Queueing Models and
Service Management
Vol. 4, No. 1, page 1-30, 2021

QMSM

Performance Analysis of a Versatile Correlated Batch-arrival and Batch-service Queue

B. Bank and S. K. Samanta*

Department of Mathematics
National Institute of Technology Raipur
Raipur-492010, Chhattisgarh, India
(Received August 2019 ; accepted September 2020)

Abstract: In this paper, we investigate the *BMAP/BMSP/*1 queueing system. Analysis of this queueing system is carried out using the zeros of the associated characteristic function of the vector-generating function of system-length distribution at random epoch. A comparative study is also carried out to compare the roots method with that of the matrix-geometric method. We then obtain the steady-state system-length distributions at pre-arrival and post-departure epochs. The sojourn-time distribution of an arbitrary customer in an arriving batch is also derived. Various performance measures such as mean system-length and mean sojourn-time are determined. We generate adequate numerical results based on the diversified inputs but only a few of them are appended in the forms of tables and graphs.

Keywords: Batch Markovian arrival process (BMAP), batch Markovian service process (BMSP), matrix-geometric method, queue, roots, sojourn-time distribution.

1. Introduction

Bulk queues (sometimes called batch queues), where jobs arrive in and/or are served in batches of random size, have received a special interest over the last few decades due to their utility in many practical real-life situations. Queueing models with correlated batch arrival as well as batch service processes play a crucial role in recent trends of the queueing behavior. The versatile Markovian point process pioneered by Neuts [23], and conveniently represented as the batch Markovian arrival process (BMAP) by Lucantoni [21] have found wide range of applications in several practical areas. The BMAP is the generalization of batch Poisson process and includes many well-known arrival processes such as Markov-modulated Poisson process, Markovian arrival process (MAP) and batch PH-renewal process, see, for example, Lucantoni [22]. The use of BMAP as an arrival process in queueing modelling readily leads to the so called matrix-analytic formalism, where scalar quantities are replaced by matrices. The BMAP is a powerful arrival process which captures dependent and non-exponentially distributed interarrival times, and correlated batch sizes. From an analytical viewpoint, the BMAP is a tractable arrival process and it is a convenient tool in many real-life stochastic modeling contexts. Keeping an eye on this prospect, the BMAP beautifully represents the queueing characteristics both in analytical and application aspects. The concept of BMAP has gained widespread use in queueing modelling of communication

*Corresponding author
Email : sksamanta.maths@nitrr.ac.in

systems such as cellular networks, web browsing, traffic modeling of IP networks (Kim *et al.* [17] and Klemm *et al.* [18]), hybrid highspeed communication systems based on laser and radio technologies (Vishnevskii *et al.* [32]), wireless networks with linear topology (Vishnevsky *et al.* [33]), production and manufacturing (Gold and Tran-Gia [10], and Shanthikumar *et al.* [30]) and other application areas. For more information about applications of BMAP, see Vishnevskii and Dudin [31], Klemm *et al.* [19], Buchholz and Kriege [7], Liu *et al.* [20], and Heyman and Lucantoni [13]. Qualitatively, the consideration of the BMAP for modelling the arrival input greatly enhances the versatility of the queueing models. This qualitative behavior of these queueing models continues to attract the attention of researchers and practitioners. The batch Markovian service process (BMSP) has the same features as that of BMAP wherein arrivals are replaced with service completions. Hence, BMSP has similar impact on analytical results and application areas for service process in queueing system. For more details about the BMAP, its history, properties, special cases and related research, see Lucantoni [22] and survey paper by Chakravarthy [8].

