Available online at www.sciencedirect.com
SCIENCE DIRECT®
Performance Evaluation 61 (2005) 41–64
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

Iterative algorithms for performance evaluation of closed network models

Zinovi L. Krougly[1], David A. Stanford*

Department of Statistical & Actuarial Sciences, University of Western Ontario, WSC 262, London, Ont., Canada N6A5B9

Received 6 January 2003; accepted 23 September 2004
Available online 8 December 2004

Abstract

A number of nonlinear programming algorithms are proposed to obtain the approximate solutions for non-product form multiclass queueing network models, as well as priority queueing networks. Using sensitivity analysis, we develop an efficient iterative technique for closed queueing networks. We compare the approximate solutions obtained from our approach with the global balance solution. Examples illustrate the accuracy of the approximation, and compare the efficiency of the different optimization methods we have implemented.
© 2004 Published by Elsevier B.V.

Keywords: Multiclass queueing networks; Sensitivity analysis; Priority approximation; Global balance solution; Optimization problems

1. Introduction

In the design and performance analysis of computer networks, closed queueing networks have played a key role [4,5,8,9,19,30,32]. Whereas product-form network models have become invaluable tools in this regard, a whole host of real networks do not satisfy the necessary conditions to make use of them. For such situations, various approximations have been proposed [1,5,6,8–10,30,32]. The present work presents a new approximation, with the main focus being networks employing a preemptive priority discipline at one or more service centers.

* Corresponding author. Tel.: +1 519 661 3612.
[1] Tel.: +1 519 661 2111x86985; fax: +1 519 661 3813.
E-mail addresses: zkrougly@stats.uwo.ca (Z.L. Krougly); stanford@stats.uwo.ca (D.A. Stanford).

0166-5316/$ – see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.peva.2004.09.004

The novel role of this paper is that it resorts to sensitivity analysis based on partial derivatives for various performance measures. This method has been previously used in [12,16,31,33] to obtain such derivative information as functions of the service demands and service rates. We present a unified nonlinear programming approach to arrive at an approximate solution. In fact, two main optimizing approaches are followed; one which employs the derivative information to develop efficient techniques to reach the optimal solution, and the other which does not.

Exact solutions for preemptive and nonpreemptive open queuing systems are given in many texts, such as [11,14,23,29]. These take the form of explicit expressions for the means, and transform solutions for the relevant distributions. Since then, related performance measures have been obtained, such as the interdeparture time distribution for each class of messages in a variety of systems featuring Poisson arrivals [26–28]. A common area of interest is multiclass feedback queues [15], where it has been shown that in certain circumstances, a priority arrangement of the various classes yields optimal performance [7,22]. However, few of these exact results carry over to open networks of priority queues. Similarly for closed networks, we have seen that exact queuing system solutions for BCMP networks based on Norton's theorem are available, but these do not apply for systems involving nonpreemptive or preemptive priorities.

A major development in the analysis of closed priority queueing networks is the "shadow server" approximation. The concept was first introduced in [2,21] to represent overhead in queueing models of operating systems (such as context switching or I/O). The shadow approximation for preemptive priority scheduling was first applied in [25].

The remainder of the paper is organized as follows. In Section 2, the pertinent sensitivity analysis background is presented. In Sections 3 and 4, we transform the closed queueing network problem to the relevant nonlinear programming model, and the necessary derivatives are obtained that are used in the objective functions. This enables us to use an efficient numerical technique, and to increase the convergence rate relative to methods not using derivatives. In Section 5, we illustrate the complexity of the global balance solution technique for a particular priority model. In Section 6, we present a diversity of examples in priority queuing networks, and compare the efficiency of the different numerical approaches. In Section 7, we compare the execution time of the nonlinear programming algorithms for the approximation models, and give some practical recommendations how to increase the convergence speed.

2. Sensitivity in closed queueing networks

Consider a closed product form queuing network with M service centers and R customer classes. The number of class v customers ($v = 1, \ldots, R$) is equal to n_v. The visit ratio e_{iv} is the solution to the system of linear equations $e_{iv} = \sum_{j=1}^{M} e_{jv} P_{jiv}; i = 0, \ldots, M; v = 1, \ldots, R$, where P_{jiv} are transition probabilities. The relative utilization of class v at center i is $x_{iv} = e_{iv}/\mu_{iv}$, where $1/\mu_{iv}$ is the mean service time for a class v customer at service center i. Let L_{iv}, U_{iv} and λ_{iv} be the mean queue length, the utilization and the throughput, respectively, at center i for class v customers. Let denote by $G(\bar{n})$ the normalization constant vector, comprising one constant for each class, after the last service center (center M) has been dealt with.

First consider single class networks with M centers and N customers. The partial derivatives for $\lambda_i(N)$, $U_i(N)$, $L_i(N)$, and $G(N)$ are given by the following equations [12,16,31,33]:

$$\frac{\partial G(N)}{\partial x_i} = \frac{G(N)}{x_i} L_i(N), \tag{2.1}$$

$$\frac{\partial L_i(N)}{\partial \mu_i} = -\frac{D_i(N)}{\mu_i}, \tag{2.2}$$

$$\frac{\partial L_i(N)}{\partial x_i} = \frac{D_i(N)}{x_i}, \tag{2.3}$$

$$\frac{\partial U_i(N)}{\partial x_j} = \frac{U_i(N)}{x_j}[L_j(N-1) - L_j(N)], \quad i \neq j, \tag{2.4}$$

$$\frac{\partial U_i(N)}{\partial x_i} = \frac{U_i(N)}{x_i}[1 + L_i(N-1) - L_i(N)], \tag{2.5}$$

$$\frac{\partial U_i(N)}{\partial \mu_i} = \frac{U_i(N)}{\mu_i}[1 + L_i(N-1) - L_i(N)], \tag{2.6}$$

$$\frac{\partial \lambda_i(N)}{\partial x_j} = \frac{\lambda_i(N)}{x_j}[L_j(N-1) - L_j(N)], \tag{2.7}$$

and

$$\frac{\partial \lambda_i(N)}{\partial \mu_i} = \frac{\lambda_i(N)}{\mu_i}[L_i(N) - L_i(N-1)], \tag{2.8}$$

where $D_i(N)$ is the variance of the number of customers at service center i, and $L_i(k)$ is the mean queue length at service center i when there are k customers in the network, $k = 1, \ldots, N$.

The mean response time is one of the most important and general performance measures for all computer-communication systems. From (2.2) and (2.8) we get:

$$\begin{aligned}\frac{\partial T}{\partial \mu_0} &= \frac{\partial (N - L_0(N))/\lambda_0}{\partial \mu_0} = \frac{1}{(\lambda_0)^2}\left[\lambda_0 \frac{\partial (N - L_0(N))}{\partial \mu_0} - (N - L_0(N))\frac{\partial \lambda_0}{\partial \mu_0}\right] \\ &= -\frac{1}{\lambda_0}\frac{\partial L_0(N)}{\partial \mu_0} - \frac{N - L_0(N)}{(\lambda_0)^2}\frac{\partial \lambda_0}{\partial \mu_0} = \frac{1}{\mu_0 \lambda_0}\{D_0(N) - (N - L_0(N))[L_0(N) - L_0(N-1)]\}\end{aligned} \tag{2.9}$$

The partial derivatives for multiple class networks are given by the following [12,33]:

$$\frac{\partial G(\bar{n})}{\partial x_{iv}} = \frac{G(\bar{n})}{x_{iv}} L_{iv}(\bar{n}) \tag{2.10}$$

$$\frac{\partial \lambda_r(\bar{n})}{\partial x_{iv}} = \frac{\lambda_r(\bar{n})}{x_{iv}}[L_{iv}(\bar{n} - 1_r) - L_{iv}(\bar{n})];\ i = 0, \ldots, M;\ v, r = 1, \ldots, R \tag{2.11}$$

where $\lambda_r = G(\bar{n} - 1)/G(\bar{n})$ is the throughput for class r customers, and $(\bar{n} - 1_r) = (n_1, \ldots, n_r - 1, \ldots, n_R)$ is the population vector with one class r customer less in the network.

In the next sections, we make use of these derivatives in order to develop an efficient iterative algorithm for queueing networks for which no exact solution exists.

3. Iterative technique for queuing network models

When the transition probabilities and service rates are allowed to depend on a system state, an exact closed form analytical solution does not exist.

The iterative procedure presented below is used for models with different classes of customers. The algorithm allows for two ways to specify the arrival process, and these are described below.

Although we employ a closed model, we can specify the input process using the interarrival time distribution, as in open models. This is asymptotically valid as the number of customers increases, as we show below. Furthermore, it provides a basis for comparison with open network models.

Assume that the input stream for each customer in class v is defined as a sequence of independent and identically distributed random variables with an exponential distribution function $A(t) = 1 - \exp(-\Lambda_{0v}t)$, where $1/\Lambda_{0v}$ is the common mean interarrival time for class v customers. Then the distribution of $k_v(t)$, the number of arrivals by time t, is given by

$$\begin{aligned}\Pr\{k_v(t) = k_v\} &= \binom{n_v}{k_v} [\Pr\{\tau_v \le t\}]^{k_v} [\Pr\{\tau_v > t\}]^{n_v - k_v} \\ &= \binom{n_v}{k_v} [1 - \exp(-\Lambda_{0v}t)]^{k_v} [\exp(-\Lambda_{0v}t)]^{n_v - k_v} \\ &= \binom{n_v}{k_v} [1 - \exp(-\Lambda_{0v}t)]^{k_v} \exp[-(n_v - k_v)\Lambda_{0v}t]. \end{aligned} \tag{3.1}$$

If one lets $n_v \to \infty$ and $\Lambda_{0v} \to 0$ so that $n_v\Lambda_{0v} \to \Lambda'_{0v}$, then the arrival process is approximately Poisson distributed with rate Λ'_{0v}. Thus, a finite population model may be approximated by an infinite one as the population size increases. Hence, we specify the arrival process in the traditional way, by using the distribution of the time τ'_v that a customer stays in the source after receiving service in the network, rather than using (3.1). This is performed in step 1 of the algorithm given below.

