An Efficient Method to Compute the Rate Matrix for Retrial Queues with Large Number of Servers

Tien Van Do

Department of Telecommunications,
Budapest University of Technology and Economics,
H-1117, Magyar tudósok körútja 2., Budapest, Hungary.

Ram Chakka

Meerut Institute of Engineering and Technology (MIET),
Meerut, India 250005.

Abstract

The approximate solution technique for the main $M/M/c$ retrial queue based on the homogenization of the model employs a quasi-birth-death (QBD) process in which the maximum retrial rate is restricted above a certain level. This approximated continuous-time Markov chain (CTMC) can be solved by the matrix-geometric method, which involves the computation of the rate matrix R. This paper is motivated by two observations. Firstly, retrial queues for the performability analysis of telecommunication systems often involve the number of servers in the order of several hundreds of thousands. Secondly, there are no workable solutions till now for systems with such large number of servers, due ill-conditioning or prohibitively large computation times. Our paper is the first to tackle the problem of large number of servers, very efficiently, in the homogenized $M/M/c$ retrial queue which has paramount applications in networks. We present an efficient algorithm with the time complexity of only $O(c)$ to compute the rate matrix R.

Keywords: stochastic models, telecommunications, retrial queues, matrix-geometric method, spectral expansion, homogenization

Email addresses: do@hit.bme.hu (Tien Van Do), ramchakka@yahoo.com (Ram Chakka).

1 Introduction

The main $M/M/c$ retrial queue is very useful to model several resource sharing problems in telecommunication systems (see [1–6] and references therein). In it, inter-arrival times of customers are exponentially distributed with parameter λ (i.e. the cumulative distribution function –CDF– of inter-arrival times is $1 - e^{-\lambda t}$). Holding-times are exponentially distributed with parameter α (i.e. CDF is $1 - e^{-\alpha t}$). The number of servers is c. Random variable $I(t)$ represents the number of occupied servers at time t, hence $0 \leq I(t) \leq c$ holds. A client who does not receive the allocation of a server upon his arrival because of the unavailability of servers (which happens when $I(t) = c$) joins the orbit in order to wait and retry. Let $J(t)$ be the number of clients in the orbit waiting for retrial at time t. Each customer retries with rate μ. Hence, the total effective retrial rate when $J(t) = j$, is $\mu_j = j \cdot \mu$.

This system can be represented by a two-dimensional continuous-time Markov chain (CTMC) $Y = \{I(t), J(t)\}$ with state space $\{0, 1, \dots, c\} \times \{0, 1, \dots\}$. Let the steady state probabilities of this CTMC be denoted by $\pi_{i,j} = \lim_{t\to\infty} \Pr(I(t) = i, J(t) = j)$. Define the row vector $\mathbf{v}_j = [\pi_{0,j}, \dots, \pi_{c,j}]$. Identifiers i and j also are used for phase and level respectively.

It is well known that the stationary probabilities of the main $M/M/c$ retrial queue with $c > 2$ can be computed only by using approximate techniques [1,7]. A well-known approximation is based on the truncation of the state space at level m, a sufficiently large integer. Only $\pi_{i,j}$ ($(i, j) \in \{0, \dots, c\} \times \{0, 1, \dots, m\}$) are then computed recursively as in [1], assuming $\pi_{i,j} = 0$ (for, $j > m$). Note that m should be selected so large that it gives rise to results with the required accuracy.

Another approximation called the *homogenization* of the model was pioneered by Neuts and Rao [8], where the main $M/M/c$ retrial queue is approximated by the multiserver retrial queue with the *total retrial rate* $\mu_j = min(J(t), N) \cdot \mu$. This means, the retrial times are exponentially distributed with parameter $\nu = N \cdot \mu$ and do not dependent on the number of clients in the orbit as long as the orbit has the number of clients greater than the specified value N. Note that the discussion for the choice of N is presented in the recent book by Artalejo and Gómez-Corral on retrial queues [1]. With this assumption, $\mathbf{v}_j$ can be obtained by any of the several algorithms [9–14] based on the matrix-geometric method (MGM). Two key steps in this method are the computation of the rate matrix R, and solving a system of linear simultaneous equations. It is well known that the algorithms in [9–11,13] have a computational complexity of $O(c^3)$, for each of these two key steps. When c is very large, of the order of tens or hundreds of thousands which is the case in many applications connected with emerging telecommunication systems, many of the existing methods fail due to ill-conditioning or prohibitively large computational time requirements.

