Research Article

A Comparative Numerical Study of the Spectral Theory Approach of Nishimura and the Roots Method Based on the Analysis of BDMMAP/G/1 Queue

Arunava Maity and U. C. Gupta

Department of Mathematics, Indian Institute of Technology, Kharagpur 721302, India

Correspondence should be addressed to U. C. Gupta; umesh@maths.iitkgp.ernet.in

Received 18 September 2014; Revised 31 December 2014; Accepted 5 January 2015

Academic Editor: Lukasz Stettner

Copyright © 2015 A. Maity and U. C. Gupta. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper considers an infinite-buffer queuing system with birth-death modulated Markovian arrival process (BDMMAP) with arbitrary service time distribution. BDMMAP is an excellent representation of the arrival process, where the fractal behavior such as burstiness, correlation, and self-similarity is observed, for example, in ethernet LAN traffic systems. This model was first appraised by Nishimura (2003), and to analyze it, he proposed a twofold spectral theory approach. It seems from the investigations that Nishimura's approach is tedious and difficult to employ for practical purposes. The objective of this paper is to analyze the same model with an alternative methodology proposed by Chaudhry et al. (2013) (to be referred to as CGG method). The CGG method appears to be rather simple, mathematically tractable, and easy to implement as compared to Nishimura's approach. The Achilles tendon of the CGG method is the roots of the characteristic equation associated with the probability generating function (pgf) of the queue length distribution, which absolves any eigenvalue algebra and iterative analysis. Both the methods are presented in stepwise manner for easy accessibility, followed by some illustrative examples in accordance with the context.

1. Introduction

In telecommunication systems, designing robust and substantial network system has become a very formidable task. Extensive studies on ethernet traffic show that, correlation, burstiness, and self-similarity are very common in the arrival rates of packaged data. Not only packetised traffic exhibits self-similar or fractal characteristics, but also there are several processes like fractional Brownian motion for storage system, autoregressive integrated moving average models, Poisson Pareto burst processes, and so forth admitting self-similarity. Conventional traffic models are not applicable to these types of self-similar traffic models. To tackle these types of communicating queues, several techniques are proposed, for example, modelling a queue with a heavy-tailed service time distribution. Among the available techniques, one of the most exquisite approaches is to consider Markovian arrival process, introduced by Neuts [1] to deal with such correlated

and bursty traffic systems; for example, see Lucantoni [2] and Chakravorthy [3].

It is quite evident from the existent literature that the preeminent challenge in analyzing MAP/G/1 queue lies in extracting the boundary probabilities accurately. Insofar as, if these probabilities are known, there are several methods available in the literature to determine the remaining state probabilities and queueing-time distributions; for example, see Ramaswami [4, 5], Bini and Meini [6]. The most classical approach to get that stationary vector is the matrix-analytic method (MAM), developed by Neuts [7]. In this method, the determination of the boundary vector $\mathbf{q}$ depends on the evaluation of the vector $\mathbf{g}$. For that we need to find the $\mathbf{G}$ matrix. Once $\mathbf{G}$ is computed then system-length probabilities and other performance measures can be obtained by exploiting it. Several algorithms have been advocated in the literature for the evaluation of $\mathbf{G}$, for example, cyclic reduction algorithm by Bini et al. [8, 9]. But all these proposed algorithms have

some serious computational issues. The reasons behind these difficulties are multidimensional; for example, it involves truncation error as it includes several matrix iterations and approximations; convergence becomes very slow for heavy traffic conditions. The spectral theory approach is one of the most widely used alternative methods available in the literature to obtain the unknown boundary vector; see, for example, Lee et al. [10], Gail et al. [11]. Although there are several published algorithms and theorems regarding evaluation of state vectors for these types of queues, its computational difficulties have been a major concern to the practitioners. To get the flavor of the difficulty arising in computation of $\mathbf{g}$ as well as state vectors in MAP/G/1 type queues, we would like to refer to Lee et al. [12], where he says, “*Even with these low-order parameter matrices, it is still a formidable task for practitioners, who have completed introductory queueing text, to understand the published theories and write their own computer code even for the simplest MAP/G/1 queues.*”

This paper concerns a queueing system with birth-death-modulated Markovian arrival process (BDMMAP), introduced by Nishimura [13], where the service time is arbitrarily distributed. BDMMAP is a special type of Markovian arrival process whose underlying process is a birth-death process with a finite state space. This process has several real life applications for details; see Nishimura [13]. He proposes a twofold spectral approach to analyze this queue. In his paper, he identifies the zeros of $|z\mathbf{I} - \mathbf{A}(z)|$ in the unit disk (which are found to be real and distinct) using a modified bisection method and subsequently presents the $\mathbf{G}$ matrix in terms of the zeros and associated right null vectors of $z\mathbf{I} - \mathbf{A}(z)$. Further, to calculate the stationary distribution at the service completion epochs, he has proposed an algorithm based on Fourier inversion formula of the generating function $\mathbf{Y}^+(z)$ of the stationary probability distribution.

In this paper, our objective is to compare the CGG approach with that of Nishimura to analyze the present model in terms of computational difficulties. In this regard, we would like to emphasize on the fact that if a practitioner or an engineer who has basic knowledge of spectral theory and understood the method of Nishimura will face extreme difficulty to numerically implement this. Also, in his paper [13], the state vectors are not presented numerically for further implementation of the users. These are the facts which actually motivated us to advocate an alternative approach, namely, CGG method, to analyze the present model, which seems to be easily understandable and rather easy for numerical demonstration. The main advantages of this approach over Nishimura are that one does not need to evaluate any eigenvalues and eigenvectors (especially in terms of z) for the analysis of the present model, which are the essential parts in Nishimura's approach. Here, by CGG method, we first evaluate the vector $\mathbf{q}$, using roots inside the unit circle of the numerator of $|z\mathbf{I} - \mathbf{A}(z)|$ and then evaluate the rest of the state probabilities by inverting the vector generating function (VGF) using the roots outside the unit circle. Further for the validation of our objective, we present the steady state probabilities at both arbitrary epoch and postdeparture epoch in tabular form, obtained from CGG method for high order arrival matrices, by writing codes in Maple 13 software.

Also to check the accuracy of the two methods, we compare $\mathbf{q}$ numerically obtained from both the methods and observe that these values agree up to a desirable degree of accuracy.