Let us specify the input process through the interarrival time distribution with the given arrival rates Λ_{0v} for class v customers.

Algorithm 3.1.

Step 0. Initialization. Set initial value of service rates in the source and the mean size of the source

$$\mu_{0v}^{(0)} = \Lambda_{0v}; \quad L_{0v}^{(0)} = n_v;\ v = 1, \ldots, R, \tag{3.2}$$

where $1/\mu_{0v}$ is the mean time that customer class v stays in the source after receiving service in the system.

Step 1. For steps $s = 0, 1, \ldots$, the iterates $\mu_{0v}^{(s)}$ and $L_{0v}^{(s)}$ are used to find the transition probabilities P_{ijv} and service rates μ_{iv} $(i, j = 0, \ldots, M; v = 1, \ldots, R)$.

Step 2. The calculation of the queuing network model is performed.

Step 3. The estimated solution is evaluated using an iterative algorithm:

$$\begin{cases} L_{0v}^{(s+1)} = \varphi_1(L_{0v}^{(s)}, \mu_{0v}^{(s)}) \\ \mu_{0v}^{(s+1)} = \varphi_2(L_{0v}^{(s+1)}, \mu_{0v}^{(s)}), \end{cases} \tag{3.3}$$

where $\mu_{0v}^{(s+1)}$ is given by either iterative formula (3.4) or (3.5) below:

$$\mu_{0v}^{(s+1)} = L_{0v}^{(s+1)}/[U_{0v}(1/\Lambda_{0v} - T_v)] \tag{3.4}$$

$$\mu_{0v}^{(s+1)} = N_v \Lambda_{0v}/U_{0v}, \quad v = 1, \ldots, R. \tag{3.5}$$

Define U_{iv} as the utilization at service center i for class v customers and T_v as the mean response time for class v customers. Since U_{0v} is a function of $L_{0v}^{(s+1)}$ and $\mu_{0v}^{(s)}$, we can write (where $\gamma_1(\cdot,\cdot)$ is some unspecified function)

$$U_{0v} = \gamma_1(L_{0v}^{(s+1)}, \mu_{0v}^{(s)}), \quad \text{and} \tag{3.6}$$

$$T_v = \sum_{i=1}^{M} \left(\frac{\mu_{iv}U_{iv}}{\mu_{0v}U_{0v}}\right)\frac{L_{iv}}{\mu_{iv}U_{iv}} = \frac{n_v - L_{0v}^{(s+1)}}{\lambda_{0v}} = \frac{n_v - L_{0v}^{(s+1)}}{\mu_{0v}^{(s)}U_{0v}}, \quad v = 1, \ldots, R, \tag{3.7}$$

where L_{iv} and λ_{0v} are accordingly mean queue length at center i for class v customers and throughput for class v customers.

The iterative formula (3.5) is based on Little's law and also follows from

$$\frac{1}{\Lambda_{0v}} = \sum_{i=0}^{M} \frac{e_{iv}}{e_{0v}}\frac{L_{iv}}{\lambda_{iv}} = \sum_{i=0}^{M} \frac{\lambda_{iv}}{\lambda_{0v}}\frac{L_{iv}}{\lambda_{iv}} = \frac{1}{\lambda_{0v}}\sum_{i=0}^{M} L_{iv} = \frac{N}{\lambda_{0v}} = \frac{N}{\mu_{0v}U_{0v}}, \tag{3.8}$$

where λ_{iv} is the throughput at service center i for class v customers.

Step 4. Convergence test: one assesses whether

$$\begin{cases} |L_{0v}^{(s+1)} - L_{0v}^{(s)}| < \varepsilon \\ |\mu_{0v}^{(s+1)} - \mu_{0v}^{(s)}| < \varepsilon \end{cases} \tag{3.9}$$

If so, the iteration stops. Otherwise, one returns to Step 1 to perform the next iteration.

If the input stream is given conventionally through μ_{0v} the simpler algorithm based on the system of nonlinear equations applies:

$$L_{0v}^{(s+1)} = \varphi_1(L_{0v}^{(s)}), \quad v = 1, \ldots, R. \tag{3.10}$$

The convergence proof for iterative Algorithm 3.1 is given in the Appendix A. Notice that the iterative formula (3.4) is superior to (3.5) and provides better algorithmic convergence because partial derivatives $\partial\mu_{0v}^{(s+1)}/\partial\mu_{0v}^{(s)}$ calculated by (3.5) are larger than by (3.4).

4. Numerical methods for priority approximation

We incorporate below a number of algorithms in the priority context. All of these employ a shadow server approximation, to reflect the utilization U_{iv} of the higher priority classes. In one approach, we resort to an iterative scheme. Another possibility is to introduce an objective function and use a direct-search procedure. Yet another option is to solve this optimization problem with the assistance of derivative information, and this is described extensively below.

The approximate shadow service rate μ'_{iv} of a class v customer at its dedicated shadow center is found from the utilization U_{iv} of the higher priority classes.

Algorithm 4.1. Iterative scheme

Step 0. Transform the original model into the shadow model. Initialize: $U_{iv}^{(0)} = 0$, $v = 1, \ldots, R-1$.

Step 1. Compute the shadow service rates

$$\mu'_{iv} = \mu_{iv}\left(1 - \sum_{k=1}^{v-1} U_{ik}^{(s)}\right), \quad s = 0, 1, \ldots, \tag{4.1}$$

where μ_{iv} denotes the actual service rate of a class v at the priority center i.

Step 2. Find the product form solution for a BCMP network with $M + R - 1$ service centers. Compute $U_{ik}^{(s+1)}$, $v = 1, \ldots, R-1$.

Step 3. If the utilizations $U_{ik}^{(s+1)}$ have not converged, return to Step 1. Otherwise, stop.

Using an m-dimensional vector-valued function $F(\bar{c})$, where $m = R - 1$, the nonlinear programming problem for priority approximation can be formally stated as

Problem 4.1. Nonlinear programming scheme:

$$\min F(\bar{c}) = \sum_{k=1}^{m} f_k^2(\bar{c}) \tag{4.2}$$

subject to

$$g_i(\bar{c}) > 0; \quad i = 1, \ldots, 2m \tag{4.3}$$

where

$$f_k(\bar{c}) = \varphi_k(\bar{c}) - c_k; \tag{4.4}$$

$$g_i(\bar{c}) = \begin{cases} c_i;\ i = 1, \ldots, m; \\ 1 - c_i;\ i = m+1, \ldots, 2m; \end{cases} \tag{4.5}$$

for $\bar{c} = (c_1, \ldots, c_m) = m$-component solution vector, and $\varphi_k(\bar{c}) = U_{ik}^{(s+1)}$ is the utilization at the shadow priority center k, $k = 1, \ldots, m$.

The nonlinear approach can be used either with or without derivative information. We consider next the case where we are able to calculate, at a given $m = R - 1$ dimensional point $\bar{c}$, not only the value of a function $f(\bar{c})$ but also the gradient vector of first partial derivatives.

Both the conjugate gradient minimization and quasi-Newton minimization methods were used for priority network implementations [17].

First assume that closed queueing network has two classes of customers. In what follows, it is assumed throughout that class 1 has preemptive priority over class 2 at the priority center i. Because

$$\frac{\partial x'_{i2}}{\partial \mu'_{i2}} = \frac{\partial (e'_{i2}/\mu'_{i2})}{\partial \mu'_{i2}} = -\frac{e'_{i2}}{(\mu'_{i2})^2} = -\frac{x'_{i2}}{\mu'_{i2}}, \tag{4.6}$$

using (2.11) we get

$$\frac{\partial U_{i1}^{(s+1)}}{\partial \mu'_{i2}} = \frac{1}{\mu_{i1}} \frac{\partial \lambda_{i1}}{\partial \mu'_{i2}} = \frac{1}{\mu_{i1}} \frac{\partial \lambda_{i1}}{\partial x'_{i2}} \frac{\partial x'_{i2}}{\partial \mu'_{i2}} = \frac{1}{\mu_{i1}} \frac{\partial (e_{i1}\lambda_{01}/e_{01})}{\partial x'_{i2}} \frac{\partial x'_{i2}}{\partial \mu'_{i2}}$$
$$= \frac{1}{\mu_{i1}} \frac{e_{i1}}{e_{01}} \frac{\lambda_{01}}{x'_{i2}} [L_{i2}(n_1 - 1, n_2) - L_{i2}(n_1, n_2)] \left(-\frac{x'_{i2}}{\mu'_{i2}} \right)$$
$$= \frac{U_{i1}^{(s+1)}}{\mu'_{i2}} [L_{i2}(n_1, n_2) - L_{i2}(n_1 - 1, n_2)]. \tag{4.7}$$

