ is of the product form

$\prod_k f_k(\mathbf{x})$, the fitting reduces to k one-dimensional problems of fitting a Gaussian function

$$g_k(x) = \frac{c_k}{\sqrt{2\pi}} \frac{1}{\sigma_k} e^{-(x-m_k)^2/2\sigma_k^2}$$

to a given function $f_k(x)$ at a given point x_k^*. As there are three parameters c_k, m_k and σ_k available, we can require the 0^{th}, 1^{st} and 2^{nd} derivatives of $f_k(\cdot)$ to match those of $g_k(\cdot)$ at x_k^*. The fitting results in a Gaussian function

$$g(\mathbf{x}) = \frac{a}{(\sqrt{2\pi})^K |\mathbf{\Gamma}|} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T \mathbf{\Gamma}^{-1}(\mathbf{x}-\mathbf{m})},$$

where $a = \prod_k c_k$ and the covariance matrix $\mathbf{\Gamma}$ is a diagonal matrix with $\Gamma_{kk} = \sigma_k^2$. Note that $g(\mathbf{x})$ is a times the density function of the multinormal distribution $N(\mathbf{m}, \mathbf{\Gamma})$. Indeed, the fitted function need not be the density of a distribution. Now the conditional distribution $\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$ can be approximated by the (discretized version of) $N(\mathbf{m}, \mathbf{\Gamma})$ distribution conditioned on $r \le \mathbf{x} \cdot \mathbf{b} \le s$ (recall that $\mathbf{b}$ denotes implicitly the vector $\mathbf{b}^j$ and b_k, to be used later, is the k^{th} component of $\mathbf{b}^j$).

To further simplify the notation in the following we assume, without loss of generality, that the traffic class $k \in \{1, \ldots, L\}$ for which we are estimating η is class K. We now make a linear transformation of variables by replacing X_K with the occupancy of the link $\sum_k b_k x_k$. This transformation and its inverse transformation are

$$\begin{cases} z_1 &= x_1, \\ &\vdots \\ z_{K-1} &= x_{K-1}, \\ z_K &= \sum_{k=1}^{K} b_k x_k, \end{cases} \quad \text{and} \quad \begin{cases} x_1 &= z_1, \\ &\vdots \\ x_{K-1} &= z_{K-1}, \\ x_K &= \frac{1}{b_K}\left(x_K - \sum_{k=1}^{K-1} b_k x_k\right). \end{cases}$$

The above equations can be expressed in matrix notation as

$$\begin{cases} \mathbf{z} &= \mathbf{A}^{-1}\mathbf{x}, \\ \mathbf{x} &= \mathbf{A}\mathbf{z}. \end{cases}$$

Now it is easy to verify that if $\mathbf{X}$ is a random variable with distribution $N(\mathbf{m}, \mathbf{\Gamma})$ then $\mathbf{Z} = \mathbf{A}^{-1}\mathbf{X}$ is a random variable with distribution $N(\widetilde{\mathbf{m}}, \widetilde{\mathbf{\Gamma}})$, where

$$\begin{cases} \widetilde{\mathbf{m}} &= \mathbf{A}^{-1}\mathbf{m}, \\ \widetilde{\mathbf{\Gamma}}^{-1} &= \mathbf{A}^T\mathbf{\Gamma}^{-1}\mathbf{A} \quad (\Rightarrow \widetilde{\mathbf{\Gamma}} = \mathbf{A}^{-1}\mathbf{\Gamma}(\mathbf{A}^{-1})^T). \end{cases} \tag{11}$$

In general, $\widetilde{\mathbf{\Gamma}}$ is no longer diagonal, i.e. the components of $\mathbf{Z}$ are not independent.

The conditional distribution of $\mathbf{X} \sim N(\mathbf{m}, \mathbf{\Gamma})$ conditioned on $r \le \mathbf{X} \cdot \mathbf{b} \le s$ corresponds to the conditional distribution of $\mathbf{Z} \sim N(\widetilde{\mathbf{m}}, \widetilde{\mathbf{\Gamma}})$ conditioned on $r \le Z_K \le s$. It is easy to generate $\mathbf{Z}$ from this distribution, and then we get $\mathbf{X}$ from $\mathbf{X} = \mathbf{A}\mathbf{Z}$.