This paper is organized as follows: Section 1 consists of introduction of the paper; Section 2 includes the model description; Section 3 consists of the analysis of the present model by both CGG method and Nishimura method; in Section 4, we present the numerical results; in Section 5 we provide conclusion and future scope of this paper, followed by the references.

2. Model Description

Although Nishimura [13] presented a detailed description about the queue BDMMAP/G/1 but for the sake of completeness, we here present a brief overview of the model.

2.1. Arrival Process. Here the arrival rate matrices $\{\mathbf{D}_r, r = 0, 1, 2, \dots\}$ corresponding to r batch arrivals are tridiagonal and of the following form:

$$\mathbf{D}_r = \begin{pmatrix} c_{r,1} & a_{r,1} & \cdots & 0 & 0 \\ b_{r,2} & c_{r,2} & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & c_{r,M-1} & a_{r,M-1} \\ 0 & 0 & \cdots & b_{r,M} & c_{r,M} \end{pmatrix}, \quad (1)$$

where $a_{r,j} \geq 0$ ($r = 0, 1, 2, \dots, j = 1, \dots, M-1$), $b_{r,j} \geq 0$ ($r = 0, 1, 2, \dots, j = 2, \dots, M$), $c_{r,j} \geq 0$ ($r = 1, 2, \dots, j = 1, \dots, M$), $c_{0,j} < 0$ ($j = 1, \dots, M$), and the infinitesimal generator of the underlying process is given by $\mathbf{D} = \sum_{r=0}^{\infty} \mathbf{D}_r$; that is,

$$\mathbf{D} = \begin{pmatrix} c_1 & a_1 & \cdots & 0 & 0 \\ b_2 & c_2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & c_{M-1} & a_{M-1} \\ 0 & 0 & \cdots & b_M & c_M \end{pmatrix}, \quad (2)$$

where $a_j = \sum_{r=0}^{\infty} a_{r,j}$, $b_j = \sum_{r=0}^{\infty} b_{r,j}$, $c_j = \sum_{r=0}^{\infty} c_{r,j}$, $j = 1, \dots, M$. In the region $|z| \leq 1$,

$$\mathbf{D}(z) = \begin{pmatrix} c_1(z) & a_1(z) & \cdots & 0 & 0 \\ b_2(z) & c_2(z) & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & c_{M-1}(z) & a_{M-1}(z) \\ 0 & 0 & \cdots & b_M(z) & c_M(z) \end{pmatrix} \quad (3)$$

is assumed to be analytic, so is $a_j(z) = \sum_{r=0}^{\infty} a_{r,j}z^r$, $b_j(z) = \sum_{r=0}^{\infty} b_{r,j}z^r$, $c_j(z) = \sum_{r=0}^{\infty} c_{r,j}z^r$. The stationary probability

vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_M)$ for the underlying process is given by

$$\begin{aligned} \pi_1 &= \left(1 + \frac{a_1}{b_2} + \dots + \frac{a_1 \cdots a_{M-1}}{b_2 \cdots b_M} \right)^{-1}, \\ \pi_j &= \left(\frac{a_1 \cdots a_{j-1}}{b_2 \cdots b_j} \right) \pi_1, \quad j = 2, \dots, M. \end{aligned} \quad (4)$$

The fundamental arrival rate λ is given by $\lambda = \boldsymbol{\pi} \sum_{r=1}^{\infty} r \mathbf{D}_r \mathbf{e}$, where $\mathbf{e}$ is column vector of 1's of appropriate dimension.

2.2. Service Time Distribution. Here the customers are served according to FCFS discipline and by a single server having service time arbitrarily distributed whose cdf, pdf, mean service time, and Laplace-Stieltjes transform are, respectively, $H(t)$, $\tilde{h}(t)$, $1/\mu$, and $h^*(s) = \int_0^{\infty} e^{-st} dH(t)$. Traffic intensity ρ is given by λ/μ .

3. Analysis

In the literature (see Lucantoni [14]), the probability generating function (pgf) $\mathbf{Y}^+(z)$ of the stationary probability vector just after service completion epochs for the $MAP/G/1$ queue is given by

$$\begin{aligned} \mathbf{y}_0^+ &= \lambda^{-1} (1 - \rho) \mathbf{g} (-\mathbf{D}_0), \\ \mathbf{Y}^+(z) &= \lambda^{-1} (1 - \rho) \mathbf{g} \mathbf{D}(z) \mathbf{A}(z) (z\mathbf{I} - \mathbf{A}(z))^{-1}. \end{aligned} \quad (5)$$

Here $\mathbf{Y}^+(z) = \sum_{r=0}^{\infty} \mathbf{y}_r^+ z^r$, where $\mathbf{y}_r^+ = (y_{r,1}^+, \dots, y_{r,M}^+)$ ($r \geq 0$) are the M -dimensional row vector whose j th entry is the stationary joint probability that just after service completion epochs, the arrival phase is j and the number of customers in the system is r .

The main steps involved in the analysis of $MAP/G/1$ type queue are as follows: (i) one needs to evaluate the vector $\mathbf{g}$ ($\mathbf{q} = (1 - \rho)\mathbf{g}$) and (ii) after getting $\mathbf{g}$, one needs to invert the generating function $\mathbf{Y}^+(z)$ to get the stationary probability vector. In the following sections, we briefly discuss the steps required for the analysis of $BDMMAP/G/1$ queue by Nishimura's method and by CGG method, respectively.

3.1. Analysis Done by Nishimura. We here briefly discuss the procedure adopted by Nishimura. For clarity of understanding we divide his approach into two subsequent segments; namely, first one is the evaluation of $\mathbf{g}$ and the second one is the evaluation of steady state probability vector, which is as follows.

3.1.1. The Evaluation of $\mathbf{g}$. In the following, we describe the procedure adopted by Nishimura for the evaluation of $\mathbf{g}$.

(i) Nishimura shows that, for a $BDMMAP/G/1$ queue, the z_j 's ($j = 1, \dots, M$) (zeros in the unit disk $|z| \leq 1$ of $|z\mathbf{I} - \mathbf{A}(z)|$) are distinct and simple and $\mathbf{G}$ can be exhibited as

$$\mathbf{G} = [\mathbf{v}_1, \dots, \mathbf{v}_M] \begin{pmatrix} z_1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & z_M \end{pmatrix} [\mathbf{v}_1, \dots, \mathbf{v}_M]^{-1}, \quad (6)$$

where $\mathbf{v}_j$ are the right null vectors of $z\mathbf{I} - \mathbf{A}(z)$ corresponding to z_j .

