Wharton

Financial
Institutions
Center

Computing Performance Measures in a Multi-Class Multi-Resource Processor-Shared Loss System

by
O. Zeynep Akşin
Patrick T. Harker

96-02-C

The Wharton School
University of Pennsylvania

THE WHARTON FINANCIAL INSTITUTIONS CENTER

The Wharton Financial Institutions Center provides a multi-disciplinary research approach to the problems and opportunities facing the financial services industry in its search for competitive excellence. The Center's research focuses on the issues related to managing risk at the firm level as well as ways to improve productivity and performance.

The Center fosters the development of a community of faculty, visiting scholars and Ph.D. candidates whose research interests complement and support the mission of the Center. The Center works closely with industry executives and practitioners to ensure that its research is informed by the operating realities and competitive demands facing industry participants as they pursue competitive excellence.

Copies of the working papers summarized here are available from the Center. If you would like to learn more about the Center or become a member of our research community, please let us know of your interest.

Anthony M. Santomero
Director

The Working Paper Series is made possible by a generous
grant from the Alfred P. Sloan Foundation

Computing Performance Measures in a Multi-Class Multi-Resource Processor-Shared Loss System [1]

December 1996

Revised: October 1998

Abstract: This paper develops methods to compute performance measures in a specific type of loss system with multiple classes of customers sharing the same processor. Such systems arise in the modeling of a call center, where the performance measures of interest are blocking the probability of a call and the reneging probability of customers that are put on hold. Expressions for these performance measures have been derived in previous work by the authors. Given the difficulty of computing these performance measures for realistic systems, this paper proposes two different approaches to simplify this computation. The first method introduces the idea of multi-dimensional convolutions, and uses this approach to compute exact blocking and reneging probabilities. The second method establishes an adaptation of the Monte Carlo summation technique in order to obtain good estimates of blocking and reneging probabilities in large systems along with their associated confidence intervals.

Keywords : Queuing; loss system; processor sharing; computational analysis; approximation

[1] Zeynep Akşin is at INSEAD, Technology Management Area, Boulevard de Constance, 77305 Fontainebleau, France.

Patrick Harker is at the Department of Operations and Information Management, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104-6366, harker@opim.wharton.upenn.edu

1 Introduction

Loss systems arise in the modeling of telecommunication and data networks, as well as certain service and manufacturing operations. This paper considers a particular type of loss system that has emerged from a study of call center operations in financial service firms (Akşin and Harker 1996a). While this is the application that motivates the analysis herein, loss systems with similar characteristics are encountered elsewhere in the modeling of telecommunication systems (Ross 1995; De Waal and Van Dijk 1991). The distinguishing features of the model are (i) the presence of multiple resources that define capacity, (ii) multiple call types that result in multiple access channels, and (iii) the handling of calls that follow a processor sharing discipline. The growing importance of product and service customization, the extensive use of technology that automates parts of operations, and the trend in consolidating information systems (i.e., data warehousing) motivates the multi-class, multi-resource, processor-shared characteristics of the model described herein.

As the application of the model to the call center management problem in Akşin and Harker (1996a,b) indicates, the use of the model to determine performance is restricted by one's ability to compute performance measures in large systems. The presence of normalization constants in the expressions for blocking and reneging probabilities gives rise to summations over typically very large state spaces, eliminating the possibility of computation via brute-force methods. This paper addresses this problem by describing two different approaches to develop effective and efficient computational schemes for the normalization constants.

In earlier literature, several approaches have been adopted to deal with this problem. Buzen (1973), Lam and Lien (1983), Reiser and Kobayashi (1975) tackle the problem by using specialized recursive algorithms in the context of certain queueing networks. Exploiting special structures in specific loss networks, Kaufman (1981), Roberts (1981), and Tsang and Ross (1990) develop efficient combinatorial algorithms to compute normalization constants and blocking probabilities. The use of integral representations and their asymptotic expansions constitutes the basic idea for the technique developed by McKenna *et al.* (1981) and Mitra (1987). Ross and Wang (1992) demonstrate an application of Monte Carlo summation and importance sampling to product-form summations; it turns out that this method can be successfully applied to a diverse set of networks with product-form solutions.

The analysis in this paper will follow the combinatorial approach in Tsang and Ross (1990) as well as the Monte Carlo summation approach in Ross and Wang (1992). After a brief overview of the model in Section 2, the two methods for computing performance measures will be explored. In particular, Section 3 shows that the expressions for blocking probabilities can be transformed into a function of single or multi-dimensional convolutions, which then enable the use of convolution based computational algorithms. A similar approach for the computation of renege probabilities is proposed. In Section 4, it is shown how one can implement the Monte Carlo summation method in the context of the loss model described in this paper. Section 5 provides numerical examples for small systems that compare exact renege probabilities to those obtained by the Monte Carlo Summation method. The paper ends with a discussion of future research directions.

2 Preliminaries

The operations of a phone center has been modeled earlier in Akşin and Harker (1996a). Capacity of this phone center is a stochastic entity, which is a function of demand and resource allocations. Resources that jointly determine capacity are human resources in the form of service agents, telecommunication resources such as phone lines and VRUs (voice response units), and information technology resources. A customer call will require the availability of a phone line, through which the call can gain access to a service representative or a VRU. At the same time, the representative will need access to certain applications or databases in order to provide the requested services. The distinguishing characteristic of the model is the information processing resource that is shared across multiple call types. As call centers have increased the intensity with which they use this resource, its role in determining system capacity has become critical.