To generate $\mathbf{Z}$, we observe that Z_K obeys a univariate normal distribution $Z_K \sim N(\widetilde{m}_K, \widetilde{\Gamma}_{KK})$, and its value in the range $r \leq Z_K \leq s$ can be generated by any of the standard methods (e.g. by inversion of the cumulative distribution function, or, more efficiently, by the acceptance rejection method using an exponential majorizing function, see [5, chap. 2], also Appendix 2). Second, given the value of Z_K, the other components of $\mathbf{Z}$, i.e. $\mathbf{Z}^{(1)} = [Z_1, \ldots, Z_{K-1}]$ again obey a multinormal distribution by Theorem 10.2 in [1, p. 324]

$$\mathbf{Z}^{(1)} \sim N(\widetilde{\mathbf{m}}^{(1)} - \mathbf{B}_{11}^{-1}\mathbf{B}_{12}(Z_K - \widetilde{m}_K), \mathbf{B}_{11}^{-1}),$$

where $\widetilde{\mathbf{m}}^{(1)}$ denotes the first $K-1$ components of $\widetilde{\mathbf{m}}$ and the $\mathbf{B}_{ij}$, $i,j = 1,2$, represent components in the partitioning of $\widetilde{\mathbf{\Gamma}}$

$$\widetilde{\mathbf{\Gamma}} = \begin{pmatrix} \overbrace{\mathbf{B}_{11}}^{K-1} & \overbrace{\mathbf{B}_{12}}^{1} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} \begin{matrix} \} \, K-1 \\ \} \, 1 \end{matrix} .$$

Note that $\mathbf{B}_{22} = \widetilde{\Gamma}_{KK}$. Thus

$$\mathbf{Z}^{(1)} \sim \widetilde{\mathbf{m}}^{(1)} - \mathbf{B}_{11}^{-1}\mathbf{B}_{12}(Z_K - \widetilde{m}_K) + \mathbf{B}_{11}^{-1/2} \cdot \mathbf{N}^{(1)}, \tag{12}$$

where $\mathbf{N}^{(1)}$ is a vector of $K-1$ independent $N(0,1)$ distributed random variables. For a fixed Z_K, the expression on the right hand side is obviously a multinormal variable with mean $\mathrm{E}[\mathbf{Z}^{(1)}] = \widetilde{\mathbf{m}}^{(1)} - \mathbf{B}_{11}^{-1}\mathbf{B}_{12}(Z_K - \widetilde{m}_K)$ and covariance

$$\mathrm{E}\left[\left(\mathbf{Z}^{(1)} - \mathrm{E}\left[\mathbf{Z}^{(1)}\right]\right)\left(\mathbf{Z}^{(1)} - \mathrm{E}\left[\mathbf{Z}^{(1)}\right]\right)^{\mathrm{T}}\right] = \mathbf{B}_{11}^{-1/2}\,\mathrm{E}\left[\mathbf{N}^{(1)}(\mathbf{N}^{(1)})^{\mathrm{T}}\right]\,\mathbf{B}_{11}^{-1/2} = \mathbf{B}_{11}^{-1}.$$

In summary, the procedure for generating samples into $\mathcal{D}$ can be described as follows. First we have the preparatory steps

1. Obtain $\mathbf{m}$ and $\mathbf{\Gamma}$ from the fitting procedure.
2. Calculate $\widetilde{\mathbf{m}}$ and $\widetilde{\mathbf{\Gamma}}$ from (11).
3. Determine the submatrices $\mathbf{B}_{ij}$ and calculate $\mathbf{B}_{11}^{-1}\mathbf{B}_{12}$ and $\mathbf{B}_{11}^{-1/2}$, needed in (12).

Then for each sample we perform the following

1. Generate Z_K from $N(\widetilde{m}_K, \widetilde{\Gamma}_{KK})$ in the interval (r,s) (see Appendix 2).
2. Generate $\mathbf{Z}^{(1)}$ using (12).
3. $\mathbf{X} = \mathbf{A}\mathbf{Z}$, where $\mathbf{Z} = [\mathbf{Z}^{(1)}, Z_K]$.
4. Round the components of $\mathbf{X}$ to closest integers.