(ii) $\mathbf{v}_j$'s are calculated from the following equations:

$$z_j \mathbf{v}_j = \sum_{k=0}^{\infty} \mathbf{A}_k z_j^k \mathbf{v}_j. \quad (7)$$

Accordingly, z_j and $\mathbf{v}_j$ are the eigenvalues and right eigenvectors of the $\mathbf{G}$ matrix. $\mathbf{A}(z)$ is the matrix probability generating function of arrival numbers, that is, of $\mathbf{A}_k$ which is an $M \times M$ matrix and whose (i, j) -entry is the transition probability under the condition that the arrival phase is i at the service starting epoch, the arrival phase at the service completion epoch is j , and the number of arriving customers during the service time is k .

(iii) Now $\mathbf{g}$ is calculated from $\mathbf{g}\mathbf{G} = \mathbf{g}$ and $\mathbf{g}\mathbf{e} = 1$.

3.1.2. The Evaluation of Steady State Probability Vector. The procedure adopted by Nishimura for evaluating the steady state probability vectors is as follows.

(i) In order to derive the steady state probability vector, he has shown that the zeros of $|z\mathbf{I} - \mathbf{A}(z)|$ and $|h^{-1}(z)\mathbf{I} - \mathbf{D}(z)|$ are the same, simple, and real in $(0, 1]$ (see Nishimura, [13, Theorems 1 and 4]), where $h(\alpha) = \int_0^{\infty} e^{\alpha t} dH(t)$. He has further assumed (see Nishimura, [13, page 434, Assumption 1]) that, for $|z| = 1$, eigenvalues $\alpha_j(z)$, ($j = 1, 2, \dots, M$) of $\mathbf{D}(z)$ are simple.

(ii) With the help of the above facts and assumptions, he has represented the generating function in the form

$$\begin{aligned} \mathbf{Y}^+(z) &= \lambda^{-1} (1 - \rho) \mathbf{g} \mathbf{V}(z) \\ &\times \begin{pmatrix} \frac{\alpha_1(z) h(\alpha_1(z))}{z - h(\alpha_1(z))} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \frac{\alpha_M(z) h(\alpha_M(z))}{z - h(\alpha_M(z))} \end{pmatrix} \mathbf{U}(z), \\ \text{for } z = 1, \quad j = 1, \quad \frac{\alpha_1(1) h(\alpha_1(1))}{1 - h(\alpha_1(1))} &= \frac{\lambda}{1 - \rho}, \end{aligned} \quad (8)$$

where

(a)

$$\mathbf{U}(z) = \boldsymbol{\Phi}(z) \mathbf{X}(z)^{1/2}, \quad \mathbf{V}(z) = \mathbf{X}(z)^{-1/2} \boldsymbol{\Phi}(z)^T. \quad (9)$$

(b)

$$\mathbf{X}(z) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \tau_2(z) & \ddots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \tau_M(z) \end{pmatrix}, \quad (10)$$

where $\tau_j(z) = a_1(z) \cdots a_{j-1}(z)/b_2(z) \cdots b_j(z)$, with $\tau_1(z) = 1$.

(c)

$$\Phi(z) = \begin{pmatrix} \Phi_1(z) \\ \vdots \\ \Phi_M(z) \end{pmatrix} = \begin{pmatrix} \phi_{1,1}(z) & \cdots & \phi_{1,M}(z) \\ \vdots & \cdots & \vdots \\ \phi_{M,1}(z) & \cdots & \phi_{M,M}(z) \end{pmatrix} \quad (11)$$

and $\phi_{j,i}(z)$'s are calculated from the following iterative procedure:

(1)

$$\phi_{j,i}(z) = \frac{\psi_{j,i}(z)}{\left(\sum_{j=1}^M \psi_{j,i}(z)^2\right)^{1/2}} \quad (12)$$

(2)

$$\psi_{j,i+1}(z) = \frac{-\{d_i(z) \psi_{j,i-1}(z) + (c_i(z) - \alpha_j(z) \psi_{j,i}(z))\}}{d_{i+1}(z)}, \quad (13)$$

where $d_i(z) = (a_{i-1}(z)b_i(z))^{1/2}$ is a complex number with $\psi_{j,1}(z) = 1$ and $\psi_{j,2}(z) = -(c_i(z) - \alpha_j(z))/d_2(z)$, where $\psi_j(z)$ is a left eigenvector corresponding to the eigenvalue $\alpha_j(z)$, ($j = 1, 2, \dots, M$) of

$$\mathbf{F}(z) \equiv \mathbf{X}(z)^{1/2} \mathbf{D}(z) \mathbf{X}(z)^{-1/2}$$

$$= \begin{pmatrix} c_1(z) & d_2(z) & \cdots & 0 & 0 \\ d_2(z) & c_2(z) & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ddots & c_{M-1}(z) & d_M(z) \\ 0 & 0 & \cdots & d_M(z) & c_M(z) \end{pmatrix}, \quad (14)$$

which is a symmetric tridiagonal matrix, with $d_j(z) = \sqrt{a_{j-1}(z)b_j(z)}$.

(iii) After obtaining $\mathbf{Y}^+(z)$, he uses fast Fourier transform (FFT) to get the stationary probabilities. In order to apply FFT he has to determine a number N (preferably $N = 2^m$) so that after that number the remaining probabilities are negligible and further he obtains the approximated state vector as $\mathbf{y}_l^+ = (1/N) \sum_{t=0}^{N-1} \mathbf{Y}^+(\omega^t) \omega^{-tl}$, ($l = 0, 1, \dots, N-1$). In order to calculate the eigenvalues $\alpha_j(\omega^t)$ of $F(\omega^t)$, $\omega = e^{2\pi i/N}$ he uses Durand-Kerner-Aberth method for calculating zeros of a polynomial.

Note 1. We have tried to implement the above method by writing codes in Maple 13 software, in a PC with Intel CORE(TM) i5, 3.10 Ghz processor. Our experience and the realizations during the computation by Nishimura's method are summarized in the following remarks.