In the sequel, this multi-channel queueing system with processor sharing is used as the performance model for a phone center. The specific assumptions underlying the proposed performance model are summarized below. The reader is referred to Akşin and Harker (1996a) for a detailed exposition and analysis of this model. Three different ways of measuring performance can be used, each based on different assumptions regarding customer behavior and the system configuration. In the basic case, which is called the *loss system*, it is assumed that customers are extremely impatient. Hence, any customer who cannot initiate service immediately will leave the system. It is assumed that all customers who leave are lost demand and will not retry until their next transaction. In this configuration of the system, the number of trunks or phone lines are equal to the number of service representatives. Next, consider a system which may have phone lines in excess of the number of service representatives. Upon arrival of a call, if all trunks are taken, the customer receives a busy signal and leaves the system. On the other hand, if a trunk is available but all agents are busy, the customer is put on hold. The case which assumes that customers on hold will always wait for service initiation is called the *queueing system*. While some customers wait until an agent becomes available, some customers may exhibit impatience and leave the system while on hold before service initiation. This loss of customers is labeled as reneges and the system is called the *system with reneges*. For many inbound call centers, the renege system will constitute the most realistic model.

Consider a phone center with K access channels. Each access channel consists of T_k, $k = 1, \ldots, K$ phone trunks and S_k, $k = 1, \ldots, K$ service agents specializing in product line k, with $T_k \geq S_k$. Customers arrive at the various access channels with an arrival rate of λ_k, where arrivals in each channel are independent of each other and the arrival process is assumed to be Poisson. Upon service initiation, the service representative will need access to the information system. This joint pool of information technology is capable of processing all transactions from different customers simultaneously. Notice that during times of high congestion, such central information systems respond with longer processing times. In other words, service times in the system are a function of the total number of customers being served in all channels. This characteristic is modeled as a processor sharing service discipline.

Let the information system be considered to be a single server that processes at a constant rate of one service unit per unit time. Assume that each customer in class k with $k = 1, \ldots, K$ has a service requirement that is exponentially distributed with an average of $1/\mu_k$. Then, letting $\mathbf{n} = (n_1, n_2, \ldots, n_K)$ denote the state vector, with n_k being the number of customers of class k in the system, the state dependent service rate for class k customers in the processor-shared system takes the form

$$\mu_k(\mathbf{n}) = \frac{n_k \; \mu_k}{(n_1 + \ldots + n_K)}$$

for the loss system, and the form

$$\mu_k(\mathbf{n}) = \frac{min(n_k, S_k) \; \mu_k}{(min(n_1, S_1) + \ldots + min(n_K, S_K))}$$

for the queueing system and the system with reneges. Note that the model assumes simultaneous use of the service representative and the information processing resource throughout the duration of the call. While in its basic form, as considered in this paper, it is assumed that the labor content and computer content of a call are equal to each other, a minor modification of the state dependent service rate allows for the case of call centers where the computer content of a call is less than its labor content. This case is discussed in more detail in Akşin and Harker (1996a). Define $\pi(\mathbf{n})$ as the equilibrium probability of being in state $\mathbf{n}$ (i.e., of having n_k customers of class k in the system). Define the sets $\mathcal{A} = \{\mathbf{n} \in \mathcal{Z}_+^K : n_k \leq T_k\}$ and $\mathcal{A}_k = \{\mathbf{n} \in \mathcal{A} : n_k < T_k\}$, where $\mathcal{Z}_+$ denotes the nonnegative integers. Finally, $\mathbf{e}_k$ is a K-dimensional vector of zeros with a one in its kth position and $\mathbf{0}$ is a K-dimensional vector of zeros.

In order to characterize the performance of the phone center, one must establish the behavior of the system in steady state. To this end, one must first determine the equilibrium distributions, $\pi(\mathbf{n})$, for the two systems being considered. In general, this distribution takes the form

$$\pi(\mathbf{n}) = \frac{\psi(\mathbf{n})}{\sum_{\mathbf{n}\in\mathcal{A}} \psi(\mathbf{n})} \tag{1}$$

where $G = \sum_{\mathbf{n}\in\mathcal{A}} \psi(\mathbf{n})$ is known as the *normalization constant.* For the derivation of equilibrium distributions in the loss and renege systems, the interested reader is referred to Akşin and Harker (1996a); these results are presented below without proof. For the loss system, one obtains the equilibrium distribution as

$$\pi(\mathbf{n}) = \frac{1}{G}(n_1 + \ldots + n_K)! \prod_{k=1}^{K} \frac{(\rho_k)^{n_k}}{n_k!} \tag{2}$$

where $\rho_k = \lambda_k/\mu_k$. For the queueing system, using the convention that $\sum_a^b x = 0$ and $\prod_a^b x = 1$ if $b < a$, this expression takes the form

$$\psi(\mathbf{n}) = (\sum_{k=1}^{K} \min(n_k, S_k))! \prod_{k=1}^{K} \left\{ \frac{(\rho_k)^{n_k} (\sum_{k=1}^{K} \min(n_k, S_k))^{(n_k - S_k)^+}}{(\min(n_k, S_k))!(S_k)^{(n_k - S_k)^+}} \right\}, \tag{3}$$

where

$$a^{+} = \max(0, a).$$

For the reneging system, the added customer loss due to call abandonments while on hold needs to be incorporated. To this end, the time a customer waits in queue k is assumed an exponential random variable with rate α_k. This implies a renege rate of $r_k(n_k) = \alpha_k(n_k - S_k)1(S_k < n_k \leq T_k)$ for $k = 1, \ldots, K$. Letting $\tau_k(j, \mathbf{n}) = \mu_k \ \min(j, S_k) + r_k(j)(\sum_{k=1}^{K} \min(n_k, S_k))$ for more compact notation, it can be shown that

$$\psi(\mathbf{n}) = (\sum_{k=1}^{K} \min(n_k, S_k))! \prod_{k=1}^{K} \frac{\lambda_k^{n_k} (\sum_{k=1}^{K} \min(n_k, S_k))^{(n_k - S_k)^{+}}}{\prod_{j=1}^{n_k} \tau_k(j, \mathbf{n})}. \tag{4}$$

The equilibrium distribution is then given by

$$\pi(\mathbf{n}) = \frac{\psi(\mathbf{n})}{\sum_{\mathbf{n} \in \mathcal{A}} \psi(\mathbf{n})}.$$