Furthermore, since

$$\frac{\partial \mu'_{i2}}{\partial U_{i1}^{(s)}} = \frac{\partial [\mu_{i2}(1 - U_{i1}^{(s)})]}{\partial U_{i1}^{(s)}} = -\mu_{i2} = -\frac{\mu'_{i2}}{1 - U_{i1}^{(s)}}, \tag{4.8}$$

the derivatives on this iteration are

$$\frac{\partial U_{i1}^{(s+1)}}{\partial U_{i1}^{(s)}} = \frac{\partial U_{i1}^{(s+1)}}{\partial \mu'_{i2}} \frac{\partial \mu'_{i2}}{\partial U_{i1}^{(s)}} = \frac{U_{i1}^{(s+1)}}{1 - U_{i1}^{(s)}} [L_{i2}(n_1 - 1, n_2) - L_{i2}(n_1, n_2)]. \tag{4.9}$$

The derivative for objective function (4.2) is

$$\frac{\partial F(\bar{c})}{\partial U_{i1}^{(s)}} = 2(U_{i1}^{(s+1)} - U_{i1}^{(s)}) \left\{ \frac{U_{i1}^{(s+1)}}{1 - U_{i1}^{(s)}} [L_{i2}(n_1 - 1, n_2) - L_{i2}(n_1, n_2)] - 1 \right\}. \tag{4.10}$$

Now assume that the customers belong to $R \geq 2$ different priority classes, indexed by the subscript v, $v = 1, \ldots, R$. Using (2.11) and (4.1) one obtains

$$\frac{\partial \mu'_{ir}}{\partial U_{iv}^{(s)}} = \frac{\partial \left[\mu_{ir} \left(1 - \sum_{k=1}^{v} U_{ik}^{(s)} \right) \right]}{\partial U_{iv}^{(s)}} = -\mu_{ir} = \frac{-\mu'_{ir}}{1 - \sum_{k=1}^{v} U_{ik}^{(s)}};$$
$$v = 1, \ldots, R-1; \; r = 2, \ldots, R; \; r > v, \tag{4.11}$$

and the derivatives are given by

$$\frac{\partial U_{iv}^{(s+1)}}{\partial U_{iv}^{(s)}} = \frac{\partial U_{iv}^{(s+1)}}{\partial \mu'_{i(v+1)}} \frac{\partial \mu'_{i(v+1)}}{\partial U_{iv}^{(s)}} = \frac{U_{iv}^{(s+1)}}{\mu'_{i(v+1)}} [L_{i(v+1)}(\bar{n}) - L_{i(v+1)}(\bar{n} - 1_v)] \left[-\frac{\mu'_{i(v+1)}}{1 - \sum_{k=1}^{v} U_{ik}^{(s)}} \right]$$
$$= \left[\frac{U_{iv}^{(s+1)}}{1 - \sum_{k=1}^{v} U_{ik}^{(s)}} \right] [L_{i(v+1)}(\bar{n} - 1_v) - L_{i(v+1)}(\bar{n})]; \quad v = 1, \ldots, R-1. \tag{4.12}$$

For $v > r$ we get the same solution as for $v = r$:

$$\frac{\partial U_{iv}^{(s+1)}}{\partial U_{ir}^{(s)}} = \frac{\partial U_{iv}^{(s+1)}}{\partial \mu'_{i(v+1)}} \frac{\partial \mu'_{i(v+1)}}{\partial U_{ir}^{(s)}} = \frac{U_{iv}^{(s+1)}}{\mu'_{i(v+1)}} [L_{i(v+1)}(\bar{n}) - L_{i(v+1)}(\bar{n} - 1_v)] \left[-\frac{\mu'_{i(v+1)}}{1 - \sum_{k=1}^{v} U_{ik}^{(s)}} \right]$$
$$= \left[\frac{U_{iv}^{(s+1)}}{1 - \sum_{k=1}^{v} U_{ik}^{(s)}} \right] [L_{i(v+1)}(\bar{n} - 1_v) - L_{i(v+1)}(\bar{n})]; \quad v > r \tag{4.13}$$

and

$$\frac{\partial U_{iv}^{(s+1)}}{\partial U_{ir}^{(s)}} = 0; \quad v < r. \tag{4.14}$$

Using (4.12)–(4.14) the first partial derivatives for objective function (4.2) are

$$\frac{\partial F(\bar{c})}{\partial U_v} = 2(U_{iv}^{(s)} - U_{iv}^{(s+1)}) + 2\sum_{k=v}^{R-1}\left\{(U_{ik}^{(s+1)} - U_{ik}^{(s)})\left[\frac{U_{ik}^{(s+1)}}{1-\sum_{l=1}^{k} U_{il}^{(s)}}\right]\right.$$

$$\left.\times\,[L_{i(k+1)}(\bar{n}-1_k) - L_{i(k+1)}(\bar{n})]\right\}; \quad v = 1,\ldots,R. \tag{4.15}$$

We also can iterate using as the unknowns the service rates $\mu'_{iv}\,(v = 1,\ldots,R-1)$ instead of the utilizations U_{iv}.

At this point, we turn our attention to the arrival process, in terms of the interarrival-time parameters introduced in Section 3. The following m-dimensional vector-valued function is used to find the numerical solution.

Problem 4.2.

$$\min F(\bar{c}) = \sum_{k=1}^{m} f_k^2(\bar{c}) \tag{4.16}$$

subject to

$$g_i(\bar{c}) > 0; \quad i = 1,\ldots,m+R \tag{4.17}$$

where

$$f_k(\bar{c}) = \varphi_k(\bar{c}) - c_k; \tag{4.18}$$

$\bar{c} = (c_1,\ldots,c_m)$ is m-dimensional solution vector; $m = 2R-1$; $\varphi_k(\bar{c}) = U_{ik}^{(s+1)}$ is the utilization at shadow priority center $k, k = 1,\ldots,R-1$; $\varphi_{R-1+k}(\bar{c}) = \mu_{0k}$ is service rate in the source for every priority class $k, k = 1,\ldots,R$; $g_i(\bar{c})$ are taking into account $2(R-1)+R = (m+R-1)$ constraints:

$$\begin{aligned} &c_i > 0, \quad i = 1,\ldots,R-1; \qquad c_i < 1, \quad i = R,\ldots,2(R-1);\\ &c_i > 0, \quad i = 2R-1,\ldots,m+R-1. \end{aligned} \tag{4.19}$$

Service rates in the source are determined by (3.4) or (3.5).

The system of nonlinear equations in $2R-1$ unknowns $c_1 = U_{i1}, \ldots, c_{R-1} = U_{i(R-1)}, c_R = \mu_{01}, \ldots, c_{2(R-1)} = \mu_{0(R-1)}$ can be described as following

$$\begin{cases} U_{i1}^{(k+1)} = f_1(U_{i1}^{(k)}, \ldots, U_{i(R-1)}^{(k)}, \mu_{01}^{(k)}, \ldots, \mu_{0R}^{(k)}) \\ \vdots \\ U_{i(R-1)}^{(k+1)} = f_{R-1}(U_{i1}^{(k)}, \ldots, U_{i(R-1)}^{(k)}, \mu_{01}^{(k)}, \ldots, \mu_{0R}^{(k)}) \\ \mu_{01}^{(k+1)} = f_R(U_{i1}^{(k)}, \ldots, U_{i(R-1)}^{(k)}, \mu_{01}^{(k)}, \ldots, \mu_{0R}^{(k)}) \\ \vdots \\ \mu_{0R}^{(k+1)} = f_{2(R-1)}(U_{i1}^{(k)}, \ldots, U_{i(R-1)}^{(k)}, \mu_{01}^{(k)}, \ldots, \mu_{0R}^{(k)}) \end{cases} \tag{4.20}$$

We can transform our constrained nonlinear problem into an equivalent unconstrained problem as follows:

$$F(\bar{c}) = f(\bar{c}) + H(\bar{c}) \tag{4.21}$$

where $H(\bar{c})$ is a "penalty" function defined by

$$H(\bar{c}) = \sum_{i=1}^{m} \delta_i g_i^2(\bar{c}). \tag{4.22}$$

In the foregoing, δ_i is zero for each constraint that is satisfied, and unity otherwise. Thus, the penalty function defined by (4.22) vanishes inside the feasible region.

The following iterative procedure is used to find the approximating solution for the system of $2R-1$ nonlinear equations with $2R-1$ unknowns: $\mu'_{iv}; v = 1, \ldots, R-1$ and $\mu_{0v}; v = 1, \ldots, R$.

We limit the analysis below to the iterative case. The nonlinear programming technique to find the solution (4.21) is provided by constrained nonlinear optimization methods [20].

Algorithm 4.2.

Step 0. Transform the original model into the shadow model.

Step 1. Initialize:

$$U_{iv}^{(0)} = 0; \quad v = 1, \ldots, R-1 \tag{4.23}$$

$$\mu_{0v}^{(0)} = \Lambda_{0v}; L_{0v}^{(0)} = n_v; \quad v = 1, \ldots, R. \tag{4.24}$$

Step 2. The parameters $U_{iv}^{(s)}$, $\mu_{0v}^{(s)}$ and $L_{0v}^{(s)}$ are determined and used for calculation of transition probabilities P_{ijv} and service rates $\mu_{iv}(i, j = 0, \ldots, M; v = 1, \ldots, R)$.

Step 3. Compute the shadow service rates

$$\mu'_{iv} = \mu_{iv}\left(1 - \sum_{k=1}^{v-1} U_{ik}^{(s)}\right) \tag{4.25}$$

Step 4. Find product form solution for BCMP network with $(M + R - 1)$ centers. Compute $U_{iv}^{(s+1)}$, $v = 1, \ldots, R-1$ and the estimated solution $\mu_{0v}^{(s+1)}$, $L_{0v}^{(s+1)}$, $v = 1, \ldots, R$ by (3.3).

