Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

SOLVING PRODUCT FORM STOCHASTIC NETWORKS WITH MONTE CARLO SUMMATION

Keith W. Ross
Jie Wang

Department of Systems
University of Pennsylvania
Philadelphia, Pennsylvania 19104

ABSTRACT

Multiclass queueing networks and stochastic loss networks often give rise to a product form solution for their equilibrium probabilities. But the product form solution typically involves a normalization constant calling for a multidimensional summation over an astronomical number of states. We propose the application of Monte Carlo summation to the problem of determining the normalization constant and related performance measures. We show that if the proper sampling technique is employed then the computational effort of Monte Carlo summation is independent of population sizes for queueing networks and is independent of link capacities for loss networks. We then discuss the application of importance sampling and antithetic variates. Importance sampling is shown to give significant variance reduction for a multirate loss network example.

1. INTRODUCTION

It is well known that many multidimensional stochastic processes give rise to a product form solution for the equilibrium state probabilities. One of the more important examples is multichain queueing networks [Baskett et al. 1975; Kelly 1979] where the stochastic process models a queueing network with multiple classes of customers circulating according to class-dependent routing matrices. Another example is product form loss networks [Kelly 1986], where the stochastic process models a telephone network supporting calls with different bandwidth requirements (e.g., voice, video, facsimile). The appeal of the product form solution is that it enables one to circumvent solving the global balance equations. But if the state space is finite, as for closed multichain queueing networks or for multirate loss networks, then a summation must be performed over an excessive number of states in order to calculate the normalization constant. Due to the occurrence of product form networks in numerous important applications, over the past two decades many researchers have considered developing efficient methods to calculate the normalization constant and related performance measures. Currently, there are two principal schools of thought, the first based on combinatorial algorithms, and the second based on asymptotic expansions.

Buzen [1973] initiated the research on combinatorial algorithms with an efficient convolution algorithm to determine the normalization constant in single-class closed queueing networks. Reiser and Kobayashi [1975] developed a generalization of the convolution algorithm for closed multichain queueing networks, and numerous variations and refinements have since been proposed (e.g., see [Reiser and Lavenberg 1980; Lam and Lien 1983; Sauer 1983]). Although these algorithms can offer considerable computational savings over brute-force summation, their computational requirements continue to grow exponentially with either the number of classes or the number of stations, rendering them of little use for many problems of practical interest. Several efficient combinatorial algorithms for multirate loss networks have also been developed [Tsang 1988; Ross and Tsang 1988]. However, these combinatorial algorithms require a loss network with specific network topologies.

McKenna et al. [1981, 1982, 1986]; have pioneered the use of asymptotic expansions for solvingproduct form queueing networks. Their idea is to expand the normalization constant in $1/N$, where N is a "large parameter" reflecting the size of network. This technique has given impressive results for large multichain queueing networks, rapidly solving problems that are completely off limits to the combinatorial algorithms. However, except for a very particular topology [Mitra 1987], no progress has been made in developing asymptotic expansions for loss networks.

In this paper, we pursue a third school of thought, namely, applying Monte Carlo summation to the problem of evaluating the the normalization constant. Unlike the combinatorial methods, the Monte Carlo summation method has computational requirements that grow polynomially in the problem size. Furthermore, the Monte Carlo summation method is quite flexible as it can be adapted to arbitrary product form networks.

Harvey and Hills [1979] have considered a related Monte Carlo method - rejection sampling combined with conditional Monte Carlo - for solving a class of loss networks. Our study differs from theirs in its concern with the application of sampling and variance reduction techniques that have a bearing on large stochastic networks. We believe that conditional Monte Carlo is unlikely to give a significant reduction in variance for a loss network with a large number of links.

In Section 2 we discuss the application on Monte Carlo summation and importance sampling to product form "integrands". Special attention is given to ratio estimation since many performance measures in stochastic networks can be expressed as the ratio of two normalization constants. In Section 3 we apply the Monte Carlo summation technique to loss networks. Significant reduction in variance with the proper choice of importance sampling function is shown possible for a test network. In Section 4, multichain queueing networks are discussed along with two importance sampling techniques.