Given the equilibrium distributions, the next step is to determine revenue losses that result from congestion in the system. To this end, certain performance measures need to be established. Specifically, one would be interested in determining the probability of a customer being blocked upon arrival, as well as the loss that occurs due to reneging. In general, blocking probability in channel k is given by

$$B_k = 1 - \frac{\sum_{\mathbf{n} \in \mathcal{A}_k} \pi(\mathbf{n})}{\sum_{\mathbf{n} \in \mathcal{A}} \pi(\mathbf{n})}. \tag{5}$$

Reneges are the second source of customer loss, made up by the portion of customers that are lost after entering the system. Denote the long-run probability of renege for a customer of type k by R_k. Then,

$$R_k = \sum_{\mathbf{n} \in \mathcal{A}} \pi(\mathbf{n}) \frac{r_k(n_k)}{\lambda_k(1 - B_k)}. \tag{6}$$

Note that obtaining these probabilities requires the calculation of a normalization constant G, which involves summing the expressions given in equations (2), (3), and (4), over a state space that can typically be very large. The following two sections address this difficulty and develop computational schemes that significantly reduce the complexity of determining performance measures in these loss sytems.

3 Calculating Performance Measures Using Convolutions

In this section, appropriate transformations to the expressions for G in the loss and queueing systems will be made, enabling the implementation of a convolution based algorithm. For the reneging system, a similar analysis based on the idea of multi-dimesnional convolutions yields exact results for blocking probabilities and reneging probabilities.

3.1 The Loss System

Let us start by analyzing the performance of the loss system. Recall that the normalization constant is given by $G = \sum_{\mathbf{n} \in \mathcal{A}} \psi(\mathbf{n})$. Similarly, define $G_k = \sum_{\mathbf{n} \in \mathcal{A}_k} \psi(\mathbf{n})$; then, the blocking probability of a type k customer can be written as

$$B_k = 1 - \frac{G_k}{G}. \tag{7}$$

Let $\otimes$ denote the convolution operator. Also, define the total number of service agents in the system as $S = \sum_{k=1}^{K} S_k$ and total number of trunks as $T = \sum_{k=1}^{K} T_k$. The convolution of two vectors $\mathbf{g_1} = [g_1(0), g_1(1), \ldots, g_1(S)]$ and $\mathbf{g_2} = [g_2(0), g_2(1), \ldots, g_2(S)]$ is given by

$$[\mathbf{g_1} \otimes \mathbf{g_2}](s) = \sum_{j=0}^{s} g_1(j) g_2(s-j), \qquad s = 0, 1, \ldots, S. \tag{8}$$

Using these definitions, if

$$g_k(s) = \frac{\rho_k^s}{s!} 1(s \leq S_k), \tag{9}$$

then the sum of equation (2) over the entire state space can be written as

$$G = \sum_{s=0}^{S} s![\mathbf{g_1} \otimes \ldots \otimes \mathbf{g_K}](s). \tag{10}$$

A similar simplification leads to the expression

$$G_k = \sum_{h=0}^{S_k-1} \mathbf{g_k}(h) \sum_{s=0}^{S-1-h} (h+s)!\mathbf{g}_{(k)}(s), \tag{11}$$

where $\mathbf{g}_{(k)}$ denotes the convolution of all vectors $\mathbf{g}_j$ with $j = 1, \ldots, K$ and $j \neq k$.

Consider a system where $S_k = \bar{S}$ for all k. Then, calculation of blocking probabilities using a brute-force summation to determine the normalization constant requires a computation of order $O(K^2\bar{S}^K)$. This computation clearly grows exponentially in problem size. Let us now consider a binary tree implementation of the convolution algorithm to compute blocking probabilities (Tsang and Ross, 1990). Without loss of generality, let $K = 2^{l-1}$ for any integer l. Using the convolution scheme, and noticing that $\sum_k S_k = K\bar{S}$, computation of blocking probabilities becomes of order $O(K(K\bar{S})^2 logK)$ or $O(K^3(\bar{S})^2 logK)$.

3.2 The Queueing System

For the queueing and renege systems, the "min" term in the expression for state dependent service rates, $\mu_k(\mathbf{n})$, creates difficulty when one tries to collapse the multiple summations for each class into a single summation as in (10). In particular, note that the terms involving $\sum_{k=1}^{K} \min(n_k, S_k)$ cannot be put in a product form, hence preventing a separation by class as required for the convolution scheme. Furthermore, given the total number of customers in the system (s), it is not obvious what will be the value of the expression $\sum_{k=1}^{K} \min(n_k, S_k)$. To overcome this difficulty, introduce an additional dimension in our convolution operator. Note that for the loss system, the vectors that are convolved ($g_k(s)$) are indexed by s, the total number of customers in the system. For the queueing system, define vectors that are indexed by two numbers s and b, where the latter will denote the total number of customers on hold. This definition allows one to extend the idea used for the analysis of the loss system to that of the queueing system, as described below.

Partition the state space such that

$$\mathcal{A}_{sb} = \{\mathbf{n} \in \mathcal{A} : \sum_{k=1}^{K} n_k = s, \sum_{k=1}^{K} (n_k - S_k)^+ = b, s > b\};$$

that is, the set of states for which there are exactly s customers in the system and exactly b of these are on hold. Based on the earlier definition, one has

$$\mathcal{A} = \cup_{b=0}^{T-S} \cup_{s=b}^{T} \mathcal{A}_{sb}.$$

Recall that the normalization constant is given by

$$G = \sum_{\mathbf{n} \in \mathcal{A}} \psi(\mathbf{n}).$$

Then, using the definition for $\mathcal{A}_{sb}$, this can be restated as

$$G = \sum_{b=0}^{T-S} \sum_{s=b}^{T} \sum_{\mathbf{n} \in \mathcal{A}_{sb}} \psi(\mathbf{n}). \tag{12}$$

Recalling the expression for $\psi(\mathbf{n})$ and the fact that $\sum_{k=1}^{K} \min(n_k, S_k) = s - b$ for all $\mathbf{n} \in \mathcal{A}_{sb}$, (12) can be rewritten as

$$G = \sum_{b=0}^{T-S} \sum_{s=b}^{T} (s-b)!(s-b)^b \sum_{\mathbf{n} \in \mathcal{A}_{sb}} \prod_{k=1}^{K} \frac{(\rho_k)^{n_k}}{(\min(n_k, S_k))!(S_k)^{(n_k - S_k)^+}}. \tag{13}$$