5.1 Likelihood ratio

Let us denote here with $\mathbf{X}_n^*$ the samples obtained in the above described manner. In order to use the samples in the estimator (8), we have to be able to calculate the likelihood ratio and therefore the probability of the generated samples, i.e. the integer lattice points in $\mathcal{D}$. Note that, because of the rounding operation, the probability of a sample $\mathbf{X}_n^*$ equals the probability mass of the conditional normal distribution within the K-dimensional unit cube with the center $\mathbf{X}_n^*$. If the cube is totally embedded in the strip defined by the condition $r \leq Z_K \leq s$, then the calculation is easy. Even though we made a change of variables in order to control the values of Z_K, the density in the strip $r \leq Z_K \leq s$ still is the product of Gaussian densities. Thus, the probability mass in the cube can be expressed as the product of probabilities of respective intervals of normal variables. On the other hand, if the cube is not wholly embedded in the strip $r \leq Z_K \leq s$, then the calculation of the probability is complicated.

Thus we have to ensure that for each integer lattice point in $\mathcal{D}$ the surrounding unit cube is wholly embedded in the strip $r \leq Z_K \leq s$. This means that we must enlarge the strip by choosing $r = C_j - b + 1 - \Delta, s = C + \Delta$, where $\Delta = \frac{1}{2}\sum_k b_k$. This is illustrated in Figure 3, where the grey area corresponds to $\mathcal{D}$. Note that $\Delta = [\frac{1}{2}, \ldots, \frac{1}{2}] \cdot \mathbf{b}$, where $[\frac{1}{2}, \ldots, \frac{1}{2}]$ is the vector distance of the corner of the cube from the center. By enlarging the strip we inevitably generate misses from the set $\mathcal{D}$ and to some extent deteriorate the performance of the sampling method.

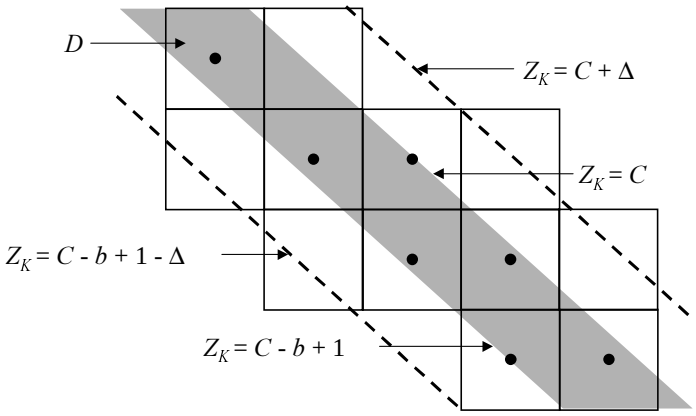

Figure 3: Enlargement of the generation interval of Z_K.

Remember that we fitted the Gaussian function $g(\cdot)$ such that it approximates the distribution $f(\cdot)$ as closely as possible at the point $\mathbf{x}^*$, i.e.

$$f(\mathbf{x}) \approx g(\mathbf{x}) = a\, f_{N(\mathbf{m},\mathbf{\Gamma})}(\mathbf{x}),$$

where $f_{N(\mathbf{m},\mathbf{\Gamma})}$ denotes the pdf of a normal distribution with mean $\mathbf{m}$ and covariance $\mathbf{\Gamma}$. Then, to be explicit, our IS distribution $p^*(\mathbf{x})$ approximating the conditional probability

$\mathrm{P}\{\mathbf{X} = \mathbf{x} \,|\, \mathbf{X} \in \mathcal{D}\}$ is given by

$$p^*(\mathbf{x}) = \frac{\prod_{k=1}^{K} \left(Q(x'_k - \frac{1}{2}) - Q(x'_k + \frac{1}{2}) \right)}{v}, \quad \mathbf{x} \in \mathcal{D},$$

where v is the total probability of the extended strip $v = Q(r'_k) - Q(s')$, the primes refer to the normalized variables

$$x'_k = \frac{x_k - m_k}{\sigma_k}, \quad r'_k = \frac{r_k - \widetilde{m}_k}{\sqrt{\widetilde{\Gamma}_{KK}}}, \quad s' = \frac{s - \widetilde{m}_k}{\sqrt{\widetilde{\Gamma}_{KK}}},$$

and $Q(\cdot)$ denotes the tail probability function of the standard $N(0,1)$ distribution

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \, e^{-y^2/2} \, dy = \frac{1}{2} \mathrm{erfc}(x/\sqrt{2}).$$