2. OVERVIEW OF MONTE CARLO SUMMATION

Let Ω denote the state space of the underlying stochastic process. Each element in Ω is a K-dimensional vector $\mathbf{n} = (n_1, \ldots, n_K)$. For the sake of simplicity we always assume that Ω is finite. The normalization constant, g, for product form stochastic networks most commonly takes the form

$$g := \sum_{\mathbf{n}\in\Omega} \prod_{k=1}^{K} q_k(n_k), \tag{1}$$

where $q_k(\cdot)$, $k = 1, \ldots, K$, are known functions (see Sections 3 and 4). Let

$$q(\mathbf{n}) := 1(\mathbf{n} \in \Omega) \prod_{k=1}^{K} q_k(n_k)$$

where $1(\cdot)$ is the indicator function. We can rewrite (1) as follows

$$g = \sum_{n_1=0}^{N_1} \cdots \sum_{n_K=0}^{N_K} q(\mathbf{n})$$

where $N_k := \max\{n_k : \mathbf{n} \in \Omega\}$. Thus calculating g involves a multidimensional summation. But it is now the believe of many researchers that, in the absence of special structure, multidimensional integration (or summation) is best solved by Monte Carlo methods [Kalos and Whitlock 1986]. Specifically, let $\mathbf{V}^i = (V_1^i, \ldots, V_K^i)$, $i = 1, 2, \ldots$, be a sequence of i.i.d. random vectors, where each $\mathbf{V}^i$ takes values in $\Lambda := \{0, \ldots, N_1\} \times \cdots \times \{0, \ldots, N_K\}$. Let $p(\mathbf{n}) := P(\mathbf{V}^i = \mathbf{n})$, $\mathbf{n} \in \Lambda$, which is a sampling distribution to be specified in order to obtain the maximum efficiency from the Monte Carlo method. Then

$$G_n := \frac{1}{n} \sum_{i=1}^{n} \frac{q(\mathbf{V}^i)}{p(\mathbf{V}^i)} \tag{2}$$

is an unbiased estimator for g (i.e., $E[G_n] = g$). Moreover, we have from the Central Limit Theorem, that for large n

$$P(|G_n - g| > \frac{c(\alpha)\sigma}{\sqrt{n}}) = 1 - \frac{\alpha}{2},$$

where $c(\alpha)$ is the critical value of the standard normal distribution and σ is the standard deviation of $q(\mathbf{V}^i)/p(\mathbf{V}^i)$. In particular, with 95% confidence we have

$$|G_n - g| \le \frac{2\sigma}{\sqrt{n}}. \tag{3}$$

Note that for any fixed n, G_n is an estimate of g whose accuracy can be accessed by the confidence interval provided by (3). As the samples are being drawn, the sample variance can be calculated and the confidence intervals can be given explicitly. Furthermore, if better accuracy is desired, more samples can be drawn, thereby decreasing the width of the confidence interval. From (2) and (3) we observe that the effectiveness of the Monte Carlo summation method largely depends on

1. the effort required to generate $\mathbf{V}^i$ from the distribution $p(\mathbf{n})$, $\mathbf{n} \in \Lambda$;
2. the effort required to evalute the "integrand" $q(\cdot)/p(\cdot)$ during the sampling procedure;
3. σ^2, the variance of $q(\mathbf{V}^i)/p(\mathbf{V}^i)$.

If $V_1^i, \ldots, V_K^i$ are independent (i.e., $p(\mathbf{n}) = p_1(n_1) \cdots p_K(n_K)$), then $\mathbf{V}^i$ can be generated in a total of $O(K)$ time with the algorithm of Ahrens and Kohrt [Ahrens 1981; Bratley et al. 1987]. Note that this effort is independent of N_k, $k = 1, \ldots, K$, the maximum values of the stochastic process! This means that the method has potential to handle multichain queueing networks with large population sizes and loss networks with large link capacities.

It has been repeatedly observed in the Monte Carlo integration literature that the variance σ^2 can often be significantly reduced if importance sampling is employed. The idea here is to choose this sampling distribution $p(\mathbf{n})$, $\mathbf{n} \in \Lambda$, in order to reduce the variance of the estimator G_n. In particular, it is desirable to sample more frequently the points $\mathbf{n}$ at which $q(\mathbf{n})$ is "important," which is typically done by considering functions $p(\cdot)$ that are similar to $q(\cdot)$. Ideally, one would like $q(\cdot)/p(\cdot)$ to be nearly constant; however, there exists a tradeoff between this similarity and the effort required to sample from $p(\cdot)$.

2.1 Ratio Estimators

The method described above can be useful in estimating the normalization constant for a product form stochastic network. However, most performance measures of interest typically take the form

$$\phi := \frac{\sum_{\mathbf{n}\in\Lambda} f(\mathbf{n}) q(\mathbf{n})}{\sum_{\mathbf{n}\in\Lambda} q(\mathbf{n})}, \tag{4}$$

where $f(\cdot)$ is a known function. A natural estimate for ϕ therefore is

$$\Phi_n := \frac{\sum_{i=1}^{n} Y_i}{\sum_{i=1}^{n} Z_i}, \tag{5}$$

where $Y_i := f(\mathbf{V}^i) q(\mathbf{V}^i)/p(\mathbf{V}^i)$ and $Z_i := q(\mathbf{V}^i)/p(\mathbf{V}^i)$.