It is now clear that the innermost summation in (13) can be expressed as a convolution just as in the case for the loss system. To this end, redefine the convolution operator given in (8), so that one has a two dimensional convolution

$$[\mathbf{g_1} \otimes \mathbf{g_2}](s, b) = \sum_{n=0}^{b} \sum_{j=n}^{s} g_1(j, n) g_2(s-j, b-n), \quad s = 0, 1, \ldots, T, \ \ b = 0, 1, \ldots, T-S. \tag{14}$$

Similarly, (9) is redefined as

$$g_k(s,b) = \begin{cases} \frac{\rho_k^s}{s!} & \text{if } s \leq S_k, \text{ and } b = 0 \\ \frac{\rho_k^s}{S_k! S_k^b} & \text{if } S_k < s \leq T_k, \text{ and } b \leq T_k - S_k \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Using the new definitions for $g_k(s,b)$, and the convolution operator in (14) within the first two summations in expression (13), one obtains the normalization constant for the queueing system as

$$G = \sum_{b=0}^{T-S} \sum_{s=b}^{T} (s-b)!(s-b)^b [\mathbf{g_1} \otimes \ldots \otimes \mathbf{g_K}](s,b). \tag{16}$$

An analogous argument allows one to state the corresponding G_k as

$$G_k = \sum_{m=0}^{(T_k-S_k-1)} \sum_{h=m}^{(T_k-1)} \mathbf{g_k}(h,m) \sum_{n=0}^{(T-S-m-1)} \sum_{j=n}^{(T-1-h)} (j+h-m-n)!(j+h-m-n)^{n+m} \mathbf{g}_{(k)}(j,n). \tag{17}$$

While this computation is more complex than the one that is required for the loss system, one would nevertheless expect it to perform better than a brute-force summation for growing problem size. To compare the two, consider a system where $T_k = \bar{T}$ for all k. For the brute-force computation, the complexity can be expressed as $O(K^2 \bar{T}^K)$. The same computation using the convolution scheme, again with a binary tree implementation, is easily seen to be of order $O(K(K\bar{T}(K\bar{T} - K\bar{S}))^2 logK)$ or equivalently $O(K^5(\bar{T}(\bar{T} - \bar{S}))^2 logK)$. Note that while this does not grow exponentially in K, performance does deteriorate rapidly for large K. One may need to resort to a different approach to deal with larger problems; one potential method will be explored in Section 4.

3.3 The System with Reneges

Calculation of blocking probabilities in the renege system require summations involving the expression in (4). Again, observe the presence of the same complicating terms encountered in the case with no reneges. In addition to these terms, observe that τ_k is defined as a function of $\sum_{k=1}^{K} \min(n_k, S_k)$, making the expression further non-separable as a product in k. It will now be demonstrated how the idea used for the queueing case can be implemented in the renege case. This time, it is convenient to introduce a third dimension p, denoting the number of customers in service in the entire system (all classes). The vectors g_k are now defined as

$$g_k(s,b,p) = \begin{cases} \frac{\lambda_k^s}{\prod_{j=1}^{s} \tau_k(j,p)} & \text{if } s \leq T_k, \;\; b \leq T_k - S_k, \;\; (s-b)^+ \leq p \leq S \\ 0 & \text{otherwise,} \end{cases} \tag{18}$$

where

$$\tau_k(j,p) = \begin{cases} \mu_k \min(j, S_k) + r_k(j)p & j \leq T_k, \;\; j \leq p \leq S \\ 1 & \text{otherwise.} \end{cases}$$

The convolution operator for this case takes the form

$$\begin{aligned} [\mathbf{g_1} \otimes \mathbf{g_2}](s,b,p) &= \sum_{n=0}^{b} \sum_{j=n}^{s} g_1(j,n,p) g_2(s-j, b-n, p), \\ & \qquad s = 0,1,\ldots,T, \;\; b = 0,1,\ldots,T-S, \;\; p = 0,1,\ldots,S. \end{aligned}$$

Note that this is identical to (14) except that we now have to perform a convolution for each $p = 0,1,\ldots,S$. The expressions for G and G_k take a similar form to those in the queueing case; however, one now uses the revised definition of the convolution operator. Specifically, we get

$$G = \sum_{b=0}^{T-S} \sum_{s=b}^{T} (s-b)!(s-b)^b [\mathbf{g_1} \otimes \ldots \otimes \mathbf{g_K}](s,b,s-b), \tag{19}$$

and

$$G_k = \sum_{m=0}^{(T_k-S_k-1)} \sum_{h=m}^{(T_k-1)} \mathbf{g_k}(h,m,h-m) \sum_{n=0}^{(T-S-m-1)} \sum_{j=n}^{(T-1-h)} (J)!(J)^{n+m} \mathbf{g}_{(k)}(j,n,j-n), \tag{20}$$

where $J = j+h-m-n$. The argument used to derive these expressions parallels that for the queueing case and is not repeated here. The complexity of calculating blocking probabilities using a binary tree implementation becomes $O(K^6(\bar{T}(\bar{T}-\bar{S}))^2 logK)$.

For all three systems, one has been able to transform the expressions for the normalization constants G and the term G_k into summations of a vector of convolutions. Refined convolution algorithms that exploit the similarity between the computation required to determine G and G_k have been developed by Tsang and Ross (1990). A straightforward adaptation of their algorithm which uses a binary tree implementation can be used to compute blocking probabilities. Their numerical experiments indicate that while this is not the fastest method in terms of CPU time required, it seems to be the most reliable in terms of numerical errors. Since the ultimate goal of the proposed model is that it be used within an optimization model, as in Akşin and Harker (1996c), it is important to choose the implementation with the least potential for such numerical errors. Other implementations are possible for other applications. The complexity analysis given above indicates that the performance of the algorithms deteriorates fast in the number of classes for the queuing and renege cases. Depending on the problem sizes one needs to deal with, and the desired computational times, one may need to refine these algorithms or resort to a non-exact computation using Monte-Carlo summation.