This paper proposes a new solution algorithm for obtaining the rate matrix R, for the homogenized model. Our algorithm is numerically highly stable even for large c values and with a computational complexity of $O(c)$ only.

2 Notations and Definitions

2.1 Notation

We deal with the homogenized system, that is the CTMC Y with $\mu_j = min(J(t), N) \cdot \mu$, unless stated otherwise. It is driven by the following transitions.

(a) $A_j(i,k)$ denotes the transition rate from state (i,j) to state (k,j) $(0 \leq i,k \leq c\ ;\ j = 0,1,\ldots)$, which is caused by either the arrival of a customer (when $i < c$) or the leaving of a client after the expiry of a holding-time. The holding-time is exponentially distributed with parameter α. Matrix A_j is of size $(c+1) \times (c+1)$ with elements $A_j(i,k)$. Since A_j is j-independent, it can be written as $A_j = A$. The nonzero elements of A_j are $A_j(i,i-1) = i\alpha$ for $i = 1,\ldots,c+1$, and $A_j(i,i+1) = \lambda$ for $i = 0,\ldots,c$.

(b) $B_j(i,k)$ represents the one-step upward transition rate from state (i,j) to state $(k,j+1)$ $(0 \leq i,k \leq c\ ;\ j = 0,1,\ldots)$, which is due to the arrival of a request when all servers are busy (*i.e.*, when $i = c$), thus increasing $J(t)$ by 1. Matrix B_j (B, since it is j-independent) is of size $(c+1) \times (c+1)$ with elements $B_j(i,k)$. The only nonzero element of B_j is $B_j(c,c) = \lambda$.

(c) $C_j(i,k)$ is the transition rate from state (i,j) to state $(k,j-1)$ $(0 \leq i,k \leq c\ ;\ j = 1,2,\ldots)$, which is due to the successful retrial of a request from the orbit. Matrix C_j is of size $(c+1) \times (c+1)$ with its elements $C_j(i,k)$. The nonzero elements of C_j $(j \geq 1)$ are $C_j(i,i+1) = \mu_j$ for $i = 0,\ldots,c$.

For $j \geq N$, we have $\mu_j = \nu = N\mu$. Therefore, C_j $(j \geq N)$ is j- independent, and let $C = C_j$ $(j \geq N)$. $C_0 = 0$ by definition.

Define D^Z $(Z = A, C, C_1, C_2, \ldots, C_{N-1})$, as a diagonal matrix whose diagonal element is the sum of all elements in the corresponding row of Z. The infinitesimal generator matrix of Y can be written as follows,

$$\begin{bmatrix} Q_1^{(0)} & Q_0^{(0)} & 0 & \dots & \dots & \dots & \dots & \dots & \dots \\ Q_2^{(1)} & Q_1^{(1)} & Q_0^{(1)} & 0 & \dots & \dots & \dots & \dots & \dots \\ 0 & Q_2^{(2)} & Q_1^{(2)} & Q_0^{(2)} & 0 & \dots & \dots & \dots & \dots \\ 0 & 0 & Q_2^{(3)} & Q_1^{(3)} & Q_0^{(3)} & 0 & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \dots & Q_2 & Q_1 & Q_0 & \dots & \dots & \\ \dots & \dots & \dots & \dots & Q_2 & Q_1 & Q_0 & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & Q_2 & Q_1 & Q_0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}, \qquad (1)$$

where $Q_0 = B$, $Q_1 = A - D^A - B - D^C$, $Q_2 = C$, and $Q_0^{(j)} = B,\ Q_1^{(j)} = A - D^A - B - D^{C_j},\ Q_2^{(j)} = C_j\ \ (j = 0, 1, \dots)$.