Although it can be shown that Φ_n converges (almost surely) to ϕ, Φ_n has the undesirable property of being biased. However, this bias diminishes as n becomes large. Moreover, the ratio estimator Φ_n can be made free of bias to order $1/n$ with a simple modification that requires an insignificant amount of additional CPU time (see [Fishman 1978], pp. 55-59).

We should also stress that the confidence interval for $|\Phi_n - \phi|$ can again be constructed as the sampling proceeds. The confidence interval is obtained on line from the sample mean, variance, and covariance of Y_n and Z_n (see [Fishman 1978], p. 59-61), and its width is again proportional to $1/\sqrt{n}$.

3. PRODUCT FORM LOSS NETWORKS

Consider a loss network with links $j = 1, \ldots, J$, where link j has C_j circuits. Suppose that the network supports K classes of calls, where each class is distinguished by its route (i.e., a subset of the J links), its bandwidth requirement (the number of circuits required on each link), and the arrival and service rates. Calls are assumed to arrive according to a Poisson process with class-dependent rate λ_k. Let A_{jk} be the number of circuits required by a class-k call on link j. (In ordinary "single-rate" telephone networks, A_{jk} is equal to 0 or 1.) When a class-k call arrives, it is accepted into the network if the number of busy circuits on link j is $\le C_j - A_{jk}$ for all $j = 1, \ldots, J$; otherwise it is blocked and assumed lost. The holding-time distribution for a class-k call has an arbitrary distribution; denote $1/\mu_k$ for its mean. Also denote $\rho_k := \lambda_k/\mu_k$ for $k = 1, \ldots, K$. Note that this formulation allows us to model narrowband (e.g., voice), wideband (e.g., facsimile, video), point-to-point and conference calls. Further note that the model does not require any restriction on the underlying network topology.

Denote $\mathbf{A}$ for the $J \times K$ dimensional matrix with elements A_{jk}; denote $\mathbf{A}_{\cdot k}$ and $\mathbf{A}_{j\cdot}$ for the kth column and the jth row, respectively, of the matrix $\mathbf{A}$. Let $\mathbf{C} = (C_1, \ldots, C_J)$ be the vector of link capacities. The state of the system is defined by the vector $\mathbf{n} = (n_1, \ldots, n_K)$, where n_k represents the number of class-k calls currently in the system. The set of all possible states is given by

$$\Omega(\mathbf{L}) := \{\mathbf{n} : \mathbf{A}\mathbf{n} \leq \mathbf{C}\}.$$

Denote $\pi(\mathbf{n})$ for the equilibrium probability of being in state $\mathbf{n}$. It is well known (e.g., see Kelly [1986]) that

$$\pi(\mathbf{n}) = \frac{\prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}}{g(\mathbf{C})}, \qquad \mathbf{n} \in \Omega,$$

where

$$g(\mathbf{C}) := \sum_{\mathbf{n} \in \Omega(\mathbf{L})} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}.$$

Most performance measures of interest can be expressed in terms of the normalization constant. For example, the probability that a class-k call is blocked is given by

$$\beta_k = 1 - \frac{g(\mathbf{C} - \mathbf{A}_{\cdot k})}{g(\mathbf{C})},$$

which involves the ratio of two normalization constants. Note that the above equation has the form (4). As another example, the sensitivity of the blocking probabilities with respect to the traffic intensities is given by

$$\frac{\partial \beta_k}{\partial \rho_l} = \frac{g(\mathbf{C} - \mathbf{A}_{\cdot k})g(\mathbf{C} - \mathbf{A}_{\cdot l}) - g(\mathbf{C})g(\mathbf{C} - \mathbf{A}_{\cdot k} - \mathbf{A}_{\cdot l})}{g^2(\mathbf{C})},$$

which is again a simple function of normalization constants.