Reneges are the second source of customer loss, so in addition to blocking probabilities, one needs to determine the portion of customers that are lost after entering the system. Denote the long-run probability of renege for a customer of type k by R_k. Then,

$$R_k = \sum_{\mathbf{n} \in \mathcal{A}} \pi(\mathbf{n}) \frac{r_k(n_k)}{\lambda_k(1 - B_k)}$$

which can equivalently be stated as

$$R_k = \sum_{\mathbf{n} \in \mathcal{A}} \psi(\mathbf{n}) \frac{r_k(n_k)}{(\lambda_k G_k)}. \tag{21}$$

The computation of R_k essentially involves a weighted sum of all the $\psi(\mathbf{n})$'s, where the weights constitute the only difference between this and the computation of the normalization constant G.

Using the weights $r_k(n_k)/\lambda_k$, it follows that R_k can be obtained as

$$R_k = \frac{1}{G_k} \sum_{m=0}^{(T_k-S_k)} \sum_{h=m}^{T_k} \mathbf{g_k}(h, m, h-m) \sum_{n=0}^{(T-S-m)} \sum_{j=n}^{(T-h)} \frac{r_k(h)\;(J)!\;(J)^{n+m}}{\lambda_k} \mathbf{g}_{(k)}(j, n, j-n), \qquad (22)$$

where $J = j + h - m - n$.

Given the similarity between this expression and the expression for G_k in (20), it is clear that a refined convolution algorithm as discussed above can be used to compute R_k.

4 The Monte-Carlo Summation Technique

In Akşin and Harker (1996b), data from a retail banking phone center is presented and, subsequently, performance measures for this center are evaluated. A quick look at this data indicates that with approximately two hundred shared 800-trunks, one hundred fifty VRU trunks, and nine departments in the center, computing the normalization constant and performance measures for this center constitute a major challenge. The most appropriate form of the model for this analysis is identified as the system with reneges. This implies that the combinatorial algorithms in Section 3 require a three dimensional convolution of vectors whose size is determined by the number of trunks, two hundred in this instance. Since the trunks are shared among departments for this particular call center, this indicates that the state space determined by the set $\mathcal{A}$ is even larger compared to a center that has devoted phone lines for each department. These characteristics of the problem indicate the need for a different type of computational scheme. In this section, a brief overview of the Monte Carlo summation technique along with a description of its implementation in the phone center context of this paper is provided.

Monte Carlo summation and integration has been suggested as a useful tool in evaluating performance measures for both queueing and loss networks (Ross *et al.* 1993; Ross and

Wang 1992). An overview of the method in the context of general loss networks is provided in Ross (1995). The method entails randomly sampling from a product form solution over the state space and then taking averages to obtain consistent estimators for performance measures. It can be adapted to arbitrary product form solutions and, in that sense, is more flexible than the earlier proposed combinatorial methods. The performance of the method is assessed based on the computational effort required to generate a sample and the variance of the estimates. In what follows, the method is adapted to the multi-class, multi-resource, processor-shared loss system.

Let $\mathbf{V}^i = (V_1^i, \ldots, V_K^i), \quad i = 1, 2, \ldots,$ be a sequence of i.i.d. random vectors where $\mathbf{V}^i$ denotes the ith sample. Each $\mathbf{V}^i$ has probability density function $p(\mathbf{n}) := P(\mathbf{V}^i = \mathbf{n})$, where $\mathbf{n}$ can take values in $\mathcal{A} = \{\mathbf{n} \in \mathcal{Z}_+^K : n_k \leq T_k\}$. Letting

$$q(\mathbf{n}) := \psi(\mathbf{n}),$$

one can define

$$Z^i := \frac{q(\mathbf{V}^i) 1(\mathbf{V}^i \in \mathcal{A})}{p(\mathbf{V}^i)}. \tag{23}$$

Then, an unbiased estimator for the normalization constant G is given by

$$\bar{Z}_L := \frac{1}{L} \sum_{i=1}^{L} Z^i, \tag{24}$$

where L denotes the total number of iterations or samples taken in the Monte Carlo method.

For the current implementation, a sampling function is proposed that takes the form

$$p(\mathbf{n}) = \prod_{k=1}^{K} p_k(\mathbf{n}), \tag{25}$$

where each $p_k(\mathbf{n})$ is given by

$$p_k(\mathbf{n}) = \frac{1}{c_k} \frac{\gamma_k^{n_k} \delta_k^{(n_k - S_k)^+}}{\prod_{j=1}^{n_k} \tau_k(j, \mathbf{n})}. \tag{26}$$

The term c_k represents the normalization constant for p_k.

The variance of Z^i, σ^2, is a critical determinant of the effectiveness of the Monte Carlo Summation technique. Choice of the appropriate sampling function $p(\mathbf{n})$ is known to reduce this variance significantly. In particular, it has been shown that a sampling function $p(\mathbf{n})$ that closely resembles $q(\mathbf{n})$ in shape yields estimates with lower variance. This idea of sampling from those parts of $q(\mathbf{n})$ with higher "importance" is known as importance sampling. In (26), γ_k is a positive real number known as the importance sampling parameter. By adjusting the value of this parameter, one can modify the shape of $p(\mathbf{n})$, thus trying to increase the similarity between $p(\mathbf{n})$ and $q(\mathbf{n})$.