Finally, the estimator for η becomes

$$\begin{aligned} \widehat{\eta} &= \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\nu(\mathbf{X}_n^*)} \, 1_{\mathbf{X}_n^* \in \mathcal{S}} \, 1_{\mathbf{X}_n^* \in \mathcal{D}} \, \frac{f(\mathbf{X}_n^*)}{p^*(\mathbf{X}_n^*)} \\ &= \frac{v}{N} \sum_{n=1}^{N} \frac{1}{\nu(\mathbf{X}_n^*)} \, 1_{\mathbf{X}_n^* \in \mathcal{S}} \, 1_{\mathbf{X}_n^* \in \mathcal{D}} \, \frac{f(\mathbf{X}_n^*)}{\prod_{k=1}^{K} \left(Q(X'_k - \frac{1}{2}) - Q(X'_k + \frac{1}{2}) \right)}, \end{aligned}$$

where the $\mathbf{X}_n^*$ denote samples obtained with the above described procedure and generated into the enlarged strip.

6 Numerical results

6.1 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 $\eta^{(j)} = \mathrm{E}[h^j(\mathbf{X})]$, $j = 1, \ldots, J$, 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

$$N_j = \frac{s_j}{\sum_{i=1}^{J} s_i} N, \quad j = 1, \ldots, J. \tag{13}$$

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 (13). 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.

6.2 Numerical examples

Here some numerical examples are presented in order to illustrate the efficiency of the presented methods 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 two new methods (labeled with "Convolution" and "Gaussian" in the table) are compared against results obtained with the composite method ("Composite") from [2], the standard MC and the methods proposed by Mandjes ("Single twist" in Table 1) in [3] and Ross in [4, 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 [7] 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	Gaussian	Composite	Single twist	Ross	MC
1	0.0036	0.019	0.051	0.060	0.066	0.099
2	0.0004	0.006	0.017	0.027	0.076	0.302
3	0.0007	0.010	0.031	0.031	0.071	0.095
4	0.0022	0.008	0.017	0.020	0.029	0.037

As can be seen, the variance reductions obtained with the Gaussian method and 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 700 and even in the large network examples the ratio is about 100 in Case 3 and 20 in Case 4. As expected, the performance of the Gaussian method is worse than that of the inverse convolution method. However, the results even for this method are much better than for any of the IS methods using exponentially twisted distributions.

7 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 two methods, which very closely approximate the generation of samples with the ideal IS distribution. In both methods 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. The first method, the inverse convolution method, achieves this objective exactly, and the second one, the Gaussian method, approximately. 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. The Gaussian method does not require high memory usage, but the performance, while remarkably good, is less optimal. Finally, 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.

Appendix 1

Here we describe how to obtain the parameters of the Gaussian function $g(\cdot)$ to be used as an approximation to the Poisson distribution $f(\cdot)$ when estimating η_k^j. Again we assume from this point on that the dependence on j and k is implicit. The first problem is to identify the point around which we will approximate $f(\cdot)$. A natural choice for this point is the most probable blocking state in $\mathcal{D}$, denoted by $\mathbf{x}^*$. This problem involves the maximization of $f(\cdot)$ on a given hyperplane representing the link constraint under consideration. In [2] we showed how the solution to this problem can be obtained numerically in a straight forward way.

Once the most probable blocking state $\mathbf{x}^*$ is known, the Gaussian function $g(\mathbf{x})$ is fitted to the distribution $f(\mathbf{x})$ to at the point $\mathbf{x}^*$. In particular, we require the 0^{th}, 1^{st} and 2^{nd} derivatives of $f(\cdot)$ to match those of $g(\cdot)$. Since $f(\cdot)$ has a product form the fitting problem reduces to a simple componentwise fitting of

$$\begin{cases} g_k(x_k^*) &= f_k(x_k^*), \\ \dfrac{\partial}{\partial x} g_k(x_k^*) &= \dfrac{\partial}{\partial x} f_k(x_k^*), \qquad ,\forall\ k = 1, \ldots, K, \\ \dfrac{\partial^2}{\partial x^2} g_k(x_k^*) &= \dfrac{\partial^2}{\partial x^2} f_k(x_k^*), \end{cases} \tag{14}$$

where

$$g_k(x) = \frac{c_k}{\sqrt{2\pi}\sigma_k}\, e^{-(x-m_k)^2/2\sigma_k^2}.$$

Equations (14) can be solved analytically to get the parameters c_k, m_k and σ_k. To this end, let $a_1 = f_k(x_k^*)$, $a_2 = f_k'(x_k^*)$ and $a_3 = f_k''(x_k^*)$. After some straightforward manipulation one can obtain

$$\begin{cases} \sigma_k^2 &= \dfrac{a_1^2}{a_2^2 - a_1 a_3}, \\ m_k &= x_k^* + \dfrac{a_1 a_2}{a_2^2 - a_1 a_3}, \\ c_k &= a_1 \sqrt{2\pi}\sigma_k\, e^{a_2^2\sigma_k^2/2a_1^2}. \end{cases} \tag{15}$$