Let us now apply the ideas of Section 2 to this class of stochastic networks. First consider estimating the normalization constant $g(\mathbf{C})$. If we set $\Omega = \Omega(\mathbf{C})$, $N_k = \min\{C_j/A_{jk} : A_{jk} > 0, j = 1, \ldots, J\}$, and $q_k(n) = \rho_k^n/n!$, then G_n given by (2) is an unbiased estimator for the normalization constant $g(\mathbf{C})$. Let us focus on importance sampling functions of the form

$$p(\mathbf{n}) = \frac{1}{c} \prod_{k=1}^{K} \frac{\gamma_k^{n_k}}{n_k!}, \qquad (6)$$

where

$$c = \prod_{k=1}^{K} \sum_{l=0}^{N_k} \frac{\gamma_k^l}{l!}.$$

(Note that c is easy to calculate.) Then the estimator G_n takes the form

$$G_n = \frac{c}{n} \sum_{i=1}^{n} \prod_{k=1}^{K} (\rho_k/\gamma_k)^{V_k^i} 1(\mathbf{V}^i \in \Omega). \qquad (7)$$

The software implementing the estimator would have two modules. The setup module would calculate and store $(\rho_k/\gamma_k)^n$ for $n = 0, \ldots, N_k$, $k = 1, \ldots, K$. The execution module would generate $\mathbf{V}^1$, $\mathbf{V}^2, \ldots$ and recursively calculate the estimators G_1, $G_2, \ldots$. Note that $O(JK)$ operations would be required, in the worst case, in order to obtain G_{n+1} from G_n.

As an example, consider the star network with four links as shown in the Figure 1. The link capacities are: $C_1 = 90$, $C_2 = 100$, $C_3 = 110$, $C_4 = 120$. We assume that there is traffic between each of the 6 pairs of leaf nodes, with no traffic between a leaf node and the central node. For each pair of leaf nodes we assume two classes of traffic: one class that requires 1 circuit on each of the two links along its route; another class that requires 5 circuits on each of the two links along its route. The routes, bandwidth requirements, and traffic intensities are specified for the 12 classes in Table 1.

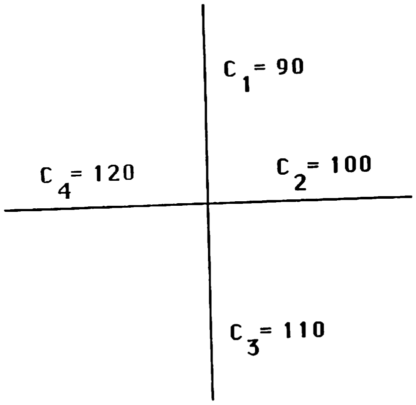

Figure 1. Star Network

Table 1. Network Data

Class	Route	Bandwidith Requirement	Offered Load Light	Moderate	Heavy
1	1,2	1	9.0	10.0	15.0
2	1,3	1	9.0	10.0	15.0
3	1,4	1	9.0	10.0	15.0
4	2,3	1	9.0	10.0	15.0
5	2,4	1	9.0	10.0	15.0
6	3,4	1	9.0	10.0	15.0
7	1,2	5	1.0	2.0	3.00
8	1,3	5	1.0	2.0	3.00
9	1,4	5	1.0	2.0	3.00
10	2,3	5	1.0	2.0	3.00
11	2,4	5	1.0	2.0	3.00
12	3,4	5	1.0	2.0	3.00

There are over a trillion states in this example. None of the known combinatorial or asymptotic techniques apply to this network. Table 2 illustrates the performance of the estimate G_n for light, moderate, and heavy traffic. For each case, we have estimates based on (i) $\gamma_k = \rho_k$, $k = 1, \ldots, K$; (ii) $\gamma_k \neq \rho_k$, $k = 1, \ldots, K$, and (iii) $\gamma_k = \rho_k$, $k = 1, \ldots, K$, with antithetic variates. 95% confidence intervals are also given in the chart for $n = 100,000$. We see from Table 2 that importance sampling and antithetic variates do not provide variance reduction in light traffic. However, both techniques give a small reduction in variance for moderate traffic and a significant reduction for heavy traffic. Note that the importance sampling parameters γ_k, $k = 1, \ldots, 12$, are (slightly) smaller than the corresponding offered loads ρ_k, $k = 1, \ldots, 12$ (for both moderate and heavy traffic cases), which causes the samples $\mathbf{V}^i$ to fall more frequently in $\Omega(\mathbf{L})$.

Table 2. Normalization Constant

Traffic	$\gamma_k = \rho_k$ $(k = 1, \ldots, 12)$	Importance Sampling		Antithetic Variants	
			improvement		improvement
Light[1]	(41682, 41707)	(41682, 41707)	0%	(41682, 41707)	0%
Moderate[2]	(18179, 18212)	(18179, 18211)	3%	(18181, 18213)	3%
Heavy[3]	(33088, 33578)	(33022, 33421)	18%	(33162, 33494)	32%

1. Confidence interval endpoints should be multiplied by 10^{23}

2. Confidence interval endpoints should be multiplied by 10^{27}. For importance sampling the following factors were used: $\gamma_1 = \cdots = \gamma_6 = 9.99$, $\gamma_7 = \cdots = \gamma_{12} = 1.985$.