Another factor determining the effectiveness of the Monte Carlo summation technique is the ease with which samples are generated from a sampling distribution $p(\mathbf{n})$. The choice of sampling function in (25) ensures that $V_1^i, V_2^i, \ldots, V_K^i$ are independent. This enables use of the alias algorithm to generate each V^i in O(K) time (Bratley *et al.* 1987). Recall that the product form solution in (6) does not have independence between the different classes. This is a result of the term $\sum_{k=1}^{K} min(n_k, S_k)$ that appears in various parts of (6). Thus, to simplify the sampling procedure, an approximation to (6) is used as the sampling function. In (26), δ_k is a parameter that attempts to approximate the term $\sum_{k=1}^{K} min(n_k, S_k)$ in (6). For the results reported herein, δ_k has been set as

$$\min(V_k^i, S_k) + \sum_{j \neq k} \rho_j,$$

where $\rho_j = \gamma_j/\mu_j$. This choice for δ_k can be motivated as follows. For each class k, at every iteration i, V_k^i is known as an estimate for n_k. For all other classes, ρ_k is taken as an approximation for the term $\min(n_k, S_k)$. In a pure loss system, $\rho_k = \lambda_k/\mu_k$ is a good approximation for the average number of customers of class k in service. In an attempt to use a simple approximation for the number of customers being served in a renege system, the same approximation is used with γ_k replacing λ_k. In the application of the method to a call center problem in Akşin and Harker (1996b), a slight modification is made to δ_k for some k. In that paper's context, some classes are served by VRUs as opposed to servers. It turns out that for these classes, reneges can also occur during service. To account for this difference, the term δ_k for the VRU channels is determined by setting $\rho_k = \gamma_k/(\mu_k + \alpha_k)$.

The performance measures that one would like to estimate can be expressed as ratios, as in (5) and (6), which are nonlinear functions of the normalization constants. In general, both acceptance $(1 - B_k)$ and renege probabilities take the form of a ratio:

$$\phi := \frac{\sum_{\mathbf{n}\in\mathcal{A}} f_1(\mathbf{n})q(\mathbf{n})}{\sum_{\mathbf{n}\in\mathcal{A}} f_2(\mathbf{n})q(\mathbf{n})} \tag{27}$$

with $f_1(\cdot)$ and $f_2(\cdot)$ as some known functions. Note that for acceptance probabilities $f_1(\mathbf{n}) = 1(\mathbf{n} \in \mathcal{A}_k)$ and $f_2(\mathbf{n}) = 1$, while for renege probabilities

$$f_1(\mathbf{n}) = \frac{r_k(n_k)}{\lambda_k},$$

and $f_2(\mathbf{n}) = 1(\mathbf{n} \in \mathcal{A}_k)$. Using the Monte Carlo technique, an estimate for ϕ can be obtained as

$$\Phi_L := \frac{\sum_{i=1}^{L} Y^i}{\sum_{i=1}^{L} Z^i} \tag{28}$$

where $Y^i := f_1(\mathbf{V}^i)q(\mathbf{V}^i)/p(\mathbf{V}^i)$ and $Z^i := f_2(\mathbf{V}^i)q(\mathbf{V}^i)/p(\mathbf{V}^i)$. $100(1-\alpha)\%$ confidence intervals for the acceptance probability $1 - B_k$ and renege probability R_k can easily be constructed as sampling proceeds. The interval estimators proposed in Ross (1995) are used in the phone center context.

5 Numerical Examples

In the Monte Carlo summation method, one obtains approximate blocking and renege probabilities. Before using the method, one would like to have a sense of performance with respect to exact blocking and renege probabilities. This section provides a set of numerical examples that explore the performance of the method in terms of its accuracy. Performance in terms of computation time is not explored in this paper. The computation time for the Monte Carlo summation method changes as a function of the number of iterations of the algorithm one chooses to perform. One can, however, state that while computation time performance for the convolution scheme is expected to deteriorate with an increase in the number of classes, the Monte Carlo summation method can deal with systems with a high number of classes more effectively.

Since the objective is to test for accuracy rather than speed, examples of identical size parameters are considered. In particular, all of the examples deal with systems with three customer types ($K = 3$), five trunks reserved for each class of customer ($\mathbf{T} = (5, 5, 5)$), and two servers for each class ($\mathbf{S} = (2, 2, 2)$). Two sets of arrival rates, $\lambda^{\mathbf{1}} = (1.0, 1.0, 1.0)$ and $\lambda^{\mathbf{2}} = (10.0, 10.0, 10.0)$ are considered. Table 1 tabulates renege rate and service rate parameters for each example problem. The problems have been labeled in a way that reflects the degree of heterogeneity in their renege rate and service rate parameters. In particular, the first letter stands for the degree of heterogeneity between α_1, α_2, and α_3, where L denotes low and H denotes high heterogeneity. Similarly, the second letter denotes the degree of heterogeneity between μ_1, μ_2, and μ_3. Two versions of LL and HH examples were tested, denoted by subscripts 1 and 2.

Exact blocking and renege probabilities for these problems are tabulated in Table 2.

For the same set of problems, the results obtained using the Monte Carlo summation method are presented next. Table 3 tabulates the results for the problems with $\lambda^{\mathbf{1}}$ and Table 4 tabulates the results for the problems with $\lambda^{\mathbf{2}}$. Both blocking and renege probabilities are reported, since both of these are determined approximately for this method. Each point

Table 1: Parameters for the Numerical Examples

Problem	α	μ
LL_1	(0.25, 0.50,0.75)	(2.5,2.8,3.5)
LL_2	(1.5, 1.7,2.0)	(2.5,2.8,3.5)
LH	(1.5, 1.7,2.0)	(0.5,2.8,13.5)
HL	(0.25, 2.50,8.75)	(2.5,2.8,3.5)
HH_1	(0.25, 2.50,8.75)	(0.5,2.8,13.5)
HH_2	(0.25, 2.50,8.75)	(13.5,2.8,0.5)

estimate is followed by its 95% confidence interval. All of the estimates have been obtained by performing 500,000 iterations of the Monte Carlo summation method, using importance sampling parameters determined by trial and error. The importance sampling parameters that were used are tabulated in Table 5, where $\gamma^{\mathbf{i}}$ is the vector of importance sampling parameters for the problems with $\lambda^{\mathbf{i}}$, $i = 1, 2$. The estimates could be further improved by increasing the number of iterations or by selecting importance sampling parameters that are closer to the optimal ones.