Step 5. Convergence test. Assess whether

$$\begin{cases} \max |U_{iv}^{(s+1)} - U_{iv}^{(s)}| < \varepsilon, v = 1, \ldots, R-1 \\ \max |L_{0v}^{(s+1)} - L_{0v}^{(s)}| < \varepsilon, v = 1, \ldots, R \\ \max |\mu_{0v}^{(s+1)} - \mu_{0v}^{(s)}| < \varepsilon, v = 1, \ldots, R \end{cases} \tag{4.26}$$

If yes, stop. Otherwise, return to Step 2 and perform the next iteration.

5. Global balance solution for priority models

Consider a closed tandem queueing network with two classes of customers, and $N_1 = N_2 = 4$ customers per class. The service discipline at server 1 is first come first served (FCFS). Class 1 has preemptive priority over class 2 at server 2. The service times at each server are exponentially distributed with rates μ_{ir} $(i = 1, 2; r = 1, 2)$, where $\mu_{11} = 1$, $\mu_{12} = 3$, $\mu_{21} = 1/3$, and $\mu_{22} = 1$. The notation $(n_1, k_1; n_2, k_2)$ says that there are n_i and k_i class i customers at servers 1 and 2, respectively, where $n_i + k_i = N_i$, $i = 1, 2$. Let $\pi(n_1, k_1; n_2, k_2)$ denote the probability for that state in equilibrium. The state transition diagram is shown in Fig. 1, where we have set $\lambda_1 = \mu_{11}$; $\lambda_2 = \mu_{12}$; $\mu_1 = \mu_{21}$; $\mu_2 = \mu_{22}$.

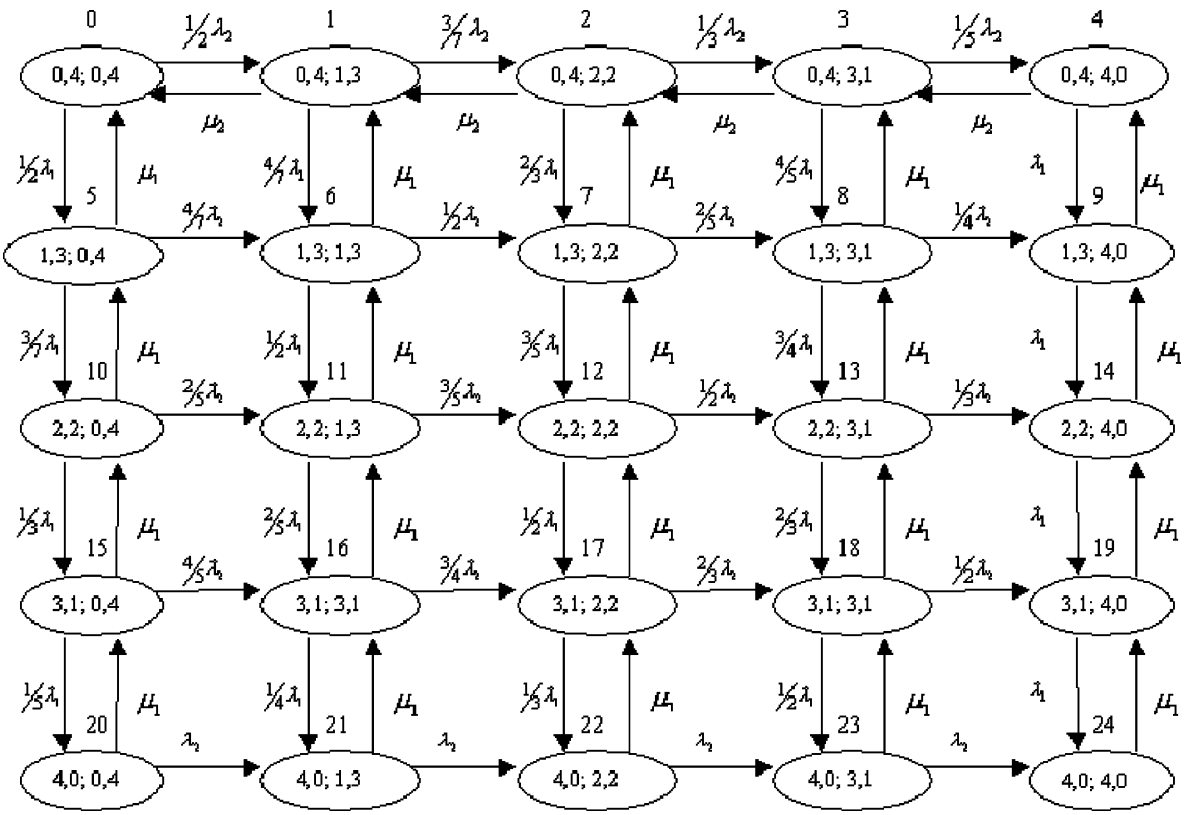

Fig. 1. State transition diagram for preemptive priority model.

Solving the global balance equations by setting the overall flux into a state equal to the overall flux out, we obtain the steady state probability vector, $\pi = (\pi_1, \ldots, \pi_{25})$, for lexicographically ordered states ((0, 4; 0, 4), (0, 4; 1, 3), ..., (0, 4; 4, 0), (1, 3; 0, 4), ..., (4, 0; 4, 0)). For the given rates, π is found to be as follows:

$$\begin{aligned}\pi = (&0.0001016150, 0.0002631570, 0.0007524080, 0.002166980, 0.00613636,\\ &4.16882 \times 10^{-5}, 0.0001477980, 0.0005043130, 0.001779140, 0.02324530,\\ &1.43303 \times 10^{-5}, 6.86008 \times 10^{-5}, 0.0002907380, 0.001310630, 0.07323780,\\ &3.72216 \times 10^{-6}, 2.40454 \times 10^{-5}, 0.0001317080, 0.000803349, 0.22190500,\\ &5.58324 \times 10^{-7}, 4.92725 \times 10^{-6}, 3.66224 \times 10^{-5}, 0.000328723, 0.66670100).\end{aligned}$$

The related performance measures are the marginal distributions, the server utilizations, and the throughputs.

The marginal probabilities $\pi_i(0)$ that the server i ($i = 1, 2$) is idle, are

$$\begin{aligned}\pi_1(0) &= \pi(0, 4; 0, 4) + \pi(0, 4; 1, 3) + \pi(0, 4; 2, 2) + \pi(0, 4; 3, 1) + \pi(0, 4; 4, 0)\\ &= 1 - 0.000101615 + 0.000263157 + 0.000752408\\ &\quad + 0.00216698 + 0.00613636 = 0.009421; \qquad (5.1)\\ \pi_2(0) &= 1 - (\pi(0, 4; 1, 3) + \pi(0, 4; 2, 2) + \pi(0, 4; 3, 1) + \pi(0, 4; 4, 0))\\ &= 1 - 0.000263157 + 0.000752408 + 0.00216698 + 0.00613636 = 0.0093189;\end{aligned}$$

The utilizations at servers 1 and 2 are

$$\begin{aligned}U_1 &= 1 - \pi_1(0) = 1 - 0.009421 = 0.990579;\\ U_2 &= 1 - \pi_2(0) = 1 - 0.990688 = 0.0093189.\end{aligned}$$

The throughputs at servers 1 and 2 are

$$\begin{aligned}\lambda_1 &= \mu_1 U_1 = (1/3)0.990579 = 0.330193;\\ \lambda_2 &= \mu_2 U_2 = (1/3)0.0093189 = 0.003106.\end{aligned}$$

A global balance solution technique is very computationally intensive for most networks due to the huge number of equations to solve, as this simple example illustrates, so for large networks we look for priority approximations. In Section 6, we compare the priority approximation with exact solutions based on the global balance technique.

6. Numerical examples

In this section we present several examples of exact and approximation solutions for various queuing network models, for a variety of service disciplines in the service centers. We used our software package ZEDNET, a Windows application written in C++, to perform the analysis. The package includes a graphical interface for input and output, and a number of C++ projects to handle product form solutions, non product form solution for priority and other approximations in closed queueing networks, Markov chain

routines, and global balance solution. It also features a set of analytical algorithms that reflect results from queueing theory (BCMP, mean value analysis, etc.).

Several numerical optimization algorithms have been implemented to solve the minimization problem and find the approximate solutions: an iterative algorithm, a direct search Powell's minimization method with does not use derivatives, and two descent methods which require the derivative information (these are conjugate gradient method and quasi-Newton method [20]).

The performance measures we compute for these examples consist of the throughput, utilization, mean queue length, mean waiting time, and mean response time per class.

Example 6.1. Consider a central server model with $R = 2$ classes of customers under three scenarios: a multiclass system operating under preemptive resume priority (PR) at service center 1, a multiclass system without priority (denoted "product-form"), and an equivalent product form system with a single class of customers. Service centers 2, 3, ..., M are FCFS centers and service center 1 is a processor sharing (PS) center. This is a computer system example, first modelled via BCMP networks in [4], in which center 1 represents the CPU and the others represent the input/output devices. As such, the results are exact for the non-priority systems. Our purpose in presenting this example is to highlight the difference in performance under the priority approximation. We address the accuracy of the priority approximation procedure in Examples 6.3 through 6.5 below.