Remark 1. An addressable disadvantage in Nishimura's approach is that we need to find the eigenvalues of $\mathbf{D}(z)$ (as well as of $\mathbf{F}(z)$) as a function of z only, using a bisection method based on Givens' algorithm, which may be cumbersome in computation for higher order arrival matrices. In this regard, we would like to quote as in Lee et al. [12] that "...we have experienced difficulties in finding all the eigenvalues as a function of z when the order is high. In this case, commercially available mathematical packages may help." As far as our experience, commonly available mathematical packages may not always solve the present problem up to the desired level.

Remark 2. For the evaluation of $\mathbf{G}$ by Nishimura's method, one has to compute the $\mathbf{v}_j$'s corresponding to each of the z_j 's, which evidently needs a thorough computational involvement even for a moderate value of m . However, it seems to us that computation up to $\mathbf{g}$ or the boundary vector $\mathbf{q}$ by this method is not as difficult as the evaluation of the other stationary vectors.

Remark 3. In order to derive $\mathbf{Y}^+(z)$ by Nishimura's method, one needs to first find $\mathbf{U}(z)$ & $\mathbf{V}(z)$. For that one has evaluate the eigenvalues and eigenvectors of $\mathbf{F}(z)$, which is again a tedious task. After getting these eigenvectors, he uses a recursive algorithm to evaluate $\mathbf{U}(z)$ & $\mathbf{V}(z)$, which may address numerical inaccuracy for high order matrices.

Remark 4. In order to apply FFT on $\mathbf{Y}^+(z)$, he has to calculate the eigenvalues $\alpha_j(\omega^t)$ of $F(\omega^t)$, $\omega = e^{2\pi i/N}$, using the Durand-Kerner-Aberth method. Although, Durand-Kerner-Aberth method has superlinear convergence for simple zeros but its convergence heavily depends on selecting the initial guess. Also, if the concerned polynomial has zeros with large moduli and small moduli at the same time, selecting the initial points becomes rather delicate. Also, for stopping the iteration, one needs a tedious backward rounding error analysis, which asserts that each computed approximation is the zero by a "nearby" polynomial (see Bini [15]). So, according to us, one may not find it very easy from the computation point of view.

Remark 5. Although FFT algorithm has computational complexity $O(n \log n)$ instead of $O(n^2)$ (as in the case of DFT) (if n is of the form 2^k , a one-dimensional FFT of length n requires less than $3n \log_2 n$ floating-point operations (times a proportionality constant)), it looks computationally very much challenging to the users to invert the pgf by FFT algorithm in Nishimura's case, due to the complex structure of the pgf presented by him. We have tried to invert the pgf by FFT for very low order arrival matrices and found it very difficult. So, as far as the authors' experience, this methodology may not be suitable for the practitioners with

TABLE 1: Distribution of stationary probability at arbitrary epoch using CGG method.

$m := 7,$	$\mathbf{D}_0 := \left(\begin{array}{cccccc} -32 & 11 & 0 & 0 & 0 & 0 \\ 1 & -6 & 2 & 0 & 0 & 0 \\ 0 & 1 & -5 & 2 & 0 & 0 \\ 0 & 0 & 3 & -8 & 1 & 0 \\ 0 & 0 & 0 & 8 & -29 & 10 \\ 0 & 0 & 0 & 0 & 3 & -9 \\ 0 & 0 & 0 & 0 & 0 & 7 \\ & & & & & & -39 \end{array} \right)$				$\mathbf{D}_1 := \left(\begin{array}{cccccc} 6 & 15 & 0 & 0 & 0 & 0 \\ 0.8 & 1 & 1.2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0.4 & 3 & 0.6 & 0 \\ 0 & 0 & 0 & 3.6 & 2 & 5.4 \\ 0 & 0 & 0 & 0 & 1.6 & 1 \\ 0 & 0 & 0 & 0 & 0 & 20 \\ & & & & & & 12 \end{array} \right)$			
$\lambda = 3.56099448$	$\rho = 0.24191538$	$z_1 = 0.248921, z_2 = 0.287031, z_3 = 0.300658, z_4 = 0.643693, z_5 = 0.761353, z_6 = 0.824482, z_7 = 1$						
$\boldsymbol{\pi} := [0.01045466, 0.15101174, 0.48323757, 0.25277042, 0.02973770, 0.06464717, 0.00814075], \boldsymbol{\pi}\mathbf{e} = 1$								
l	$\gamma_{l,1}$	$\gamma_{l,2}$	$\gamma_{l,3}$	$\gamma_{l,4}$	$\gamma_{l,5}$	$\gamma_{l,6}$	$\gamma_{l,7}$	$\gamma_l\mathbf{e}$
0	0.0053095	0.1225359	0.3838302	0.1915388	0.0147667	0.0381945	0.0019090	0.7580846
	$= \mathbf{q}_1$	$= \mathbf{q}_2$	$= \mathbf{q}_3$	$= \mathbf{q}_4$	$= \mathbf{q}_5$	$= \mathbf{q}_6$	$= \mathbf{q}_7$	$= \mathbf{q}\mathbf{e} = 1 - \rho$
1	0.0036946	0.0199942	0.0760095	0.0448316	0.0092265	0.0125441	0.0029294	0.1692299
2	0.0010356	0.0060397	0.0165498	0.0111656	0.0033780	0.0070010	0.0015859	0.0467555
3	0.0002948	0.0017013	0.0044458	0.0032516	0.0013115	0.0034591	0.0008467	0.0153108
4	0.0000839	0.0004993	0.0014338	0.0011316	0.0005576	0.0017286	0.0004328	0.0058676
5	0.0000246	0.0001551	0.0005396	0.0004575	0.0002558	0.0008620	0.0002183	0.0025129
6	0.0000076	0.0000523	0.0002280	0.0002043	0.0001226	0.0004299	0.0001094	0.0011542
7	0.0000025	0.0000194	0.0001038	0.0000965	0.0000600	0.0002145	0.0000547	0.0005515
8	0.0000009	0.0000079	0.0000494	0.0000469	0.0000297	0.0001070	0.0000273	0.0002691
9	0.0000004	0.0000035	0.0000240	0.0000231	0.0000147	0.0000534	0.0000136	0.0001327
10	0.0000002	0.0000016	0.0000118	0.0000115	0.0000073	0.0000266	0.0000068	0.0000658
15	0.0000000	0.0000000	0.0000004	0.0000004	0.0000002	0.0000008	0.0000002	0.0000020
≥ 21	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
$\mathbf{P}(\text{phase} = \mathbf{j}) = \overline{\boldsymbol{\pi}}_{\mathbf{j}}$	0.0104547	0.1510117	0.4832376	0.2527704	0.0297377	0.0646472	0.0081408	1.0000000