Then, the balance equations and the normalization equation pertaining to the CTMC Y are

$$\mathbf{v}_0 Q_1^{(0)} + \mathbf{v}_1 Q_2^{(1)} = \mathbf{0}, \qquad (2)$$

$$\mathbf{v}_{j-1} Q_0^{(j-1)} + \mathbf{v}_j Q_1^{(j)} + \mathbf{v}_{j+1} Q_2^{(j+1)} = \mathbf{0}\ \ (j \geq 1), \qquad (3)$$

$$\sum_{j=0}^{\infty} \mathbf{v}_j \mathbf{e}_{c+1}^T = 1.0\ \ (\mathit{normalization}). \qquad (4)$$

Note that $\mathbf{e}_{c+1}$ is the row vector of size $c+1$ with each element equal to unity. For $j \geq N$, equation (3) can be rewritten as

$$\mathbf{v}_{j-1} Q_0 + \mathbf{v}_j Q_1 + \mathbf{v}_{j+1} Q_2 = \mathbf{0}\ \ (j \geq N). \qquad (5)$$

The coefficient matrices in the difference equations (5) are j- independent. This leads to the following solution based on the MGM.

$$\mathbf{v}_j = \mathbf{v}_{N-1} R^{j-N+1}\ \ (j \geq N), \qquad (6)$$

where R is the unique minimal nonnegative solution of the quadratic matrix equation $Q_0 + RQ_1 + R^2 Q_2 = 0$ (cf. [9,11]). After the computation of R, the rate matrix, the steady state probabilities for states $j \leq N-1$ can be determined by solving the balance equations pertaining to the levels $j \leq N$ and the normalization equation. R can be computed by the original algorithm of the MGM [9] and further improved algorithms of MGM [10,11,13]. However, the time complexity of these algorithms is $O(c^3)$. But, for large c values, of the order of tens or hundreds of thousands, there have not been workable (with numerical stability and affordable computation times) solutions, so far in the literature. In what follows, we provide an efficient algorithm to calculate R, with time complexity of $O(c)$ only, which is numerically stable for large c values.

2.2 Eigenvalues and Eigenvectors of the Characteristic Matrix Polynomial

$Q(x) = Q_0 + Q_1 x + Q_2 x^2$ is the *characteristic matrix polynomial* associated with the difference equations (5) or with Y. In [15,12], it is shown that the steady state probabilities of the CTMC are closely related to the left eigenvalue-eigenvector pairs $(x, \boldsymbol{\psi})$ of $Q(x)$. They satisfy,

$$\boldsymbol{\psi} Q(x) = \mathbf{0}; \quad det[Q(x)] = 0 \,. \tag{7}$$

$Q(x)$ is a tri-diagonal matrix of size $(c+1) \times (c+1)$, can be obtained as,

$$Q(x) = \begin{bmatrix} q_{1,1}(x) & q_{1,2}(x) & 0 & \dots & 0 & 0 & 0 \\ q_{2,1}(x) & q_{2,2}(x) & q_{2,3}(x) & \dots & 0 & 0 & 0 \\ 0 & q_{3,2}(x) & q_{3,3}(x) & q_{3,4}(x) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \dots & q_{c,c-1}(x) & q_{c,c}(x) & q_{c,c+1}(x) \\ 0 & 0 & & \dots & 0 & q_{c+1,c}(x) & q_{c+1,c+1}(x) \end{bmatrix}$$

where

$$\begin{aligned} q_{1,1}(x) &= -(\lambda + \nu)x, \\ q_{i,i}(x) &= -(\lambda + \nu + (i-1)\alpha)x \quad (i = 2, \dots, c), \\ q_{c+1,c+1}(x) &= \lambda - (\lambda + c\alpha)x, \\ q_{i,i+1}(x) &= \lambda x + \nu x^2 \qquad (i = 1, \dots, c), \\ q_{i+1,i}(x) &= \alpha \cdot i \cdot x \qquad (i = 1, \dots, c). \end{aligned}$$