The parameters for the equivalent single-class system are calculated by first solving for the equilibrium state probabilities of the corresponding multiclass model. From these we can find r_v, the rate at which class v customers leave service center 1, $v = 1, \ldots, R$. The equivalent customers have parameters

$$\frac{1}{\mu_1} = \sum_{v=1}^{R} \left(\frac{r_v}{\sum_{v=1}^{R} r_v} \right) \frac{1}{\mu_{1v}} \tag{6.1}$$

$$P_{j1} = \sum_{v=1}^{R} \left(\frac{r_v}{\sum_{v=1}^{R} r_v} \right) P_{ijv}, \quad j = 2, \ldots, M. \tag{6.2}$$

Table 1 gives the utilization and mean response time for the three scenarios previously mentioned. We see that the introduction of PR priority service at center 1, when compared with PS scheduling (product form solution), results in a slight increase in the occupancy of high priority customers, and a slightly larger decrease in the low priority occupancy.

Example 6.2. In this example, we present a central server model with four classes of customers. Table 2 gives the utilization and mean response time for the multiclass product form model, the "equivalent" product form single class model and the priority model with preemptive resume scheduling at service center 1. The introduction of PR priority service at center 1, relative to PS scheduling (product form solution), results in significant improvement in mean response time of class 1 customers, a slight decrease for classes 2 and 3 customers, and a slight increase in the mean response time of the lowest priority customers. Also notable from the results in Tables 1 and 2 is the fact that the occupancies at each center for the single class "equivalent" model are all smaller than the corresponding multiclass model.

Example 6.3. Table 3 gives the mean queue length for a network with preemptive resume priority discipline. There are two classes of customers and two service centers, where centers 1 and 2 are modelled as infinite server (IS) and FCFS centers, respectively. The service rates are $\mu_{11} = 0.4$, $\mu_{12} = 1.0$, $\mu_{21} = 20$

Table 1
Utilization and mean response time for the central server model with two classes of customers (product form solution), one class of "equivalent" customers and preemptive resume priority at service center 1

Class	1	2	All	Equivalent
Utilization (PR)				
Center 1	0.245	0.502	0.747	
Center 2	0	0.335	0.335	
Center 3	0.771	0	0.771	
Center 4	0.514	0	0.514	
Center 5	0.330	0	0.330	
Utilization (product form)				
Center 1	0.236	0.508	0.744	0.665
Center 2	0	0.339	0.339	0.303
Center 3	0.743	0	0.743	0.664
Center 4	0.495	0	0.495	0.442
Center 5	0.319	0	0.318	0.284
Mean response time (PR)	1.362	1.991		
Mean response time (product form)	1.413	1.968		1.702
Number of customers in class	3	1	4	4
Transition probabilities				
P_{12v}	0	1		0.193
P_{13v}	0.35	0		0.282
P_{14v}	0.35	0		0.282
P_{15v}	0.30	0		0.242
Service rates				
Center 1	9.0	1.0		3.536
Center 2	1.5	1.5		1.5
Center 3	1.0	1.0		1.0
Center 4	1.5	1.5		1.5
Center 5	2.0	2.0		2.0

and $\mu_{22} = 10$. The customer populations for each class, n_1 and n_2, are varying. The exact results were taking from [14, p. 105]. Comparing the priority approximation results with the exact solution, one observes that the accuracy is generally very good, but that it degrades slightly as the number of customers in the network increases.

Example 6.4. Consider a closed tandem queuing network with two service centers and two classes of customers. There are four customers in each class. Center 1 operates under a PR discipline. The service time of customers at each center are exponentially distributed with service rates $\mu_{jv} = 1/s_{jv}$ ($v = 1, \ldots, R$; $j = 1, 2$).

The results are presented in Table 4. We compare the throughput for priority approximation models with exact results and product form solutions. Exact results were computed for every class using the global balance solution technique, presented in Section 5. We observe that while the throughput rates for the high priority class are quite accurate (with errors ranging from 1 to 5%), the low priority can be off as much as 30%.

Table 2
Utilization and mean response time for the central server model with four classes of customers (product form), one class of "equivalent" customers and preemptive resume priority at service center 1

Class	1	2	3	4	All	Equivalent
Utilization (PR)						
Center 1	0.131	0.073	0.494	0.090	0.788	
Center 2	0.315	0.232	0.039	0.072	0.659	
Center 3	0.084	0.186	0.063	0.058	0.391	
Center 4	0.140	0.155	0.105	0.290	0.691	
Utilization (product form)						
Center 1	0.115	0.068	0.451	0.104	0.738	0.677
Center 2	0.275	0.217	0.036	0.083	0.612	0.561
Center 3	0.073	0.174	0.058	0.067	0.372	0.341
Center 4	0.122	0.145	0.096	0.334	0.698	0.640
Mean response time (PR)	3.810	3.441	10.130	5.525		
Mean response time (product form)	4.359	3.680	11.079	4.796		5.455
Number of customers in class	1	1	1	1	4	4
Transition probabilities						
P_{12v}	0.6	0.4	0.2	0.2		0.383
P_{13v}	0.2	0.4	0.4	0.2		0.291
P_{14v}	0.2	0.2	0.4	0.6		0.327
Service rates						
Center 1	2.0	4.0	0.2	2.0		1.084
Center 2	0.5	0.5	0.5	0.5		0.5
Center 3	0.625	0.625	0.625	0.625		0.625
Center 4	0.375	0.375	0.375	0.375		0.375

Table 3
The mean queue lengths for the network with preemptive resume priority discipline

Model	n_2	1		2		3		4		5		10	
	n_1	L_{21}	L_{22}	L_{21}	L_{22}	L_{21}	L_{22}	L_{21}	L_{22}	L_{21}	L_{22}	L_{21}	L_{22}
Approximation	1	0.02	0.09	0.02	0.20	0.02	0.33	0.02	0.48	0.02	0.65	0.02	2.21
Exact	1	0.02	0.09	0.02	0.20	0.02	0.33	0.02	0.48	0.02	0.66	0.02	2.22
Approximation	2	0.04	0.09	0.04	0.20	0.04	0.33	0.04	0.49	0.04	0.67	0.04	2.28
Exact	2	0.04	0.10	0.04	0.21	0.04	0.34	0.04	0.49	0.04	0.68	0.04	2.30
Approximation	3	0.06	0.10	0.06	0.21	0.06	0.34	0.06	0.50	0.06	0.69	0.06	2.35
Exact	3	0.06	0.10	0.06	0.21	0.06	0.35	0.06	0.51	0.06	0.70	0.06	2.38
Approximation	4	0.08	0.10	0.08	0.21	0.08	0.35	0.08	0.51	0.08	0.70	0.08	2.43
Exact	4	0.08	0.10	0.08	0.22	0.08	0.36	0.08	0.52	0.08	0.72	0.08	2.46
Approximation	5	0.11	0.10	0.11	0.22	0.11	0.36	0.11	0.53	0.11	0.72	0.11	2.50
Exact	5	0.11	0.10	0.11	0.23	0.11	0.37	0.11	0.54	0.11	0.75	0.11	2.55
Approximation	10	0.24	0.11	0.24	0.24	0.24	0.40	0.24	0.59	0.24	0.83	0.24	2.95
Exact	10	0.22	0.12	0.24	0.26	0.24	0.44	0.24	0.64	0.24	0.89	0.24	3.04

Table 4
Comparison the throughput for priority approximation model with exact result and product form solution

Model number	Class	s_{1v}	s_{2v}	Throughput		
				Approximation	Exact	Product form
1	1	3	3	0.206625	0.215642	0.148148
	2	3	3	0.111717	0.080654	0.148148
2	1	3	1	0.330551	0.330193	0.166650
	2	3	1	0.002782	0.003106	0.166650
3	1	5	2.5	0.193190	0.191647	0.099804
	2	5	2.5	0.006810	0.007962	0.099804

Example 6.5. Consider a network that comprises a terminal system (center 0) and five service centers: the CPU (center 1) and four disks (centers 2 through 5). The parameter values are the following: $P_{011} = P_{121} = P_{231} = P_{341} = P_{451} = P_{501} = 1.0$; $P_{122} = P_{232} = P_{342} = P_{412} = 1.0$; $\mu_{01} = 0.1$, $\mu_{11} = 0.25$, $\mu_{21} = 0.5$, $\mu_{31} = 0.5$, $\mu_{41} = 0.5$, $\mu_{51} = 0.5$; $\mu_{12} = 0.025$, $\mu_{22} = 0.5$, $\mu_{32} = 0.125$, $\mu_{42} = 0.167$, $\mu_{52} = 0.125$ (all service rates are in sec^{-1}). There are six low priority customers, while the number of high priority customers is allowed to vary. We compare in Table 5 the mean response time for the priority approximation with product form solution and simulation. Simulation results were taking from [19, p. 260].

Example 6.6. Consider a computer system composed of a CPU (center 1) and two disks (centers 2 and 3) used to support an interactive system with 35 terminals (center 0), split into two classes for the workload: $n_1 = 20$ and $n_2 = 15$. Center 0 represents the think time at the terminals, the PS station represents the CPU, and two load independent stations represent the disks (FCFS).

Table 6 presents the mean response time for the interactive system. If both classes have the same priority, the mean response time for class 1 is 2.63 and 5.98 s for class 2. If class 1 has a higher CPU priority over class 2, we have a significant improvement in the mean response time of the class 1 (from 2.63 to 0.60 s) with a moderate increase in the mean response time of the class 2: 13.4% (from 5.98 to 6.78).

Example 6.7. A client server system (Fig. 2) includes k client workstations (center 0) that are connected by Ethernet network (center 1) to a database server. The database server consists of a CPU (center 2) and two disk devices (centers 3 and 4). The client workstations are modeled as IS centers, and submit SQL requests to a database server.