TABLE 2: Distribution of stationary probability at service completion epoch using CGG method.

l	$\gamma_{l,1}^+$	$\gamma_{l,2}^+$	$\gamma_{l,3}^+$	$\gamma_{l,4}^+$	$\gamma_{l,5}^+$	$\gamma_{l,6}^+$	$\gamma_{l,7}^+$	$\gamma_l^+\mathbf{e}$
0	0.01330220	0.08227482	0.30875159	0.18155513	0.03429153	0.05131165	0.01018140	0.68166832
1	0.00441302	0.02643001	0.07316726	0.04919645	0.01444583	0.02989969	0.00656687	0.20411913
2	0.00130584	0.00758268	0.01981490	0.01444597	0.00574059	0.01501491	0.00364341	0.06754829
3	0.00037519	0.00223206	0.00636291	0.00500380	0.00244431	0.00753719	0.00188202	0.02583747
4	0.00011005	0.00069211	0.00238099	0.00201229	0.00111998	0.00376296	0.00095227	0.01103065
5	0.00003394	0.00023240	0.00100176	0.00089575	0.00053597	0.00187759	0.00047777	0.00505518
6	0.00001130	0.00008587	0.00045501	0.00042229	0.00026223	0.00093662	0.00023891	0.00241224
7	0.00000413	0.00003485	0.00021606	0.00020499	0.00012964	0.00046721	0.00011930	0.00117619
8	0.00000166	0.00001527	0.00010507	0.00010093	0.00006440	0.00023305	0.00005954	0.00057993
9	0.00000072	0.00000706	0.00005174	0.00005003	0.00003206	0.00011625	0.00002971	0.00028757
10	0.00000033	0.00000337	0.00002564	0.00002488	0.00001598	0.00005798	0.00001482	0.00014301
15	0.00000001	0.00000010	0.00000079	0.00000077	0.00000049	0.00000179	0.00000046	0.00000440
20	0.00000000	0.00000000	0.00000002	0.00000002	0.00000002	0.00000006	0.00000001	0.00000014
≥ 25	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
$\mathbf{P}(\text{phase} = \mathbf{j}) = \bar{\boldsymbol{\pi}}_{\mathbf{j}}^*$	0.01955872	0.11959374	0.41235832	0.25393723	0.05909841	0.11127281	0.02418077	1.00000000

basic knowledge of queueing, spectral theory, and computer programming to computationally implement it.

Encountering the difficulty to numerically implement the present queueing model by Nishimura's method, we apply the recently developed CGG method (see [16]) to obtain the steady state vectors. For the sake of completeness, we present

the CGG approach in stepwise manner in the following section.

3.2. CGG Approach. This method exploits the roots of the equation $|z\mathbf{I} - \mathbf{A}(z)| = 0$ that lies within the domain $\Omega \equiv \{|z| \leq 1\}$ and exterior to Ω ; we denote it as Ω^* . In this method,

TABLE 3: Comparison of the vector $\mathbf{q}$ obtained by Nishimura and CGG method for the $BDMMAP/D/1$ queue.

$m := 10,$	$\mathbf{D}_0 := \begin{pmatrix} -32 & 11 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -6 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -5 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & -8 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8 & -29 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & -9 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & -39 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & -51 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -13 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -10 \end{pmatrix}$	$\mathbf{D}_1 := \begin{pmatrix} 12 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 4 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 9 & 13 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 18 & 11 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 6 \end{pmatrix}$
$\boldsymbol{\pi} = [0.01017576, 0.10175763, 0.30527290, 0.22895467, 0.06244218, 0.14569843, 0.02081406, 0.01280865, 0.04483029, 0.06724543]$		
$\boldsymbol{\pi}\mathbf{e} = 1.$		
$\lambda = 5.607735, \rho = 0.70096688, \mu = 8, h^*(s) = e^{s/\mu}$		
$z_1 = 0.00098380, z_2 = 0.00935755, z_3 = 0.01780611, z_4 = 0.02168071, z_5 = 0.19095495, z_6 = 0.31260774, z_7 = 0.49455248,$ $z_8 = 0.55816270, z_9 = 0.60694795, z_{10} = 1$		
$\mathbf{g} := [0.00546278, 0.15683878, 0.47445678, 0.25974316, 0.02097492, 0.07602987, 0.00237815, 0.00020033, 0.00154551, 0.00236971]$		
$\mathbf{q} := [0.00163355, 0.04689999, 0.14187829, 0.07767181, 0.00627220, 0.02273544, 0.00071115, 0.00005991, 0.00046216, 0.00070862],$ $\mathbf{q}\mathbf{e} = 0.29903312 = 1 - \rho$		

they first find the arbitrary epoch probabilities $\mathbf{y}_l$ and thereby obtain the departure epoch probabilities.

In the following, we briefly describe the entire CGG method in stepwise manner. For the sake of easy understanding, we divide the steps in two segments: first one describes the procedure to evaluate $\mathbf{q}$ (as well as $\mathbf{g}$) and the second one is about how they evaluate the other state vectors.

3.2.1. Evaluation of the Vector $\mathbf{q}$

Step 1. We write the pgf as

$$\mathbf{Y}(z)(z\mathbf{I} - \mathbf{A}(z)) = (z - 1)\mathbf{q}\mathbf{A}(z). \quad (15)$$

- (a) $\mathbf{Y}(z) = [Y_1(z), Y_1(z), \dots, Y_M(z)]$ is the vector generating function of $\{y_l\}_0^\infty$ that is, $\mathbf{Y}(z) = \sum_{l=0}^\infty \mathbf{y}_l z^l$ and $Y_j(z)$ is the j th component of $\mathbf{Y}(z)$, that is, $Y_j(z) = \sum_{l=0}^\infty y_{l,j} z^l, (1 \leq j \leq M)$.
- (b) $\mathbf{y}_l = [y_{l,1}, y_{l,2}, \dots, y_{l,M}]$ is a row vector whose j th element denotes the joint probability that there are l ($l \geq 0$) customers in the system and the phase of the arriving customer is j ($1 \leq j \leq M$) at arbitrary epoch.
- (c) $\mathbf{q} = [q_1, q_2, \dots, q_M]$ is a row vector whose j th element represents the joint probability that an arriving customer is in phase j and the server is idle.