To illustrate the fitting, let us consider an example, where $g(x) = c/(\sqrt{2\pi}\sigma)\, e^{-(x-m)/2\sigma^2}$ is fitted at the point $x^* = 10$ to the Poisson distribution $f(x) = (\rho^x/x!)\, e^{-\rho}$ with $\rho = 5$. The fitting gives $c = 2.66, m = 2.20$ and $\sigma^2 = 10.51$. In Figure 4 we have plotted the probability density function (pdf) of the original discrete Poisson distribution and the continuous Gaussian function. As can be seen from the figure, the fitting is, indeed, very good around the point $x^* = 10$. Also, note that the fitted Gaussian function is equal to c times the pdf of the $N(m, \sigma^2)$ distribution.

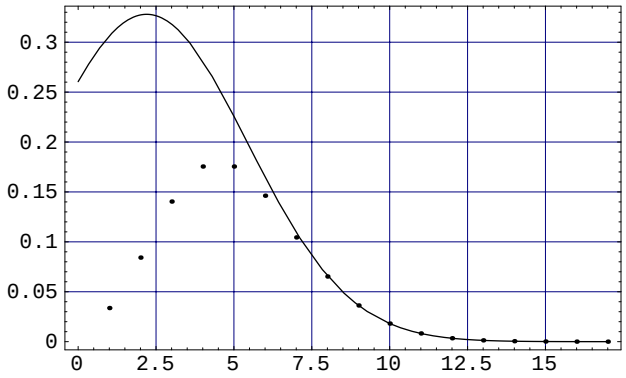

Figure 4: Point probabilities of a Poisson(5) distribution and the Gaussian function fitted at the point $x^* = 10$.

Appendix 2

Here we briefly describe how to efficiently generate samples from the $N(0,1)$ distribution in an interval (a,b), where $0 \leq a \leq b$, i.e samples X obeying the distribution

$$f(x) = \frac{1}{\sqrt{2\pi}}\, e^{-x^2/2},$$

conditioned on $X \in (a,b)$. We use an exponential majorizing function $g(x)$, which touches $f(x)$ at $x = a$,

$$g(x) = f(a) \cdot e^{-a(x-a)}.$$

With this function we have $f(x)/g(x) = e^{-(x-a)^2/2}$. Now, desired samples can be generated by first drawing a sample X from $\mathrm{Exp}(a)$ distribution conditioned on $X \in (a,b)$ and then accepting the result with probability $e^{-(X-a)^2/2}$. In summary, the algorithm is the following:

1. Compute $X = -\log\left(1 - \alpha U\right)/a$, where $\alpha = 1 - \left(1 - e^{-a(b-a)}\right)$,
2. If $e^{-X^2/2} \geq U$, then return $X + a$, else go to 1,

where each instance of U denotes an independent uniformly distributed random variable in the interval $(0,1)$. In the tail region, for a small interval (a,b), the method is very efficient in terms of the acceptance ratio. For given a the worst case is $b = \infty$, i.e. when the interval is not small. Then the acceptance ratio is $0.66, 0.84$ or 0.91 for $a = 1, 2, 3$, respectively. However, the acceptance ratios are much closer to 1, when we are considering small intervals, as is the case in our application.

Acknowledgement

The authors thank Jouni Karvo for useful discussions which led them to consider IS distributions outside the family of exponentially twisted Poisson distributions.

References

[1] S. M. Kay, "Fundamentals of Statistical Signal Processing: Estimation Theory", Prentice-Hall, 1993.

[2] 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.

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

[4] K. W. Ross, "Multiservice Loss Models for Broadband Telecommunication Networks", Springer-Verlag, London, 1995.

[5] R. Y. Rubinstein, B. Melamed, "Modern Simulation and Modeling", John Wiley & Sons, 1998.

[6] J. S. Sadowsky, J. A. Bucklew, "On Large Deviations Theory and Asymptotically Efficient Monte Carlo Estimation", IEEE Transactions on Information Theory, vol. 36, no. 3, 1990, pp. 579–588.

[7] 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.