Table 5
Mean queue length at CPU and mean response time for class 1 in the network with preemptive resume priority discipline

Solution technique	n_1									
	1		5		10		15		20	
	L_{11}	T_1	L_{11}	T_1	L_{11}	T_1	L_{11}	T_1	L_{11}	T_1
Approximation	0.2	12.9	1.3	19.5	4.3	32.7	8.6	50.4	13.5	70.0
Simulation		12.0		19.1		32.0		50.4		70.0
Product form	0.6	34.8	3.2	46.5	7.0	63.1	11.3	81.0	15.8	99.7

Table 6
Mean response time for interactive system with 35 workloads and two classes of customers

Class	1	2
Mean response time (s)		
Product form	2.63	5.98
Preemptive resume	0.60	6.78
Transition probabilities		
P_{10v}	0.1667	0.1
P_{11v}	0.3333	0.2
P_{12v}	0.1667	0.5
P_{13v}	0.1666	0.2
Service rates		
Center 0	0.05	0.0667
Center 1	4.2	1.4
Center 2	12.5	20.0
Center 3	10.0	16.667
Number of terminals	20	15

The Ethernet network (operating under carrier sense multiple access with collision detection, or CSMA/CD) can be modeled as a load dependent (LD) center to represent the effect of network contention [13,18]. The rate at which the Ethernet delivers packets, given k stations trying to use the channel, is:

$$m_{\mathrm{p}}(k) = \frac{1}{(L_{\mathrm{p}}/B + SC(k))}, \tag{6.3}$$

where $C(k) = (1 - A(k))/A(k)$ denotes the average number of collisions per request, and $A(k) = (1 - 1/k)^{k-1}$ denotes the probability of a successful transmission.

The other parameters are specified as follows: the mean length in bytes per SQL request $L_{\mathrm{SQL}} = 1000$ bytes, the network bandwidth $B = 10$ Mbits/s, the slot duration $S = 51.2$ ms, the mean packet length $L_{\mathrm{p}} = 1518$ bits, the maximum length of the data field of a packet $L_{\mathrm{d}} = 1492$ bits, and the mean number of packets per SQL request $N_{\mathrm{SQL}} = 1 + [L_{\mathrm{SQL}}/L_{\mathrm{d}}] = 7$ packets.

Given k active clients and one database server in the system, there are $k + 1$ workstations in the network. But, as the workstation transmits only on request, there are no collisions if there is only one client active.

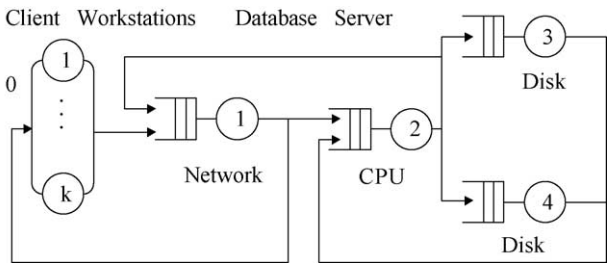

Fig. 2. Client server queuing network model.

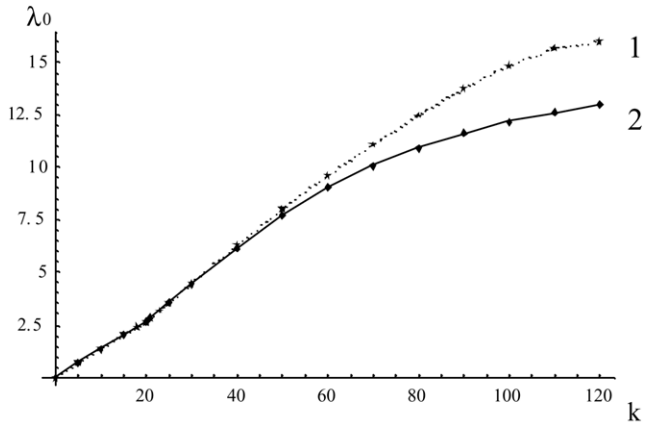

Fig. 3. Throughput as a function of the number of workstations. (1) Preemptive priority model; (2) Product form model.

Thus, considering the number of sent packets per SQL request, the service rate of the network, measured in SQL requests per second, is:

$$\mu_{\text{net}}(k) = \begin{cases} \dfrac{\mu_{\text{p}}(1)}{N_{\text{SQL}}}, k = 1 \\ \dfrac{\mu_{\text{p}}(k+1)}{N_{\text{SQL}}}, k > 1 \end{cases} \tag{6.4}$$

The client server model was evaluated using a two-class product form model, and a priority model where the first class has preemptive resume priority over the second class at CPU (center 2). The transition probabilities and service rates for classes 1 and 2 are as follows: $P_{011} = 1.0$; $P_{101} = 0.5$, $P_{121} = 0.5$, $P_{211} = 0.5$, $P_{231} = 0.2$, $P_{241} = 0.3$; $P_{321} = 1.0$; $P_{421} = 1.0$; $\mu_{01} = 0.2\,s^{-1}$, $\mu_{11} = \mu_{\text{net}}(k)$, $\mu_{21} = \mu_{\text{CPU}} = 32.2\,s^{-1}$, $\mu_{31} = 29.6\,s^{-1}$, $\mu_{41} = 15.5\,s^{-1}$; $P_{012} = 1.0$; $P_{102} = 0.5$, $P_{122} = 0.5$, $P_{212} = 0.15$, $P_{222} = 0.2$, $P_{232} = 0.4$, $P_{242} = 0.25$; $P_{322} = 1.0$; $P_{422} = 1.0$; $\mu_{02} = 0.15\,s^{-1}$, $\mu_{12} = \mu_{\text{net}}(k)$, $\mu_{22} = \mu_{\text{CPU}} = 25.4\,s^{-1}$, $\mu_{32} = 29.6\,s^{-1}$ and $\mu_{42} = 15.5\,s^{-1}$. The number of workstations in class 2 is $k_2 = 20$. The number of workstations in class 1 (denoted k_1) is varying.

The throughput $\lambda_0 = \lambda_{01} + \lambda_{02}$ as a function of the number of workstations $k = k_1 + k_2$ is shown in Fig. 3.

Example 6.8. Consider an open M/G/1 queuing system with preemptive resume priority. The system has two classes of customers. Arrival rates for classes 1 and 2 are $\Lambda_1 = 0.08$, $\Lambda_2 = 0.08$. Service rates for classes 1 and 2 are $\mu_1 = \mu_2 = 0.25$. The standard deviation of service times for classes 1 and 2 is $\sigma_1 = \sigma_2 = 1$.

Using Algorithm 4.2, we compare the priority approximation results in the closed model with exact results for the open model.

The associated closed tandem queuing network has two service centers and two classes of customers. Arrival rates are calculated using $\lambda_{01} = \Lambda_1/n_1$ and $\lambda_{02} = \Lambda_2/n_2$. Service rates at center 1 are $\mu_{11} = \mu_1$ and $\mu_{12} = \mu_2$.

The mean response time in the network with preemptive resume priority discipline for closed and open models is presented in Table 7.

Table 7
Mean response time in the network with preemptive resume priority discipline for closed and open models

Mean response time	$n_1\ n_2$	$n_1\ n_2$	$n_1\ n_2$	$n_1\ n_2$	$n_1\ n_2$	$n_1\ n_2$	$n_1\ n_2$	Open model
	2 2	10 10	20 20	40 40	50 50	60 60	80 80	
T_1	4.66	5.53	5.69	5.78	5.80	5.82	5.83	5.88
T_2	7.35	9.77	10.34	10.69	10.77	10.82	10.89	16.40

Table 8
Mean response times for the preemptive resume priority discipline compared with exact solution and open queueing network model

Model	n_2	5				10			
	n_1	T_1	T_2	Λ_1	Λ_2	T_1	T_2	Λ_1	Λ_2
Approximation	5	0.054	0.169	1.956	4.250	0.054	0.330	1.956	7.450
Exact	5	0.056	0.176	1.956	4.250	0.056	0.342	1.956	7.450
Open model		0.055	0.221	1.956	4.250	0.055	0.670	1.956	7.450
Approximation	10	0.060	0.196	3.904	4.110	0.060	0.406	3.904	6.960
Exact	10	0.060	0.216	3.904	4.110	0.060	0.440	3.904	6.960
Open model		0.062	0.285	3.904	4.110	0.060	1.130	3.904	6.960

Example 6.9. We compare the approximate solution given by Algorithm 4.2 with the exact result, and with the associated open M/G/1 model with preemptive resume discipline. There are two customers classes and two service centers, modelled as IS and PR centers, respectively. The arrival rates for closed model are given by $\lambda_{01} = \Lambda_1/n_1$, $\lambda_{02} = \Lambda_{01}/n_2$. The service rates are $\mu_{11} = 20$ and $\mu_{12} = 10$. The exact results were taking from [14, p. 106].

Table 8 gives mean response time for the preemptive resume priority discipline in comparison with exact solution and open queueing network model. A comparison of the results in Tables 7 and 8 demonstrate the benefits of using Algorithm 4.2 to run priority approximation, rather than ignoring the customer's population.

7. Comparison of optimization methods

Numerical examples were run to evaluate the performance of a variety of computer network models to assess the accuracy of the approximations, and to compare the convergence speed of the various nonlinear approximation techniques.

Below we estimate the execution time for priority approximation models using different numerical algorithms. Table 9 gives the relative CPU time of the numerical solution for the priority approximation models using different numerical algorithms: an iterative algorithm, quasi-Newton minimization, conjugate gradient minimization and Powell's method. The last of these is one of the best minimization procedures that does not resort to derivatives. The penalty function approach is used to transform a constrained nonlinear programming problem into an unconstrained problem by adding one or more functions of the constraints to the objective function.