Step 2. It is seen that, in applications, most of the distributions have rational Laplace-Stieltjes transform. They exploit this fact. They consider the LST of the service time distribution as $h(s) = P(s)/Q(s)$, where the degree of the polynomial $Q(s)$ is k and that of $P(s)$ is at most k . Now the (i, j) th element of $(z\mathbf{I} - \mathbf{A}(z))$ is given by $(z\mathbf{I} - \mathbf{A}(z))_{i,j} = r_{i,j}(z)/d(z)$, where

$$r_{i,j}(z) = \begin{cases} zd(z) - a_{i,i}(z), & i = j; \\ -a_{i,j}(z), & \text{otherwise.} \end{cases} \quad (16)$$

Step 3. From the system of (15), using Cramer's rule, we can find $Y_j(z) = (z - 1)|\mathbf{R}_j(z)|/|\mathbf{R}(z)|, 1 \leq j \leq M$. We denote $|\mathbf{R}_j(z)|$ and $|\mathbf{R}(z)|$ by $S_j(z)$ and $S(z)$, respectively, where

$$(\mathbf{R}_j(z))_{i,k} = \begin{cases} r_{k,i}(z), & k \neq j; \\ \sum_{j=1}^M a_{j,i}(z) q_j, & k = j, \end{cases} \quad (17)$$

$$1 \leq i, k, j \leq M.$$

Step 4. Each component $\mathbf{Y}_j(z)$ of the VGF $\mathbf{Y}(z)$ given in being analytic in $|z| \leq 1$ implies that $\gamma_1, \gamma_2, \dots, \gamma_m$ must be the zeros of the numerator of $\mathbf{Y}_j(z)$ and hence we can determine the unknown vector $\mathbf{q}$ by considering any one component of $\mathbf{Y}(z)$ say $Y_j(z)$ ($1 \leq j \leq M$). This implies that $S_j(\gamma_i) = 0, 1 \leq i \leq M$ and another equation comes from normalizing condition, that is, $S_j(1) = \pi_j S'(1)$.

3.2.2. Evaluation of the State Probability Vector at Postdeparture Epoch $\mathbf{y}_l^+$. If $S(z)$ has d roots, namely, $\xi_i (1 \leq i \leq d)$, outside the unit circle then by partial fraction method $Y_j(z) = b_{0,j} + \sum_{i=1}^d (b_{i,j}/(z - \xi_i)), j = 1, \dots, M$. We get

$$y_{l,j} = \begin{cases} b_{0,j} - \sum_{i=1}^d \frac{b_{i,j}}{\xi_i}, & l = 0; \\ \sum_{i=1}^d \frac{b_{i,j}}{\xi_i^{l+1}}, & l \geq 1, \end{cases} \quad (18)$$

where,

$$b_{i,j} = \begin{cases} \pi_j - \sum_{k=1}^d \frac{b_{k,j}}{1 - \xi_k}, & i = 0; \\ \frac{(\xi_i - 1) S_j(\xi_i)}{S'(\xi_i)}, & i = 1, 2, \dots, d. \end{cases} \quad (19)$$

TABLE 4: Distribution of stationary probability at arbitrary epoch using CGG method for deterministic service time distribution.

l	$y_{l,1}$	$y_{l,2}$	$y_{l,3}$	$y_{l,4}$	$y_{l,5}$	$y_{l,6}$	$y_{l,7}$	$y_{l,8}$	$y_{l,9}$	$y_{l,10}$	$y_l e$	Tail probability ($l \geq 1$)
0	0.001634	0.046814	0.141828	0.077830	0.006272	0.022737	0.000711	0.000061	0.000441	0.000708	0.299036	
	$\approx q_1$	$\approx q_2$	$\approx q_3$	$\approx q_4$	$\approx q_5$	$\approx q_6$	$\approx q_7$	$\approx q_8$	$\approx q_9$	$\approx q_{10}$	$\approx qe \approx 1 - \rho$	
1	0.002811	0.016898	0.054698	0.047333	0.014415	0.023149	0.001694	0.000382	0.001168	0.001622	0.164170	0.041843
2	0.001795	0.008742	0.021327	0.022088	0.009515	0.017742	0.001806	0.000739	0.001822	0.002387	0.087964	0.038314
3	0.001015	0.005005	0.011847	0.012207	0.005651	0.011572	0.001687	0.000897	0.002406	0.003131	0.055420	0.035081
4	0.000573	0.003182	0.008287	0.008159	0.003570	0.008109	0.001463	0.000905	0.002680	0.003661	0.040588	0.032122
5	0.000349	0.002275	0.006573	0.006242	0.002566	0.006273	0.001266	0.000846	0.002733	0.003903	0.033025	0.029412
6	0.000239	0.001796	0.005581	0.005188	0.002045	0.005216	0.001107	0.000772	0.002656	0.003918	0.028518	0.026931
7	0.000183	0.001515	0.004914	0.004513	0.001735	0.004526	0.000982	0.000701	0.002516	0.003790	0.025376	0.024659
8	0.000151	0.001329	0.004407	0.004021	0.001524	0.004022	0.000882	0.000638	0.002350	0.003586	0.022911	0.022579
9	0.000131	0.001190	0.003992	0.003629	0.001364	0.003622	0.000797	0.000582	0.002177	0.003348	0.020832	0.020674
10	0.000117	0.001077	0.003634	0.003296	0.001233	0.003285	0.000725	0.000531	0.002007	0.003101	0.019007	0.018930
11	0.000106	0.000981	0.003317	0.003005	0.001121	0.002991	0.000661	0.000486	0.001845	0.002859	0.017371	0.017333
12	0.000096	0.000895	0.003032	0.002744	0.001023	0.002731	0.000603	0.000444	0.001693	0.002628	0.015890	0.015871
13	0.000088	0.000819	0.002773	0.002509	0.000934	0.002496	0.000551	0.000406	0.001553	0.002412	0.014542	0.014532
14	0.000080	0.000749	0.002538	0.002296	0.000854	0.002283	0.000504	0.000372	0.001423	0.002212	0.013311	0.013306
15	0.000073	0.000685	0.002323	0.002101	0.000782	0.002089	0.000462	0.000341	0.001303	0.002027	0.012186	0.012183
16	0.000067	0.000627	0.002127	0.001923	0.000715	0.001912	0.000423	0.000312	0.001194	0.001857	0.011157	0.011156
17	0.000061	0.000574	0.001947	0.001761	0.000655	0.001750	0.000387	0.000285	0.001093	0.001701	0.010215	0.010214
18	0.000056	0.000526	0.001783	0.001612	0.000600	0.001602	0.000354	0.000261	0.001001	0.001558	0.009353	0.009353
19	0.000051	0.000482	0.001632	0.001476	0.000549	0.001467	0.000324	0.000239	0.000917	0.001426	0.008564	0.008564
20	0.000047	0.000441	0.001494	0.001352	0.000503	0.001343	0.000297	0.000219	0.000839	0.001306	0.007841	0.007841
21	0.000043	0.000404	0.001368	0.001237	0.000460	0.001230	0.000272	0.000201	0.000769	0.001196	0.007180	0.007180
22	0.000039	0.000370	0.001253	0.001133	0.000421	0.001126	0.000249	0.000184	0.000704	0.001095	0.006574	0.006574
23	0.000036	0.000338	0.001147	0.001037	0.000386	0.001031	0.000228	0.000168	0.000644	0.001003	0.006020	0.006020
24	0.000033	0.000310	0.001050	0.000950	0.000353	0.000944	0.000209	0.000154	0.000590	0.000918	0.005512	0.005512
25	0.000030	0.000284	0.000962	0.000870	0.000323	0.000865	0.000191	0.000141	0.000540	0.000841	0.005047	0.005047
30	0.000019	0.000183	0.000619	0.000560	0.000208	0.000556	0.000123	0.000091	0.000348	0.000541	0.003248	0.003248
40	0.000008	0.000076	0.000256	0.000232	0.000086	0.000230	0.000051	0.000038	0.000144	0.000224	0.001345	0.001345
50	0.000003	0.000031	0.000106	0.000096	0.000036	0.000095	0.000021	0.000016	0.000060	0.000093	0.000557	0.000557
100	0.000000	0.000000	0.000001	0.000001	0.000000	0.000001	0.000000	0.000000	0.000001	0.000001	0.000007	0.000007
:	:	:	:	:	:	:	:	:	:	:	:	:
P (phase = j) = $\bar{\pi}_j$												
	0.010175763	0.101757632	0.305272895	0.228954672	0.062442183	0.145698427	0.020814061	0.012808653	0.044830285	0.067245428	1	0.999998