In many applications pertaining to telecommunications networks, the value of c can be as large as tens of thousands or even more. Traditional algorithms to compute the eigenvalues and eigenvectors can fail with ill-conditioning, or produce inaccurate results using tremendous computational time, for such large values of c. We discover in this paper certain nice spectral properties of $Q(x)$, and explore these properties to bring out a greatly faster computational algorithm for computing the eigenvalues and eigenvectors, and then the rate matrix R. Characteristic matrix polynomial $Q(x)$ has c zero-eigenvalues $x_1, \ldots, x_c$ (*null-eigenvalues*) with corresponding independent left eigenvectors $\boldsymbol{\psi}_1 = [1, 0, \ldots, 0]$, $\boldsymbol{\psi}_2 = [0, 1, 0, \ldots, 0]$,...,$\boldsymbol{\psi}_c = [0, 0, \ldots, 1, 0]$, respectively. This can be easily verified, by substitution in equations (7).

If the system is ergodic (which is so when $\lambda < c\alpha$), then the number of eigenvalues of $Q(x)$, which are strictly inside the unit disk, has to be $c + 1$ (cf. [12,15]). Therefore, when the system is ergodic, $Q(x)$ should have a single non-zero eigenvalue x_{c+1} strictly inside the unit disk because $Q(x)$ has c zero-eigenvalues. Let $\boldsymbol{\psi}_{c+1}$ be the left eigenvector of $Q(x)$ corresponding to the left eigenvalue x_{c+1}.

The steady state probabilities $\mathbf{v}_j$ can be expressed, using the spectral expansion method [12,15], as

$$\mathbf{v}_j = \sum_{k=1}^{c+1} b_k \boldsymbol{\psi}_k x_k^{j-N+1} \quad (j \geq N-1) \tag{8}$$

where b_i are suitable coefficients which can be determined using the balance equations pertaining to rows 0 to $N-1$, and the normalization equation. Since the probabilities are non-negative, x_{c+1} is real and $0 < x_{c+1} < 1$ holds.

3 Main Result

3.1 An Algorithm to Compute the R Matrix

Let us introduce

$$\Psi = \begin{bmatrix} \psi_{1,1} & \psi_{1,2} & \psi_{1,3} & \cdots & \psi_{1,c+1} \\ \psi_{2,1} & \psi_{2,2} & \psi_{2,3} & \cdots & \psi_{2,c+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \psi_{c,1} & \psi_{c,2} & \psi_{c,3} & \cdots & \psi_{c,c+1} \\ \psi_{c+1,1} & \psi_{c+1,2} & \psi_{c+1,3} & \cdots & \psi_{c+1,c+1} \end{bmatrix}, \tag{9}$$

where $\boldsymbol{\psi}_i = [\psi_{i,1}, \psi_{i,2}, \ldots, \psi_{i,c+1}]$ for $i = 1, 2, \ldots, c+1$.

Based on equations (6) and (8), the rate matrix R can be obtained from the eigenvalues and eigenvectors of $Q(x)$ using simple algebraic work, as follows

$$
\begin{aligned}
R &= \Psi^{-1} \cdot diag(0, 0, \ldots, 0, x_{c+1}) \cdot \Psi \\
&= \begin{bmatrix}
0 & 0 & 0 & \ldots 0 & 0 \\
0 & 0 & 0 & \ldots 0 & 0 \\
\vdots & \vdots & \vdots & \vdots\ \ \vdots & \vdots \\
0 & 0 & 0 & \ldots 0 & 0 \\
\psi_{c+1,1}x_{c+1} & \psi_{c+1,2}x_{c+1} & \psi_{c+1,3}x_{c+1} & \ldots\ \psi_{c+1,c}x_{c+1} & x_{c+1}
\end{bmatrix}.
\end{aligned}
\tag{10}
$$