Table 9
Relative CPU time of the numerical solution for priority approximation models

# Model	Iterative algorithm	Quasi-Newton	Conjugate gradient	Powell's method
1. Example 6.3 $M = 2\,(i = 1, 2)$, $R = 2$, $N_1 = 10$, $N_2 = 10$	1	1	1	2
2. Example 6.4 $M = 2\,(i = 1, 2)$, $R = 2$, $N_1 = 4$, $N_2 = 4$	0.4	0.3	0.6	0.6
3. Example 6.6 $M = 3\,(i = 0, 1, 2, 3)$, $R = 2$, $N_1 = 20$, $N_2 = 15$	3	3	3	6
4. Example 6.7 $M = 4\,(i = 0, 1, 2, 3, 4)$, $R = 2$, $N_1 = 30$, $N_2 = 30$	21	21	21	30
5. Example 6.7 $M = 4\,(i = 0, 1, 2, 3, 4)$, $R = 3$, $N_1 = 5$, $N_2 = 10$, $N_3 = 25$	69	81	117	255
6. Example 6.7 $M = 4\,(i = 0, 1, 2, 3, 4)$, $R = 4$, $N_1 = 5$, $N_2 = 5$, $N_3 = 5$, $N_4 = 5$	33	39	75	171
7. Example 6.7 $M = 4\,(i = 0, 1, 2, 3, 4)$, $R = 6$, $N_1 = 2$, $N_2 = 2$, $N_3 = 2$, $N_4 = 2$, $N_5 = 2$, $N_6 = 2$	24	27	50	156

All times are relative to the time to run model 1. All the test problems shown in Table 9 used the objective function (4.2) that was minimized with respect to the initial vector $\overline{c} = (c_1, \ldots, c_m)$, starting from the vector $\overline{c}^{(0)} = (c_1^{(0)}, \ldots, c_m^{(0)})$. All algorithms terminated when the objective function $F(\overline{c})$ fell below 10^{-13}.

The test results presented so far can give only a fragmentary picture of the relative effectiveness of the algorithms, because each study used a different unidimensional search method, different termination criteria and different methods of counting function evaluations. Table 9 demonstrates that numerical optimization methods using derivatives can yield a significant improvement for convergence speed. We observe that the quasi-Newton method and the iterative algorithm are generally superior at minimizing the $F(\overline{c})$ function. The conjugate gradient method appears to be nearly as satisfactory as the quasi-Newton method.

As expected, the search algorithms were slower than the algorithms that used derivatives, but what is notable is the high ranking of Powell's algorithm.

The quasi-Newton method works well if the starting vector is selected close enough to the minimum. We can use product form solution or perform several iterations to find a starting vector. Calculation by quasi-Newton and conjugate gradient methods are much more intensive than using an iterative algorithm, because at each iteration it is necessary to find the matrix of partial derivatives and to solve the system of linear equations. One solution to this computational problem is to calculate the matrix of partial derivatives only on the initial iteration, and use the results for all others iterations. But convergence in this case became linear, and the matrix of partial derivatives can be quite different from that at the final iteration.

Calculation (4.12) needs to be done using the mean queue length $L_{iv}(\overline{n} - 1_r)$ in closed network with one fewer class r customer. So as to avoid additional computing, one can employ the Bard [3] and Schweitzer

[24] approximation, which was proposed to approximate the mean value analysis algorithm:

$$L_{iv}(\bar{n} - 1_r) = \frac{(\bar{n} - 1_r)_v}{n_v} L_{iv}(\bar{n}), \tag{7.1}$$

where

$$(\bar{n} - 1_r)_v = \begin{cases} n_v, v \neq r \\ n_v - 1, v = r \end{cases} \tag{7.2}$$

Other simplifications have been proposed; see for example [5].

8. Conclusions

The performance evaluation algorithms presented in this paper use a nonlinear programming approach to obtain approximate solutions in queueing network models. A number of algorithms are proposed for determine the numerical results for priority approximation and other models. We introduced the minimization criteria and used a direct search procedure with efficient algorithms based on the calculation of derivative information to perform the optimization.

Acknowledgment

The authors thank Professor E. Gelenbe for useful suggestions which improved the present work.

Appendix A. Convergence proof for the Algorithm 3.1

Define the ordered pair $y = (y_1, y_2)$ by $y_1 = L_0$; $y_2 = 1/\mu_0$. Suppose that the iterates $y^{(s+1)}$ for $s = 0, 1, 2, \ldots$ are determined by (3.3):

$$L_0^{(s+1)} = \varphi_1(L_0^{(s)} \mu_0^{(s)}) \tag{A.1}$$

$$\mu_0^{(s+1)} = \varphi_1(L_0^{(s+1)}, \mu_0^{(s)}) = \frac{L_0^{(s+1)}}{U_0(1/\Lambda_0 - T)} \tag{A.2}$$

and set

$$\begin{cases} y_1^{(s+1)} = \varphi_1(y_1^{(s)}, y_2^{(s)}); \\ y_2^{(s+1)} = \varphi_2(y_1^{(s+1)}, y_2^{(s)}). \end{cases} \tag{A.3}$$

For $s = 0$ we obtain

$$y_1^{(1)} = \varphi_1\left(N, \frac{1}{\Lambda_0}\right), \tag{A.4}$$

$$y_2^{(1)} = \varphi_2\left(U_0, \frac{1}{\Lambda_0}\right). \tag{A.5}$$

Then the iterates $y_1^{(1)}$ and $y_2^{(1)}$, obtained on the first step, are less then the previous values:

$$y_1^{(1)} < y_1^{(0)} = N; \tag{A.6}$$

$$y_2^{(1)} < y_2^{(0)} = \frac{1}{\Lambda_0}. \tag{A.7}$$

Suppose $\mu_{B_0}(k)\,(k = 1, \ldots, N)$ is the throughput as a function of customers population k, determined when service center 0 is shorted. The statement (A.6) is obvious. By Norton's theorem for service center 0, there exists an equivalent reduced two-service center network consisting of center 0 and its "complementary center" B_0. The service rate at center B_0 is equal to the throughput $\mu_{B_0}(k)$, $k = 1, \ldots, N$ when center 0 is shorted and therefore, $\mu_{B_0}(k) > 0$, $k = 1, \ldots, N$.

The statement (A.7) is obtained from (A.2):

$$y_2^{(1)} = \frac{1/\Lambda_0 - T}{L_0^{(1)}/U_0} \leq \frac{1}{\Lambda_0} - T < \frac{1}{\Lambda_0} \tag{A.8}$$

because $L_0^{(1)} \leq U_0$ and $T > 0$.

When we use the iterative formula (3.5) $\mu_0 = N\Lambda_0/L_0$ and since $U_0 < N$

$$y_2^{(1)} < \frac{U_0}{N\Lambda_0} < \frac{1}{\Lambda_0}. \tag{A.9}$$

From (2.3) [31] and

$$x_i\mu_i = \alpha_i x_0 \mu_0 \tag{A.10}$$

we get

$$\frac{\partial L_0^{(s+1)}}{\partial L_0^{(s)}} = \frac{\partial\left[N - \sum_{i=1}^{M} L_i^{(s+1)}\right]}{\partial L_0^{(s)}} = -\sum_{i=1}^{M} \frac{\partial L_i^{(s+1)}}{\partial L_0^{(s)}} = -\sum_{i=1}^{M}\left(\frac{\partial L_i^{(s+1)}}{\partial x_i}\frac{\partial x_i}{\partial \alpha_i}\frac{\partial \alpha_i}{\partial L_0^{(s)}}\right)$$

$$= -\sum_{i=1}^{M}\frac{1}{x_i}D_i\frac{x_i}{\alpha_i}\frac{\partial\alpha_i}{\partial L_0} = -\sum_{i=1}^{M}\frac{D_i}{\alpha_i}\frac{\partial\alpha_i}{\partial L_0}. \tag{A.11}$$

Therefore if, $\partial\alpha_i/\partial L_0 < 0$; $i = 1, \ldots, M$, we see that

$$\frac{\partial\varphi_1(y_1, y_2)}{\partial y_1} > 0. \tag{A.12}$$

From (2.2) [31]

$$\frac{\partial L_i(N)}{\partial\mu_i} = -\frac{D_i(N)}{\mu_i} \tag{A.13}$$

it follows that

$$\frac{\partial\varphi_1(y_1, y_2)}{\partial y_2} = \frac{\partial\varphi_1(y_1, y_2)}{\partial\mu_0}\frac{\partial\mu_0}{\partial y_2} = -\frac{1}{\mu_0}D_0\left(-\frac{1}{y_2^2}\right) = \mu_0 D_0 = \frac{D_0}{y_2}. \tag{A.14}$$

Therefore,

$$\frac{\partial\varphi_1(y_1, y_2)}{\partial y_2} > 0. \tag{A.15}$$

Differentiating $\varphi_2(y_1, y_2)$ with respect to y_1,

$$\frac{\partial\varphi_2(y_1, y_2)}{\partial y_1} = \frac{\partial(1/\mu_0^{(s+1)})}{\partial L_0^{(s)}} = -\frac{1}{(\mu_0^{(s+1)})^2}\frac{\partial\mu_0^{(s+1)}}{\partial L_0^{(s)}}. \tag{A.16}$$