3. Confidence interval endpoints should be multiplied by 10^{42}. For importance sampling the following factors were used: $\gamma_1 = \cdots = \gamma_6 = 14.7$, $\gamma_7 = \cdots = \gamma_{12} = 2.7$.

Now consider the problem of calculating blocking probabilites via the ratio estimate Φ_n with the importance sampling function (6):

$$\Phi_n = 1 - \frac{\sum_{i=1}^{n} 1[\mathbf{V}^i \in \Omega(\mathbf{C} - \mathbf{A}_{\cdot k})] \prod_{k=1}^{K} (\rho_k/\gamma_k)^{V_k^i}}{\sum_{i=1}^{n} 1[\mathbf{V}^i \in \Omega(\mathbf{C})] \prod_{k=1}^{K} (\rho_k/\gamma_k)^{V_k^i}}. \qquad (8)$$

Tables 3-5 illustrate the performance of the estimator for the network of Figure 1 with $n = 100,000$, again in light, moderate, and heavy traffic. For each case we have estimates based on (i) $\gamma_k = \rho_k$, $k = 1, \ldots, K$; (ii) $\gamma_k \neq \rho_k$, $k = 1, \ldots, K$. In contrast to the results for the normalization constant, importance sampling has given very impressive results for light and moderate traffic, and only a small improvement for heavy traffic. Also note that the importance sampling parameters are now larger than the corresponding offered loads, which causes the samples to fall near the boundary of $\Omega(\mathbf{L})$ more frequently.

Table 3. Percent Blocking for Light Traffic

Class	$\gamma_k = \rho_k$ $(k = 1, \ldots, 12)$	Importance Sampling[1]	Improvement
1	(.040, .069)	(.042, .053)	62%
2	(.038, .066)	(.037, .048)	61%
3	(.037, .065)	(.037, .047)	64%
4	(0, .008)	(.005, .008)	62%
5	(0, .006)	(.004, 007)	50%
6	(0, .003)	(0, .001)	67%
7	(.35, .43)	(.34, .38)	50%
8	(.32, .40)	(.30, .34)	50%
9	(.32, .39)	(.30, .33)	57%
10	(.032, .058)	(.046, .056)	62%
11	(.024, 048)	(.042, .051)	63%
12	(.003, .015)	(.004, .006)	83%

1. For importance sampling the following factors were used: $\gamma_1 = \gamma_2 = \gamma_3 = 9.6$, $\gamma_4 = \gamma_5 = \gamma_6 = 9.5$, $\gamma_7 = \gamma_8 = \gamma_9 = \gamma_{10} = \gamma_{11} = \gamma_{12} = 2.2$, $\gamma_{12} = 2.1$

Table 4. Percent Blocking for Moderate Traffic

Class	$\gamma_k = \rho_k$ $(k = 1, 2 \ldots, 12)$	Importance Sampling[1]	Improvement
1	(0.298, 0.370)	(0.328, 0.375)	35%
2	(0.261, 0.329)	(0.280, 0.325)	34%
3	(0.253, 0.320)	(0.273, 0.318)	33%
4	(0.044, 0.070)	(0.059, 0.075)	48%
5	(0.031, 0.064)	(0.052, 0.067)	55%
6	(0.005, 0.018)	(0.008, 0.012)	69%
7	(2.2, 2.39)	(2.23, 2.37)	26%
8	(1.87, 2.05)	(1.90, 2.03)	28%
9	(1.81, 1.98)	(1.86, 1.99)	24%
10	(0.467, 0.557)	(0.461, 0.509)	47%
11	(0.403, 0.486)	(0.411, 0.458)	43%
12	(0.072, 0.110)	(0.066, 0.079)	66%

1. For importance sampling the following factors were used: $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 10.5, \gamma_5 = 10.4$, $\gamma_6 = 10.2$, $\gamma_7 = \gamma_8 = \gamma_9 = \gamma_{10} = \gamma_{11} = \gamma_{12} = 2.75$

Table 5. Percent Blocking for Heavy Traffic

Class	$\gamma_k = \rho_k$ $(k = 1, 2 \ldots, 12)$	Importance Sampling[1]	Improvement
1	(5.465, 5.910)	(5.307, 5.740)	3%
2	(4.421, 4.825)	(4.227, 4.617)	3%
3	(3.939, 4.321)	(3.756, 4.125)	3%
4	(2.310, 2.608)	(2.300, 2.596)	1%
5	(1.835, 2.102)	(1.836, 2.103)	0%
6	(0.772, 0.949)	(0.715, 0.886)	3%
7	(27.94, 28.80)	(27.76, 28.60)	2%
8	(23.18, 23.99)	(22.95, 23.74)	2%
9	(21.13, 21.93)	(20.92, 21.69)	4%
10	(13.14, 13.80)	(13.19, 13.83)	3%
11	(10.88, 11.49)	(10.91, 11.51)	2%
12	(4.631, 5.043)	(4.558, 4.965)	1%