As the consequence, computation of R boils down to the computation of x_{c+1} and $\boldsymbol{\psi}_{c+1}$. Due to the tri-diagonal structure, the component matrices of the LU decomposition of $Q(x_{c+1})$ can be written as follows

$$
L(x_{c+1}) = \begin{bmatrix}
l_1(x_{c+1}) & 0 & 0 \ldots & 0 & 0 & 0 \\
\alpha x_{c+1} & l_2(x_{c+1}) & 0 \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots\ \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \ldots & \alpha(c-1)x_{c+1} & l_c(x_{c+1}) & 0 \\
0 & 0 & \ldots & 0 & \alpha c x_{c+1} & l_{c+1}(x_{c+1})
\end{bmatrix}, \tag{11}
$$

$$
U(x_{c+1}) = \begin{bmatrix}
1 & u_1(x_{c+1}) & \ldots & 0 & 0 & 0 & 0 \\
0 & 1 & u_2(x_{c+1}) & \ldots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & & \ldots & 0 & 1 & u_c(x_{c+1}) \\
0 & 0 & & \ldots & 0 & 0 & 1
\end{bmatrix}, \tag{12}
$$

Here, $l_i(x_{c+1})$ $(i = 1, \ldots, c+1)$ and $u_i(x_{c+1})$ $(i = 1, \ldots, c)$ are the elements of $L(x_{c+1})$ and $U(x_{c+1})$, respectively. By equating the corresponding elements of $Q(x_{c+1})$ and $L(x_{c+1}) \cdot U(x_{c+1})$, and using some algebraic simplification, we get

$$
l_1(x_{c+1}) = q_{1,1}(x_{c+1}) = -(\lambda + \nu)x_{c+1}, \tag{13}
$$

$$
l_i(x_{c+1}) + \alpha(i-1)x_{c+1}u_{i-1}(x_{c+1}) = q_{i,i}(x_{c+1}), \quad (i = 2, \ldots, c+1), \tag{14}
$$

$$
l_i(x_{c+1})u_i(x_{c+1}) = \lambda x_{c+1} + \nu x_{c+1}^2, \quad (i = 1, \ldots, c). \tag{15}
$$

Since the determinant of a tri-diagonal matrix is the product of its diagonal entries, we can write

$$Det[Q(x_{c+1})] = Det[L(x_{c+1})]Det[U(x_{c+1})] = \prod_{i=1}^{c+1} l_i(x_{c+1}). \tag{16}$$

From equations (13),(14) and (15), it can be verified that $l_i(x_{c+1}) \neq 0$ $(1 \leq i \leq c)$. Hence, $Det[Q(x_{c+1})] = 0$ (from equation (7)) gives rise to $l_{c+1}(x_{c+1}) = 0$. This means x_{c+1} is the root of $l_{c+1}(x)$.

To compute the root of $l_{c+1}(x)$ in interval $(0, 1)$, several alternative algorithms such as bisection, secant method, false position method, Dekker's algorithm and Brent's method (the interested reader can find the implementation of these algorithms in [16]) can be applied. In this paper, we have used Brent's method [17], applied to the set of equations (13),(14) and (15), to find x_{c+1} in the interval $(0, 1)$.

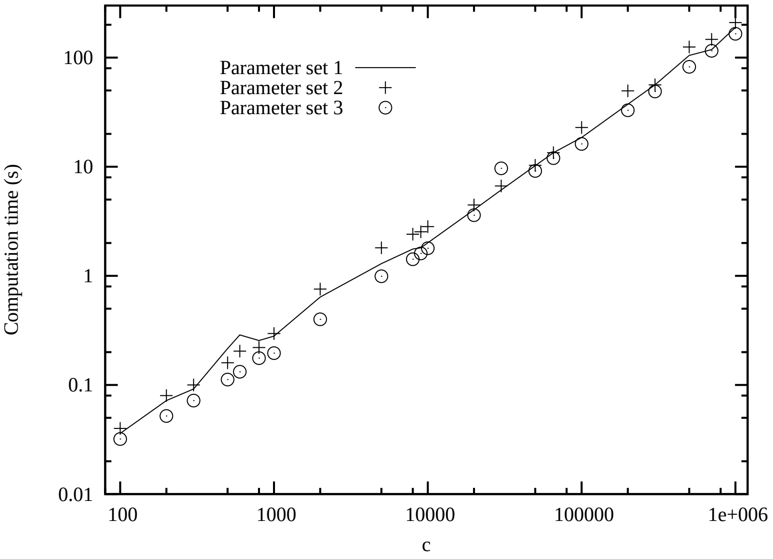

Fig. 1. Computational time versus c, the number of servers (the first parameter set $\lambda = 0.8 \times c \times \alpha$, $1/\alpha = 120$, $1/\nu = 0.5$, the second parameter set $\lambda = 0.9 \times c \times \alpha$, $1/\alpha = 60$, $1/\nu = 0.2$), the third parameter set $\lambda = 0.2 \times c \times \alpha$, $1/\alpha = 90$, $1/\nu = 0.6$