It is easy to prove that

$$\frac{\partial\mu_0^{(s+1)}}{\partial L_0^{(s)}} < 0, \tag{A.17}$$

and from (A.16)

$$\frac{\partial\varphi_2(y_1, y_2)}{\partial y_1} > 0. \tag{A.18}$$

Differentiating $\varphi_2(y_1, y_2)$ with respect to y_2,

$$\begin{aligned}\frac{\partial\varphi_2(y_1, y_2)}{\partial y_2} &= \frac{\partial(1/\mu_0^{(s+1)})}{\partial\mu_0^{(s)}} = -\frac{1}{(\mu_0^{(s+1)})^2}\frac{\partial\mu_0^{(s+1)}}{\partial y_2} = -\frac{1}{(\mu_0^{(s+1)})^2}\frac{\partial\mu_0^{(s+1)}}{\partial\mu_0^{(s)}}\frac{\partial\mu_0^{(s)}}{\partial y_2^{(s)}}\\ &= -\frac{1}{(\mu_0^{(s+1)})^2}\left(-\frac{1}{(y_2^{(s)})^2}\right)\frac{\partial\mu_0^{(s+1)}}{\partial\mu_0^{(s)}} = \left(\frac{\mu_0^{(s)}}{\mu_0^{(s+1)}}\right)^2\frac{\partial\mu_0^{(s+1)}}{\partial\mu_0^{(s)}}.\end{aligned} \tag{A.19}$$

Using (2.6)

$$\begin{aligned}\frac{\partial\mu_0^{(s+1)}}{\partial\mu_0^{(s)}} &= -\frac{N\Lambda_0}{U_0^2}\frac{\partial U_0}{\partial\mu_0^{(s)}} = -\frac{N\Lambda_0}{U_0^2}\left\{-\frac{1}{\mu_0^{(s)}}U_0[1 + L_0^{(s+1)}(N-1) - L_0^{(s+1)}(N)]\right\}\\ &= \frac{\mu_0^{(s+1)}}{\mu_0^{(s)}}[1 + L_0^{(s+1)}(N-1) - L_0^{(s+1)}(N)].\end{aligned} \tag{A.20}$$

It is easy to see that

$$[1 + L_0^{(s+1)}(N-1) - L_0^{(s+1)}(N)] > 0; \quad i = 1, \ldots, M, \tag{A.21}$$

so we get from (A.20) that

$$\frac{\partial\mu_0^{(s+1)}}{\partial\mu_0^{(s)}} > 0, \text{ and} \tag{A.22}$$

$$\frac{\partial\varphi_2(y_1, y_2)}{\partial y_2} > 0. \tag{A.23}$$

As a consequence of (A.12), (A.15),(A.18), and (A.23), all the partial derivatives of the vector $y(y_1 = \varphi_1(y_1, y_2), y_2 = \varphi_2(y_1, y_2))$ are positive.

But according to (A.6) and (A.7), $y^{(1)} < y^{(0)}$, so the sequence $\{y^{(s)}\}$ is monotonically decreasing:

$$y^{(s+1)} < y^{(s)}; \quad s = 0, 1, 2, \ldots. \tag{A.24}$$

The vector y is continuous and limited, so as a consequence $\{y^{(s)}\}$ converges to the optimum point $\bar{y}$, where $\bar{y}_1 = \varphi_1(\bar{y}_1, \bar{y}_2)$ and $\bar{y}_2 = \varphi_2(\bar{y}_1, \bar{y}_2)$.

References

[1] S. Agraval, Metamodeling: A Study of Approximation in Queueing Models, The MIT Press, 1985.
[2] M. Badel, E. Gelenbe, J. Leroudier, D. Potier, Adaptive optimization of time-sharing system, in: Proceedings of the IEEE (special issue on time-sharing system), 6 (1975) 958–965.
[3] Y. Bard, Some extentions to multiclass queueing network analysis, in: Proceedings of the Fourth International Symposium On Modelling and Performance Evaluation of Computer Systems, vol. 1, Vienna, New York, North-Holland, 1979, pp. 51–62.
[4] F. Baskett, K. Chandy, R. Muntz, F. Palacios, Open, closed and mixed networks of queues with different classes of customers, J. ACM 2 (1975) 248–260.
[5] G. Bolch, S. Greiner, H. DeMeer, K. Trivedi, Queuing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, John Wiley & Sons, 1998.
[6] R. Bryant, A. Krzesinski, M. Lakshmi, K. Chandy, The MVA priority approximation, ACM Transact. Comput. Syst. 4 (1984) 335–359.
[7] B. Dimitrov, D. Green, V. Rykov, On the influence of the observablility and controlability to the optimal control quality, in: Proceedings of the Fifth International Conference on Optimization, techniques and applications, vol. 2, ICOTA, Hong Kong, 2001, pp. 669–687.
[8] E. Gelenbe, I. Mitrani, Analysis and Synthesis of Computer Systems, Academic Press, London and New York, 1980.
[9] E. Gelenbe, G. Pujolle, Introduction to Queueing Networks, John Wiley & Sons, Chichester and New York, 1999.
[10] E. Gelenbe, On appoximate computer system models, J. ACM 22 (2) (1975) 261–269.
[11] B. Gnedenko, E. Danielyan, B. Dimitrov, G. Klimov, V. Matveev, Priority Queues, Moscow State University, Russian, Moscow, 1973.
[12] K. Gordon, L. Dowdy, The impact of certain parameter estimation errors in queueing network models, in: Proceeding Performance'80, Toronto, Canada printed as Performance Evaluation Review 2, 1980, pp. 3–9.
[13] G. Haring, J. Luthi, S. Majumdar, Mean value analysis for computer systems with variabilities in workload, in: Proceeding of the IEEE International Computer Performance and Dependability Symposium (IPDS'96), Urbana-Champaign, September 1996, pp. 32–41.
[14] N. Jaiswal, Priority Queues, Academic Press, 1968.
[15] M. Kitaev, V. Rykov, Controlled Queueing Systems, CRC Press, New York, 1995.
[16] Z. Krougly, M. Murshtein, Computational algorithms of optimization of closed queuing networks, Automat. Remote Control 7 (1990) 926–936.
[17] Z. Krougly, D. Stanford, Nonlinear Programming Algorithms for Performance Modelling of Computer Networks, Distributed Computer and Communication Networks: Stochastic Modelling and Optimization (DCCN-2003), Technosphera, Moscow, 2003, pp. 11–22.
[18] D. Menasce, V. Almeida, L.W. Dowdy, Capacity Planning and Performance Modeling: from Mainframes to Client-Server Systems, Prentice Hall, Upper Sandle River, NJ, 1994.
[19] E. Lazowska, J. Zahorjan, G. Graham, K. Sevcik, Quantitative System Performance: Computer System Analysis Using Queueing Network Models, Prentice-Hall, Englewood Cliffs, NJ, 1984.
[20] E. Polak, Optimization: Algorithms and Consistent Approximations, Springer, NY, 1997.
[21] D. Potier, E. Gelenbe, J. Lenfant, Adaptive allocation of CPU quanta, J. ACM 1 (1976) 97–102.
[22] V. Rykov, Monotone control of queueing systems with heterogeneous servers, Queue. Syst. 4 (2001) 391–403.
[23] K. Sapna Isotupa, D. Stanford, An infinite-phase quasi-birth-and-death model for the non-preemptive priority M/PH/1 queue, Stochastic Models 3 (2002) 387–424.

[24] P. Schweitzer, Approximate analysis of multiclass closed networks of queues, in: Proceeding of the International Conference on Stochastic Control and Optimization, Amsterdam, 1979, pp. 25–29.
[25] K. Sevcik, Priority scheduling disciplines in queueing network models of computer systems, in: Proceedings of the IFIP Congress 77, Amsterdam, 1977, pp. 565–570.
[26] D. Stanford, Interdeparture time distributions in priority Mi/Gi/1 queue, Performance Eval. 12 (1991) 43–60.
[27] D. Stanford, S. Drekis, Interdeparture time distributions in the nonpreemptive priority Gi/Mi/1 queues, Queue. Syst., Theor. Appl. 1 (2000) 1–22.
[28] D. Stanford, W. Fischer, The interdeparture-time distribution for each class in the Mi/Gi/1 queue, Queue. Syst.: Theor. Appl. 4 (1989) 177–190.
[29] H. Takagi, Queueing analysis: a foundation of performance evaluation, in: Vacation and Priority Systems, vol. 1, North-Holland, 1991.
[30] K. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications, John Wiley and Sons, 2002.
[31] V. Vishnevsky, Z. Krougly, Optimization of closed stochastic networks, Autom. Remote Control 2 (1987) 173–183.
[32] V. Vishnevsky, Theoretical Foundations for Computer Network Design, Technosphera, Russian, Moscow, 2003.
[33] A. Williams, R. Bhandiwad, A generating function approach to queueing network models of multiprogrammed computer systems, Network 6 (1976) 1–22.

Zinovi Krougly is an Adjunct Professor in the Department of Statistical & Actuarial Sciences at the University of Western Ontario. He received his PhD in 1984 from the Institute of Control Problems, Russian Academy of Science. He worked at the Central Research Institute for Applied Computer Science in Minsk until 1998. His research interests include performance evaluation of distributed computer and communication networks, queueing networks, stochastic modelling and optimization, and C++ scientific computing. His previous papers include several in *Automation and Remote Control*, and elsewhere.

David Stanford is a Professor in the Department of Statistical & Actuarial Sciences at the University of Western Ontario. He received his PhD in 1981 from Carleton University, and worked until 1986 at Bell Northern Research in Ottawa. His research interests include queues and risk processes, call centers and telecommunications, and stochastic modelling of natural phenomena. His has published previously in *Performance Evaluation*, *Operations Research*, *Queueing Systems*, *Journal of Applied Probability*, *ASTIN Bulletin*, and elsewhere. He is a member and past President of the Canadian Operational Research Society, and a member of INFORMS.