1. For importance sampling the following factors were used: $\gamma_1 = \gamma_2 = \gamma_3 = 14.9$, $\gamma_4 = \gamma_5 = \gamma_6 = 14.85$, $\gamma_7 = \gamma_8 = \gamma_9 = \gamma_{10} = \gamma_{11} = 2.9, \gamma_{12} = 2.85$

4. MULTICHAIN QUEUEING NETWORKS

It is well known that there is a product form solution for the equilibrium state probabilities for a large class of queueing networks. This class allows for

- Multiple types, or chains, of customers. Customers in different classes can have different routing probabilities and, at non-FCFS service centers, different service distributions. Class hopping between customer classes is also permitted, a feature often used for analyzing precedence constrained sequences of actions in a queueing network.
- Mixed networks, meaning networks with both open and closed chains.
- A variety of service disciplines including infinite server (IS), processor sharing, FCFS, and multiple service stations with concurrent classes of customers [LeBoudec 1986].
- General service distributions at infinite server and processor sharing centers. Exponential service at FCFS centers, where the rates are permitted to depend on the center but not on the class. Load-dependent service rates at each of the nodes.
- A limited form of state-dependent routing [Towsley 1980; Yao 1987].

In order to simplify the notation, we limit our discussion to closed multichain queueing networks where (*i*) no class hopping is permitted; (*ii*) each node is either a FCFS or an IS service center; (*iii*) service times are exponentially distributed and do not depend on the class.

Suppose there are J classes, where class j has N_j customers. Suppose there are M service centers, where each server at center m works at rate μ_m. at center m is denoted by μ_m. Let λ_{jm} be the relative visit ratio of class-j customers at service center m, and denote $\rho_{jm} := \lambda_{jm}/\mu_m$. The state of the system is denoted by $\mathbf{n} := (n_{jm} : 1 \le j \le J, 1 \le m \le M)$, where n_{jm} denotes the number of class j customers at service center m. The set of all possible states is given by

$$\Omega = \{\mathbf{n} : n_{j1} + \cdots + n_{jM} = N_j, \quad j = 1, \ldots, J\}.$$

Denote $\mathbf{n}_m := (n_{jm} : 1 \le j \le J)$ and $n_m = n_{1m} + \cdots + n_{Jm}$. It is well known that the equilibrium probability of being in state $\mathbf{n} \in \Omega$ is given by

$$\pi(\mathbf{n}) = \frac{1}{g} \prod_{m=1}^{M} f_m(\mathbf{n}_m),$$

where

$$f_m(\mathbf{n}_m) = \begin{cases} n_m! \prod_{j=1}^{J} \frac{\rho_{jm}^{n_{jm}}}{n_{jm}!} & \text{if center } m \text{ is FCFS} \\ \prod_{j=1}^{J} \frac{\rho_{jm}^{n_{jm}}}{n_{jm}!} & \text{if center } m \text{ is IS,} \end{cases}$$

and

$$g = \sum_{\mathbf{n} \in \Omega} \prod_{m=1}^{M} f_m(\mathbf{n}_m).$$

Moreover, many performance measures of interest, (including average throughputs, the moments of the queue lengths, and the gradients of these measures) can be expressed as simple functions of normalization constants.

Let us now apply the ideas of Section 2 to this class of networks. We focus our discussion on determining the normalization constant g. The estimator G_n given by (2) becomes

$$G_n = \frac{1}{n} \sum_{i=1}^{n} \frac{q(\mathbf{V}^i)}{p(\mathbf{V}^i)} \tag{9}$$

where $\mathbf{V} = (V_{jm}^i : 1 \le j \le J, \ 1 \le m \le M)$ and

$$q(\mathbf{n}) = 1(\mathbf{n} \in \Omega) \prod_{m=1}^{M} f_m(\mathbf{n}_m), \qquad \mathbf{n} \in \Lambda.$$

Note that for multiclass queueing networks, the sampling distribution $p(\cdot)$ is defined over

$$\Lambda = \Lambda_1 \times \cdots \times \Lambda_J,$$

where $\Lambda_j := \{0, \ldots, N_j\}^M$. Thus in each iteration of the algorithm, JM variates are drawn from a uniform distribution. The JM variates are then transformed so that they correspond to a sample from Λ with distribution $p(\cdot)$.