Comparing the estimates in Tables 3 and 4 to the exact values in Table 2, it is clear that the Monte Carlo summation technique performs well in terms of accuracy. The performance of this method does not seem to depend on the homogeneity of parameters in a direct way. However, for the examples with $\lambda^{\mathbf{2}}$ (tabulated in Table 4) the method performs better than for the same problems with $\lambda^{\mathbf{1}}$. The importance sampling parameters in Table 5 demonstrate that while for most of the problems with λ_2 one obtains good estimates with $\gamma^{\mathbf{2}} = \lambda^{\mathbf{2}}$, the same is not true for the problems with $\lambda^{\mathbf{1}}$. Four out of six of these problems required setting

Table 2: Exact Renege and Blocking Probabilities

Problem	(R_1, R_2, R_3)	(B_1, B_2, B_3)
LL_1, $\boldsymbol{\lambda^1}$	(0.1352, 0.1701, 0.1527)	(0.0549, 0.0239, 0.0098)
LL_2, $\boldsymbol{\lambda^1}$	(0.2091, 0.1897, 0.1505)	(0.0036,0.0024,0.0013)
LH, $\boldsymbol{\lambda^1}$	(0.6913, 0.2107, 0.0139)	(0.0159,0.0027,0.0)
HL, $\boldsymbol{\lambda^1}$	(0.1078, 0.2293, 0.2153)	(0.0423,0.0012,0.0)
HH_1, $\boldsymbol{\lambda^1}$	(0.5823, 0.2912, 0.0394)	(0.2825,0.0015,0.0001)
HH_2, $\boldsymbol{\lambda^1}$	(0.0024, 0.2126, 0.6990)	(0.0002,0.0011,0.0002)
LL_1, $\boldsymbol{\lambda^2}$	(0.4588, 0.5939, 0.6271)	(0.8461, 0.7707, 0.6890)
LL_2, $\boldsymbol{\lambda^2}$	(0.8120, 0.8088, 0.7902)	(0.5589, 0.5156, 0.4513)
LH, $\boldsymbol{\lambda^2}$	(0.9566, 0.8058, 0.4283)	(0.6097, 0.5145, 0.2632)
HL, $\boldsymbol{\lambda^2}$	(0.4569, 0.8484, 0.8921)	(0.8455, 0.3954, 0.0697)
HH_1, $\boldsymbol{\lambda^2}$	(0.8070, 0.8426, 0.6159)	(0.9100, 0.3933, 0.0423)
HH_2, $\boldsymbol{\lambda^2}$	(0.1102, 0.8478, 0.9839)	(0.5054, 0.3952, 0.0799)

γ^1 to values other than λ^1. This suggests that for systems with higher load (characterized by higher arrival rates relative to renege and service rates), the proposed implementation generates reliable estimates, even in the absence of importance sampling. The robustness of this observation needs to be further tested. In general, the accuracy of the method depends on the choice of importance sampling parameters. As a result, determination of optimal or near-optimal importance sampling parameters should be the first step in future research endeavours.

6 Conclusion and Directions for Future Research

Recognizing the difficulty of implementing loss models with product form or near-product form solutions in the absence of computational methods that simplify their analysis, this paper has developed such methods for a specific type of loss system. By introducing the notion of multi-dimensional convolutions, one was able to extend the existing methodology for computing performance measures for the pure loss case to that of the queueing and reneging cases. Observing that for the reneging case, this combinatorial method may still be too expensive for some applications, the use of Monte Carlo summation to perform these computations was explored. An adaptation of the Monte Carlo summation technique to the multi-class, multi-resource, processor-shared system was proposed.

Depending on the industry they are in, phone centers may be very large in terms of the total number of service representatives and phone trunks. For very large problems, difficulties have been encountered in implementing the methods to compute the performance measures proposed in this paper. Numerical underflow and overflow problems that arise from having extremely large values for the normalization constant constitute the source for these difficulties. The factorial term in the product form solutions further magnifies this

Table 3: Renege and Blocking Probabilities and Associated Confidence Intervals for the Monte Carlo Summation Approach: Problems with $\lambda^{\mathbf{1}}$

Problem	R_1	R_2	R_3
	B_1	B_2	B_3
LL_1	0.1418 (0.1255,0.1581)	0.1467 (0.1232,0.1703)	0.1319 (0.1020,0.1619)
	0.052 (0.0366,0.0663)	0.0110 (0.0059,0.0165)	0.0060 (0.0030,0.0072)
LL_2	0.2098 (0.1937,0.2259)	0.1869 (0.1694,0.2044)	0.1429 (0.1288,0.1569)
	0.0041 (0.0022,0.0060)	0.0013 (0.0008,0.0018)	0.0011 (0.0007,0.0014)
$LH,$	0.6798 (0.6681,0.6915)	0.02373 (0.2274,0.2472)	0.0094 (0.0045, 0.0142)
	0.0162 (0.0150,0.0176)	0.0023 (0.0025,0.0041)	0.0001 (0.0001,0.0002)
HL	0.1100 (0.1009,0.1191)	0.2150 (0.1910,0.2390)	0.2041 (0.1671,0.2411)
	0.0446 (0.0438, 0.0544)	0.0005 (0.0003,0.0007)	0.0 (0.0,0.0001)
HH_1	0.5806 (0.5738,0.5873)	0.2872 (0.2745,0.3000)	0.0325 (0.0201, 0.0448)
	0.2809 (0.2762,0.2866)	0.0015 (0.0011,0.0019)	0.0 (0.0,0.0001)
HH_2	0.0015 (0.0008, 0.0021)	0.2401 (0.2278,0.2524)	0.7066 (0.6729,0.7403)
	0.0 (0.0,0.0)	0.0011 (0.0007,0.0014)	0.0002 (0.0001,0.0002)

Table 4: Renege and Blocking Probabilities and Associated Confidence Intervals for the Monte Carlo Summation Approach: Problems with $\lambda^{\mathbf{2}}$