Now the system-length distribution at postdeparture epoch is obtained by

$$y_l^+ = \begin{cases} \frac{y_0(-D_0)}{\lambda}, & l = 0; \\ \frac{1}{\lambda} \left(y_l(-D_0) - \sum_{i=1}^{l-1} y_i D \right), & l \geq 1. \end{cases} \quad (20)$$

Remark 6. In Nishimura's method, we have to compute all the eigenvalues z_j and the corresponding eigenvectors $\mathbf{v}_j$ of $z\mathbf{I} - \mathbf{A}(z)$. Also we need to evaluate the eigenvalues and eigenvectors of $\mathbf{D}(z)$ and $\mathbf{F}(z)$, which is not easy even for a moderate value of m , as mentioned earlier. CGG method has an impeccable advantage over this method as it does not involve determination of any eigenvalues and eigenvectors. In this method, one needs to find the roots of the c.e. in Ω and Ω^* . Nowadays, with improved computational aids, the determination of roots of a large degree polynomial, however, close and in any interval is not a big deal. Also, by CGG method, we can extract the vector $\mathbf{q}$ from any one component of the vector generating function $\mathbf{Y}(z)$ which is a notable side of this approach.

Remark 7. Nishimura [13] shows that for $BDMMAP/G/1$ queue the roots of the characteristic equation $|z\mathbf{I} - \mathbf{A}(z)|$ inside the unit circle are real and distinct, so we need not to bother about the multiple root case inside Ω in CGG method. This is an added advantage of using CGG method.

Remark 8. In order to apply the CGG method only we have to assume that the service time distribution has rational LST. It is well known that a wide class of distribution produces rational LST. Also, for the distribution which does not possess rational LST, we can use some approximation method which will replace it with a rational function; for example, for deterministic distribution (which has transcendental LST), we can use Padé approximation to rationalize its LST (for references see [17–20]). Also to justify our claim for rational approximation, we present the boundary vector in Table 3, for the present model in the case of deterministic service time distribution and match it to that obtained by Nishimura's method.

4. Numerical Results

An extensive numerical effort has been put forth and a considerable number of examples have been studied, in order to validate the objective of this paper. However, in this section, we present few of them in the form of self-explanatory tables. For examples 1 and 2, considered here, we took $\lambda = 3.560994$, service time distribution as phase type distribution with parameters $n := 2, T := \begin{pmatrix} -41.2 & 35 \\ 25 & -52.5 \end{pmatrix}, \alpha := \begin{pmatrix} 1 & 0 \end{pmatrix}$ with mean 0.067934. In Table 1, we obtain the distribution of stationary probability at arbitrary epoch using CGG method for $m = 7$. The other input parameters are displayed in the table. We present the stationary probability vector $\boldsymbol{\pi}, z_j$'s in the table. It is shown in the table that the numerical value of $\mathbf{q}\mathbf{e}$ matches $1 - \rho$. Here, we would like to mention that we have compared the values of $\mathbf{q}$ obtained by both the Nishimura and CGG

method and which are found to be the same. As we were unable to invert the pgf $Y^+(z)$ even for the low value of m by Nishimura's method due to its complicated structure presented by him, we restrict up to the comparison of $\mathbf{q}$ only. The last row and column of Table 1 represent the marginal distributions of the number of the customers in the system and the distribution of the phases, respectively. In Table 2, we present the distribution of stationary probability at service completion epoch using CGG method. The input parameters in this example are the same as in Table 1. The other things are analogous to that of Table 1.

In Table 3, we present the boundary vector $\mathbf{q}$ for the present model in case of deterministic service time distribution and further it is observed that it matches up to desired degree of accuracy with that obtained by Nishimura's method. Here we use *Padé approximation* with the parameters [5, 5] to rationalize the transcendental LST which results from the deterministic distribution. The other particulars about the parameters taken for the example are presented in the table. In Table 4, we present the higher order random epoch probabilities for the deterministic service time distribution for the input parameters the same as in Table 3. Here the c.e. $S(z) = 0$ admits 510 roots of which 10 roots lie inside Ω and the others lie in Ω^* . For this very example we would like to emphasize on the fact that it is observed that of the 500 roots in Ω^* several roots are repeated and the number of distinct roots is 100. It is often commented in queueing literature that it is quite difficult to handle the multiple roots although it may be admitted from our observation that it is not always the case. Here we find the probabilities from the inversion of $Y(z)$ using the partial fraction technique with simple algebraic manipulation in case of roots with multiplicity more than 1. Also to check the accuracy of the probabilities computed here, we find the tail probabilities using the root with smallest absolute value (here it is $z = 1.09213352$) of the c.e. that lie in Ω^* and these values agree up to satisfaction after $l = 18$ (for details see the Table 4).