Importance Sampling Technique #1

Recall that the method of importance sampling is to choose a distribution $p(\cdot)$ from which it is easy to sample and which keeps the variance of $q(\mathbf{V}^i)/p(\mathbf{V}^i)$ to a minimum. One possibility is to choose $p(\cdot)$ so that V_{jm}^i, $1 \le j \le J$, $1 \le m \le M$, are mutually independent. Although such a distribution would be easy to sample from, this may be a bad choice because with high probability $V_{j1}^i + \cdots + V_{jM}^i \ne N_j$, which in turn leads to large variance for the performance estimators.

A more fruitful approach might be to choose a sampling distribution such that $V_{j1}^i + \cdots + V_{jM}^i = N_j$ for all $1 \le j \le J$ and all $i \ge 1$. This could be done with sampling functions of the form

$$p(\mathbf{n}) = \prod_{j=1}^{J} p_j(n_{j1}, \ldots, n_{jM}), \tag{10}$$

where $p_j(\cdot)$ is a distribution over

$$\Omega_j := \{(n_{j1}, \ldots, n_{jM}) : n_{j1} + \cdots + n_{jM} = N_j\}.$$

This would indeed guarantee that $V_{j1}^i + \cdots + V_{jM}^i = N_j$. Note that a sampling distribution of the form (10) calls for independent sampling across classes but dependent sampling across service centers within a given class.

Unfortunately, this method has its drawbacks since the integrand $q(\mathbf{n})/p(\mathbf{n})$ becomes difficult to evaluate. More specifically,

$$\frac{q(\mathbf{n})}{p(\mathbf{n})} = \prod_{m=1}^{M} n_m! \prod_{j=1}^{J} \frac{\rho_{jm}^{n_{jm}}/n_{jm}!}{p_j(n_{j1}, \ldots, n_{jm})}$$

contains the term $\prod_{m=1}^{M} n_m!$, which can become exceedingly large.

Importance Sampling Technique #2

We now discuss another approach to importance sampling, which does not have the integrand evalulation problem discussed above. This method requires at least one node, say node 0, to be an IS center. (It is interesting to note that the method of asymptotic expansions *always* requires that at least one node be an IS service center.)

If for any $\mathbf{n} \in \Omega$ we know $\mathbf{n}_1, \ldots, \mathbf{n}_M$, then we also know $\mathbf{n}_0$

through the relation $n_{j0} = N_j - n_{j1} - \cdots - n_{jM}$. Therefore, the normalization constant can be expressed as

$$g = \sum_{\mathbf{n} \in \Omega'} \prod_{m=1}^{M} f_m(\mathbf{n}_m) \prod_{j=1}^{J} t_j(N_j - n_{j1} - \cdots - n_{jM}),$$

where

$$t_j(n) := \frac{\rho_{0j}^n}{n!}$$

and

$$\Omega' := \{(\mathbf{n}_1, \cdots, \mathbf{n}_M) : n_{j_1} + \cdots + n_{jM} \leq N_j, \;\; 1 \leq j \leq J\}.$$

Note that Ω' has been defined with *inequality* constraints.

Now consider the following importance sampling function $p(\cdot)$. First we choose $n_1, \ldots, n_M$ from a distribution $r(n_1, \cdots, n_M)$. Then for each m, we choose $n_{1m}, \ldots, n_{Jm}$ according to placing n_m balls into J boxes, where the probability that a ball is placed in the jth box is ρ_{jm}/ρ_m, where $\rho_m := \rho_{1m} + \ldots + \rho_{jM}$. Thus

$$p(\mathbf{n}) = r(n_1, \ldots, n_M) \prod_{m=1}^{M} n_m! \prod_{j=1}^{J} \frac{(\rho_{jm}/\rho_m)^{n_{jm}}}{n_{jm}!}, \quad \mathbf{n} \in \Lambda. \tag{11}$$

For the sake of presentation, suppose the nodes 1 through M are FCFS service centers. Then the "integrand" becomes

$$q(\mathbf{n})/p(\mathbf{n}) = 1(\mathbf{n} \in \Omega') \frac{\rho_1^{n_1} \cdots \rho_M^{n_M}}{r(n_1, \ldots, n_M)} \prod_{j=1}^{J} t_j(N_j - n_{j1} - \cdots - n_{jM}). \tag{12}$$

We could further set $r(n_1, \ldots, n_M) = c\gamma^{n_1} \cdots \gamma^{n_M}$ so that (12) becomes

$$q(\mathbf{n})/p(\mathbf{n}) = \frac{1(\mathbf{n} \in \Omega')}{c} \prod_{m=1}^{M} (\frac{\rho_m}{\gamma_m})^{n_m} \prod_{j=1}^{J} t_j(N_j - n_{j1} - \cdots - n_{jM}),$$

which is easy to evaluate. However, other choices of $r(n_1, \ldots, n_M)$ may give better variance reduction.