Problem	R_1	R_2	R_3
	B_1	B_2	B_3
LL_1	0.4598 (0.4566, 0.4630)	0.5925 (0.5891, 0.5958)	0.6275 (0.6245, 0.6305)
	0.846 (0.8455,0.8475)	0.7710 (0.7710,0.7715)	0.6900 (0.6888,0.6904)
LL_2	0.8130 (0.8096,0.8163)	0.8078 (0.8047, 0.8110)	0.7906 (0.7877,0.7935)
	0.5593 (0.5379,0.5617)	0.5163 (0.5139,0.5168)	0.4515 (0.4501,0.4529)
$LH,$	0.9554 (0.9505,0.9603)	0.8053 (0.8017,0.8090)	0.4278 (0.4261, 0.4295)
	0.6094 (0.6087,0.6110)	0.5142 (0.5126,0.5169)	0.2632 (0.2618,0.2646)
HL	0.4577 (0.4541, 0.4614)	0.8479 (0.8447, 0.8512)	0.8917 (0.8884, 0.8950)
	0.8458 (0.8446,0.8460)	0.3952 (0.3936,0.3977)	0.0700 (0.0690,0.0710)
HH_1	0.8065 (0.7943,0.8188)	0.8450 (0.8402,0.8498)	0.6138 (0.6095,0.6180)
	0.9100 (0.9087,0.9124)	0.3943 (0.3920,0.3966)	0.0421 (0.0411, 0.0430)
HH_2	0.1082 (0.1077,0.1087)	0.8320 (0.8285,0.8354)	0.9863 (0.9827,0.9900)
	0.5054 (0.5037,0.5070)	0.3966 (0.3949,0.3982)	0.0080 (0.0794,0.0812)

Table 5: Importance Sampling Parameters for the Numerical Examples

Problem	$\gamma^{\mathbf{1}}$	$\gamma^{\mathbf{2}}$
LL_1	(1.0,1.0,1.0)	(10.0,10.0,10.0)
LL_2	(1.0,1.0,1.3)	(10.0,10.0,10.0)
LH	(0.98,1.0,1.0)	(11.0,10.0,12.0)
HL	(1.0,1.0,1.4)	(10.0,10.0,10.0)
HH_1	(1.0,1.0,1.3)	(10.0,10.0,10.0)
HH_2	(1.0,1.0,1.0)	(10.0,10.0,10.0)

problem of excessive growth in the normalization constant. An implementation such as the one suggested by Mitra and Ramakrishnan (1990) and Ross *et al.* (1993) has been used to overcome this problem.

In the proposed implementation of the Monte Carlo summation method, it was mentioned that γ_k acted as an importance sampling parameter which can be used to reduce the variance of the estimators. For the numerical examples in this paper as well as the analysis that was performed in Akşin and Harker (1996b), these parameters were obtained through trial and error. Determining optimal importance sampling parameters is an important issue for future research. Use of Monte Carlo summation coupled with optimal importance sampling would result in tighter confidence intervals for performance measure estimates.

Acknowledgements

This research was funded by a generous grant from the Alfred P. Sloan Foundation to the Wharton Financial Institutions Center. The comments of Keith Ross are warmly acknowledged.

References

[1] De Waal, P.R. and Van Dijk, N.M. "Monotonicity of performance measures in a processor sharing queue". *Performance Evaluation*, 12:5–16, 1991.

[2] Akşin O.Z. and Harker, P.T. "Modeling a phone center: Analysis of a multi-class, multi-resource, processor shared loss system". Working Paper, Financial Institutions Center, The Wharton School, 1996a.

[3] Akşin, O.Z. and Harker, P.T. "To sell or not to sell: Determining the tradeoffs between sales and service in retail banking phone centers". Working Paper, Financial Institutions Center, The Wharton School, 1996b.

[4] Akşin, O.Z. and Harker, P.T. "Staffing a call center". INSEAD Working Paper, 1996c.

[5] Kaufman, J.S. "Blocking in a shared resource environment". *IEEE Transactions on Communications*, COM-29:1474–1481, 1981.

[6] Bratley, P., Fox, B.L., and Shrage, L.E. *A Guide to Simulation* . Springer Verlag, 1987.

[7] Buzen, J.P. "Computational algorithms for closed queuing networks with exponential servers". *Communications of the ACM*, 16:527–531, 1973.

[8] Lam, S.S. and Lien, Y.L. “A tree convoluted algorithm for the solution of queueing networks”. *Communications of the ACM*, 26:203–215, 1983.

[9] McKenna, J., Mitra, D., and Ramakrishnan, K.G. “A class of closed Markovian queueing networks: Integral representations, asymptotic expansions, and generalizations”. *Bell Systems Technical Journal*, 60:599–641, 1981.

[10] Mitra, D. “Asymptotic analysis and computational methods for a class of simple circuit-switched networks with blocking”. *Advances in Applied Probability*, 19:219–239, 1987.

[11] Reiser, M. and Kobayashi, H. “Queueing networks with multiple closed chains: theory and computational algorithms”. *IBM Journal of Research and Development*, 19:283–294, 1975.

[12] Roberts, J.W. “A service system with heterogeneous user requirements”. *Performance of Data Communication Systems and their Applications*, pages 423–431, 1981.

[13] Ross, K.W. *Loss Models for Multiservice Telecommunication Networks.* Springer Verlag, 1995.

[14] Tsang, D.H.K., and Ross, K.W. “Algorithms to determine exact blocking probabilities for multirate tree networks”. *IEEE Transactions on Communications*, 38:1266–1271, 1990.

[15] Ross, K.W. and Wang, J. “Monte Carlo summation applied to product-form loss networks”. *Probability in the Engineering and Informational Sciences*,6:323 1992.

[16] Ross, K.W., Tsang, D.H.K., and Wang, J. “Monte Carlo summation and integration applied to multiclass queueing networks”. *Journal of the Association for Computing Machinery*, 41:1110–1135, 1993.

[17] Mitra, D. and Ramakrishnan, K.G. "A numerical investigation into the optimal design of congestion controls for high speed data networks". Technical report, AT & T Bell Laboratories, 1990.