Since $(x_{c+1}, \boldsymbol{\psi}_{c+1})$ are left eigenvalue-eigenvector pair, we have,

$$\begin{aligned}
\boldsymbol{\psi}_{c+1}Q(x_{c+1}) &= \mathbf{0}, \\
\boldsymbol{\psi}_{c+1}L(x_{c+1})U(x_{c+1}) &= \mathbf{0}, \\
\boldsymbol{\psi}_{c+1}L(x_{c+1})U(x_{c+1})U(x_{c+1})^{-1} &= \mathbf{0}U(x_{c+1})^{-1}, \quad \text{because } U(x_{c+1}) \text{ is non-singular}, \\
\boldsymbol{\psi}_{c+1}L(x_{c+1}) &= \mathbf{0}. \qquad (17)
\end{aligned}$$

Expanding equation (17) we obtain the recursive relations $\psi_{c+1,i} = \frac{-i\alpha x_{c+1}\psi_{c+1,i+1}}{l_i(x_{c+1})}$ between $\psi_{c+1,i}$ and $\psi_{c+1,i+1}$, for $i = c, \ldots, 1$.

An eigenvector remains as the eigenvector corresponding to the same eigenvalue when multipled by a scalar. Using this property, we can determine $\boldsymbol{\psi}_{c+1} = [\psi_{c+1,1}, \psi_{c+1,2}, \ldots, \psi_{c+1,c+1}]$ by setting $\psi_{c+1,c+1} = 1$ and using the above recursive relations and equations (13),(14), (15), to compute $\psi_{c+1,i}$ for $i = c, \ldots, 1$.

3.2 Computational Time Complexity

Proposition 1 *The computational time complexity of the proposed algorithm in Section 3.1 is of $O(c)$.*

Proof. Kerber [18] rigourously proved that the number of iterative steps of root finding algorithms depends only on the interval (it is $(0, 1)$ in the present paper), the number of bits used to represent numbers in machines and the assumed tolerance (i.e. the error would be smaller than the tolerance). We have rigorously shown that $l_{c+1}(x)$ has a single root in $(0, 1)$. Based on equations (13), (14) and (15), the computation of $l_{c+1}(x)$ for a given x requires the execution of a loop statement whose action block must be repeated exactly c times. The action block involves only some elementary arithmetic operations. This can then be summarized by concluding that the computational time complexity of root finding of $l_{c+1}(x)$ would be $O(c)$. In addition, ψ is also determined in c steps. Therefore, the computational complexity of our algorithm has to be of $O(c)$. □

Proposition 1 is indeed supported empirically in our experiments as illustrated below. In Figure 1, we plot the computational time of the proposed algorithm versus c on a machine with Intel® Xeon® E5410 2.33GHz processor (note that the algorithm is implemented in Mathematica). The computational time complexity of our analytical method is of $O(c)$ as confirmed in Figure 1 with three parameter sets (the first parameter set $\lambda = 0.8 \times c \times \alpha$, $1/\alpha = 120$, $1/\nu = 0.5$, the second parameter set $\lambda = 0.9 \times c \times \alpha$, $1/\alpha = 60$, $1/\nu = 0.2$ and the third parameter set $\lambda = 0.2 \times c \times \alpha$, $1/\alpha = 90$, $1/\nu = 0.6$). Note that the number c of servers varies between 100 and 10^6. Similar observation can be obtained with other parameter values as well.