5. CONCLUDING REMARKS

Monte Carlo summation and importance sampling have been considered for solving large-scale product form stochastic networks. We have shown that for a loss network with a star topology, confidence intervals can be greatly reduced. We have also outlined two importance sampling techniques for multichain product-form queueing networks. In our future research we will investigate (*i*) alternative importance sampling functions for loss networks; (*ii*) indirect estimation via Little's formula; (*iii*) computational testing for queueing networks.

REFERENCES

Ahrens, J.H. and K.D. Kohrt (1981), "Computer Methods for Efficient Sampling from Largely Arbitrary Statistical Distributions," *Computing*, 26, 19-31.

Baskett, F., M. Chandy, R. Muntz, and J. Palacios (1975), "Open, Closed and Mixed Network of Queues with Different Classes of customers," *Journal of Associated Computing Machinery*, 22, 248-260.

Bratley, P., B.L. Fox and L.E. Schrage (1987), *A Guide to Simulation.* Springer-Verlag, New York.

Buzen, J.P. (1973), "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Communications of ACM*, 16, 527-531.

Fishman, G. (1978), *Principles of Discrete Event Simulation.* John Wiley, New York.

Harvey, C. and C.R. Hills (1979), "Determining Grades of Service in a Network," In *9th International Teletraffic Conference.*

Kalos, M and P.A. Whitlock (1986), *Monte Carlo Methods, Volume 1: Basics.* John Wiley, New York.

Kelly, F. (1986), "Blocking Probabilities in Large Circuit-Switched Networks," *Advances in Applied Probability* 18, 473-505.

Kelly, F.P. (1979), *Reversibility and Stochastic Networks.* Wiley, Chichester.

Lam , S.S. and Y.L. Lien (1983), "A Tree Convolution Algorithm for the Solution of Queueing Networks," *Communications of ACM*, 26, 203-215.

LeBoudec, J.Y. (1986), "A BCMP Extension to Multiserver Stations with Concurrent Classes of Customs," In *Performance '86*, pages 78-91.

McKenna, J. and D. Mitra (1982), "Integral Representation and Asymptotic Expansions for Closed Markovian Queueing Networks: Normal Usage. *Bell Systmes Technical Journal* 61, 661-683.

McKenna, J., D. Mitra, and K.G. Ramakrishnan (1981), "A Class of Closed Markovian Queueing Networks: Integral Representations, Asymptotic Expansions, and Generalization, " *Bell Systems Technical Journal*, 60, 559-641.

Mitra, D. (1987), "Asymptotic Analysis and Computational Methods for a Class of Simple, Circuit-Switched Networks with Blocking, *Adv. Appl. Prob.*, 19, 219-239.

Mitra, D. and J. McKenna (1986), "Asymptotic Expansions for Closed Markovian Queueing Networks with State Dependent Service Rates, *Journal for the Association of Computing Machinery*, 33, 568-592.

Ramakrishnan, K.G. and D. Mitra (1986), "An Overview of PANACEA, a Software Package for Analyzing Markovian Queueing Networks," *Bell Systems Technical Journal* 61, 2849-2872.

Reiser, M. and H. Kobayashi (1975), "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," *IBM J. of Research Development.* 19, 283-294.

Reiser, M. and S.S. Lavenberg (1980), "Mean-Value Analysis of Closed Multichain Queueing Networks, *Journal of Assocation for Computing Machinery.* 27, 313-322.

Ross, K.W. and D. Tsang, "Teletraffic Engineering for Product-form Circuit-Switched Networks," to appear in *Advances in Applied Probability.*

Sauer, C.H. (1983), "Computational Algorithms for State-Dependent Queueing Networks," *ACM Trans. Computing Systems*, 1, 67-92.

Towsley, D.F. (1980), "Queueing Network Models with State-Dependent Routing," *Journal of Association for Computing Machinery* 27, 323-337.

Tsang, D. and K.W. Ross, "Algorithms for Determining Exact Probabilities in Tree Networks," To appear in *IEEE Transactions on Communications.*

Yao , D.D. and J.A. Buzacott. (1987), "Modeling a Class of Flexible Manufacturing Systems with Reversible Routing," *Operations Research*, 35-87.