The purpose of the examples presented here is to compare the accuracy of the two methods (up to the level of $\mathbf{q}$) and provide some numerical demonstration of the present queueing model by obtaining steady state distributions both at arbitrary and postdeparture epoch by CGG method for various input parameters. Due to the lack of space, we have not presented the numerical results for higher values of m . However, these can be obtained from the author, if needed.

5. Conclusion and Future Scope

This paper presents a parallel study of the Nishimura and CGG method, on the basis of the numerical implementation of the $BDMMAP/G/1$ queue. The main objective of this paper is to inform the readers about the two methods from the computation point of view. A number of varied examples throughout the investigation bring out the fact that implementation of Nishimura's method is very very difficult if not impossible, even for the low order input parameters, whereas the CGG method seems to tackle the present queueing model in a much better way.

Further, the CGG method may be applied to analyze a wide variety of queueing models with practical importance, for example, TSMAP/G/1 queue (which is mainly used to model tree structured LAN traces); for details, see [21]. Also, it would be interesting if any recursive algorithm or some analytic computation scheme for the computation of the roots of the c.e. can be developed for the present queueing model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of the paper.

Acknowledgments

The authors thank the anonymous referees for their valuable comments and suggestions that helped to improve the presentation of the paper in the current form. The second author acknowledges the Department of Science and Technology, Govt. of India, for the financial support under the Project Grant SR/S4/MS: 789/12.

References

- [1] M. F. Neuts, "A versatile Markovian point process," *Journal of Applied Probability*, vol. 16, no. 4, pp. 764–779, 1979.
- [2] D. M. Lucantoni, "The BMAP/G/1 queue: a tutorial," in *Performance Evaluation of Computer and Communication Systems*, vol. 729 of *Lecture Notes in Computer Science*, pp. 330–358, Springer, Berlin, Germany, 1993.
- [3] S. R. Chakravathy, "The batch Markovian arrival process: a review and future work," in *Advances in Probability Theory and Stochastic Processes*, vol. 1, pp. 21–49, 2001.
- [4] V. Ramaswami, "A stable recursion for the steady state vector in Markov chains of M/G/1 type," *Stochastic Models*, vol. 4, no. 1, pp. 183–188, 1988.
- [5] V. Ramaswami, "Nonlinear matrix equations in applied probability—solution techniques and open problems," *SIAM Review*, vol. 30, no. 2, pp. 256–263, 1988.
- [6] D. Bini and B. Meini, "On the solution of a nonlinear matrix equation arising in queueing problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 4, pp. 906–926, 1996.
- [7] M. F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications*, vol. 5 of *Probability: Pure and Applied*, Marcel Dekker, New York, NY, USA, 1989.
- [8] D. A. Bini, B. Meini, S. Steffé, and B. van Houdt, "Structured Markov chains solver: software tools," in *Proceedings of the 2006 Workshop on Tools for Solving Structured Markov Chains*, p. 14, ACM, 2006.
- [9] D. A. Bini and B. Meini, "Improved cyclic reduction for solving queueing problems," *Numerical Algorithms*, vol. 15, no. 1, pp. 57–74, 1997.
- [10] H. W. Lee, J. M. Moon, J. K. Park, and B. K. Kim, "A spectral approach to compute the mean performance measures of the queue with low-order BMAP input," *Journal of Applied Mathematics and Stochastic Analysis*, vol. 16, no. 4, pp. 349–360, 2003.
- [11] H. R. Gail, S. L. Hantler, and B. A. Taylor, "Spectral analysis of M/G/1 and G/M/1 type Markov chains," *Advances in Applied Probability*, vol. 28, no. 1, pp. 114–165, 1996.
- [12] H. W. Lee, J. M. Moon, B. K. Kim, J. G. Park, and S. W. Lee, "A simple eigenvalue method for low-order D-BMAP/G/1 queues," *Applied Mathematical Modelling*, vol. 29, no. 3, pp. 277–288, 2005.
- [13] S. Nishimura, "A MAP/G/1 queue with an underlying birth-death process," *Stochastic Models*, vol. 19, no. 4, pp. 425–447, 2003.
- [14] D. M. Lucantoni, "New results on the single server queue with a batch Markovian arrival process," *Communications in Statistics: Stochastic Models*, vol. 7, no. 1, pp. 1–46, 1991.
- [15] D. A. Bini, "Numerical computation of polynomial zeros by means of Aberth's method," *Numerical Algorithms*, vol. 13, no. 2, pp. 179–200, 1996.
- [16] M. L. Chaudhry, G. Singh, and U. C. Gupta, "A simple and complete computational analysis of MAP/R/1 queue using roots," *Methodology and Computing in Applied Probability*, vol. 15, no. 3, pp. 563–582, 2013.
- [17] B. Fisher and M. Chaudhry, "Computing the distribution for the number of renewals with bulk arrivals," *INFORMS Journal on Computing*, vol. 26, no. 4, pp. 885–892, 2014.
- [18] N. Akar, "Solving the ME/ME/1 Queue with State-space Methods and the Matrix Sign Function," *Queueing Systems*, vol. 63, no. 2, pp. 131–145, 2006.
- [19] N. Akar and E. Arikan, "A numerically efficient method for the MAP/D/1/K queue via rational approximations," *Queueing Systems*, vol. 22, no. 1–2, pp. 97–120, 1996.
- [20] S. Winitzki, "Uniform approximations for transcendental functions," in *Computational Science and Its Applications—ICCSA*, vol. 2667 of *Lecture Notes in Computer Science*, pp. 780–789, Springer, Berlin, Germany, 2003.
- [21] S. Nishimura and M. Shinno, "Modeling burst traffic using a MAP with a tree structure," *Journal of the Operations Research Society of Japan*, vol. 47, no. 3, pp. 129–144, 2004.