4 Conclusions

Discovering and exploring certain nice spectral properties of the characteristic matrix polynomial $Q(x)$ of the CTMC Y, we are able to develop a new algorithm for the computation of R. This has a computational complexity of $O(c)$ which is indeed a very significant improvement in the computation times. Our algorithm is applicable to large c values (as well as small c values) and its numerical stability is practically established, which likely opens a new application opportunity for performance evaluation in emerging telecommunication systems.

Acknowledgement

Ram Chakka thanks the Chancellor, Sri Sathya Sai University, Prasanthi Nilayam, India, for inspiration and support.

References

[1] J. R. Artalejo, A. Gómez-Corral, Retrial Queueing Systems, Springer-Verlag, Berlin Heidelberg, 2008.

[2] J. R. Artalejo, A. Economou, M. Lopez-Herrero, Algorithmic approximations for the busy period distribution of the M/M/c retrial queue, European Journal of Operational Research 176 (2007) 1687–1702.

[3] J. R. Artalejo, A. Economou, A. Gómez-Corral, Applications of maximum queue lengths to call center management, Computers & OR 34 (4) (2007) 983–996.

[4] J. R. Artalejo, M. J. Lopez-Herrero, Cellular mobile networks with repeated calls operating in random environment, Comput. Oper. Res. 37 (7) (2010) 1158–1166.

[5] J. R. Artalejo, Accessible bibliography on retrial queues: Progress in 2000-2009, Mathematical and Computer Modelling (2010) doi:http://dx.doi.org/10.1016/j.mcm.2009.12.011.

[6] P. Wüchner, J. Sztrik, H. de Meer, Finite-source M/M/S retrial queue with search for balking and impatient customers from the orbit, Computer Networks 53 (8) (2009) 1264–1273.

[7] T. Phung-Duc, H. Masuyama, S. Kasahara, Y. Takahashi, M/M/3/3 and M/M/4/4 retrial queues, Journal of Industrial and Management Optimization 5 (2009) 431–451.

[8] M. F. Neuts, B. M. Rao, Numerical investigation of a multiserver retrial model, Queueing Syst. 7 (2) (1990) 169–189.

[9] M. F. Neuts, Matrix Geometric Soluctions in Stochastic Model, Johns Hopkins University Press, Baltimore, 1981.

[10] V. Naoumov, U. Krieger, D. Wagner, Analysis of a Multi-server Delay-loss System with a General Markovian Arrival Process, in: S. Chakravarthy, A. Alfa (Eds.), Matrix-analytical methods in Stochastic models, Marcel Dekker, 1997, pp. 43–66.

[11] G. Latouche, V. Ramaswami, Introduction to Matrix Analytic Methods in Stochastic Modeling, ASA-SIAM Series on Statistics and Applied Probability, 1999.

[12] I. Mitrani, R. Chakka, Spectral expansion solution for a class of Markov models: Application and comparison with the matrix-geometric method, Performance Evaluation 23 (1995) 241–260.

[13] D. A. Bini, G. Latouche, B. Meini, Numerical Methods for Structured Markov Chains (Numerical Mathematics and Scientific Computation), Oxford University Press, Inc., New York, NY, USA, 2005.

[14] V. Naoumov, U. R. Krieger, D. Warner, Analysis of a Multi-Server Delay-Loss System With a General Markovian Arrival Process, in: S. R. Chakravarthy, A. S. Alfa (Eds.), Matrix-analytic methods in stochastic models, Vol. 183 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, 1996, pp. 43–66.

[15] R. Chakka, Performance and Reliability Modelling of Computing Systems Using Spectral Expansion, Ph.D. thesis, University of Newcastle upon Tyne (Newcastle upon Tyne) (1995).

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes 3rd Edition: The Art of Scientific Computing, Cambridge University Press, New York, NY, USA, 2007.

[17] R. Brent, Algorithms for Minimization without Derivatives, Prentice-Hall, Englewood Cliffs, NJ, 1972.

[18] M. Kerber, On the complexity of reliable root approximation, in: V. Gerdt, E. Mayr, E. Vorozhtsov (Eds.), Computer Algebra in Scientific Computing - 11th International Workshop, CASC 2009, Kobe, Japan, Vol. 5743 of Lecture Notes in Computer Science, Springer, 2009, pp. 155–167.