Many authors have analyzed several queueing models with various types of arrival as well as service processes and such results are available in the literature. However, very few authors dealt with correlated arrival and service processes. Abate *et al.* [1] and Alfa *et al.* [3] discussed the stationary distributions of $MAP/MSP/1$ queue based on the perturbation theory. Ozawa [26] derived the stationary sojourn-time distribution and asymptotic properties of the $MAP/MSP/1$ queue through its matrix exponential form. Horváth *et al.* [15] analyzed the queueing networks of $MAP/MSP/1$ queue by proposing the decomposition based approximate numerical analysis. Zhang *et al.* [36] proposed a family of finite approximations for the departure process of a $BMAP/MSP/1$ queue and the departure process approximations are derived via an exact aggregate solution technique (called ETAQA). Samanta *et al.* [28] analyzed the $BMAP/MSP/1$ queueing model based on roots of the associated characteristics equation of the probability vector-generating function of system-length distribution at random epoch. Wang *et al.* [35] applied a matrix-analytical approach to investigate the finite-buffer $DBMAP/DMSP/1/K$ queue in discrete-time and examined the bursty nature of packet loss pattern in wireless local communications. From the above literature overview, we may see that the existing research related to the bursty and correlated nature of arrival as well as service processes has mainly focused on the single service. However, a few works have been done on the corresponding batch service queue. Banik [4] discussed the $BMAP/MSP^{(a,b)}/1$ queueing system based on the use of matrix-analytic method developed by Neuts [25], where customers are served in batches of maximum size 'b' with a minimum batch size 'a' in which the service rate for all service batches remains the same. To the best of authors' knowledge, very few results on BMSP are available so far in the queueing literature, where the service rate depends on service batch size. Using a matrix-analytical approach, Wang *et al.* [34] analyzed the finite-buffer $DBMAP/DBMSP/1/K$ queue in discrete-time to evaluate the long term packet loss probabilities over wireless networks. Sandhya *et al.* [29] studied an infinite-buffer $BMAP/BMSP/1$ queue by partitioning the infinitesimal generator with blocks having groups of customers of maximum size of arrival and service batch sizes. Based on the use of matrix-geometric method (MGM) pioneered by Neuts [24], they determined the stationary probability of the number of customers waiting for service and other performance measures.

The above literature survey motivates us to analyze the $BMAP/BMSP/1$ queueing system in which customers are served in batches with different service rate for different batch size. For analytical and computational purpose, we present the queueing model with a suitable matrix structure form which makes it easy to study in a unified and algorithmically tractable manner. We assume that the random batch sizes of arrival and service processes are restricted to be finite. First, we obtain the system-length distribution at random epoch based on the zeros of the characteristic polynomial of the probability vector-generating function. A comprehensive analysis of the system-length distribution at random epoch is also carried out using the matrix-geometric method based on reblocking the underlying Markov chain in QBD form. See Horváth [14], He [12, p. 260], Benzi *et al.* [6, p. 77], and Alexander [2, p.74] for details. We then derive the steady-state system-length distributions at pre-arrival and post-departure epochs. In this queueing system, the determination of the sojourn-time distribution of an arbitrary customer in an arriving batch is challenging because customers are arrived in batches according to BMAP as well as they are also served in batches according to BMSP. However, we overcome the challenge successfully to obtain the sojourn-time distribution of an arbitrary customer in an arriving batch. The main advantage of our work is to derive the sojourn-time distribution of an arbitrary customer in terms of time parameter directly without converting from the Laplace-Stieltjes transform (L.-S.T.). This is analytically simple and easy to compute. Further, we find the L.-S.T. of the sojourn-time distribution function to obtain the mean sojourn-time. Various performance measures such as the mean system-length and the mean sojourn-time are obtained. To justify our analytical results, we generate adequate outputs based on the diversified inputs but only a few of them are appended here in the forms of tables and graphs.

This paper is organized as follows. In Section 2, we give the description of the model. The steady-state system-length distributions at various time epochs and the sojourn-time distribution of an arbitrary customer in an arriving batch are analyzed in Section 3. Numerical results are presented in Section 4. Section 5 concludes the paper.

2. Model Description

We consider an infinite-buffer single-server $BMAP/BMSP/1$ queueing system, wherein customers arrive according to a batch Markovian arrival process (BMAP). The customers are served in batches in accordance with the first-come-first-served (FCFS) queueing discipline under a batch Markovian service process (BMSP). The arrival process BMAP is characterized by the $m_a \times m_a$ rate matrices $\mathbf{D}_k,\ k \geq 0$, where $\mathbf{D}_k$ corresponds to an arrival of batch size k if $k \geq 1$, and without an arrival if $k = 0$. The m_a-state of the BMAP is usually referred to as the phase (state) of the underlying Markov chain (UMC) corresponding to the BMAP. The (i, j)-th element $[D_k]_{ij}$ of $\mathbf{D}_k$ denotes the phase transition rate of the UMC corresponding to the BMAP from state i to j with a batch arrival of size k. The matrix $\mathbf{D}_0$ has non-negative off-diagonal and negative diagonal elements, and the matrix $\mathbf{D}_k,\ k \geq 1$, has non-negative elements. The diagonal element $[D_0]_{ii}$ of $\mathbf{D}_0$ represents the mean rate of exponential sojourn time in state i, $1 \leq i \leq m_a$. From practical point of view, the size of arriving batch is to be of finite support. Therefore, we assume that the random batch size of arrival process to be finite with maximum batch size N_0. This gives $\mathbf{D}_k = \mathbf{0}$, for $k \geq N_0+1$. Since $\mathbf{D} = \sum_{k=0}^{N_0} \mathbf{D}_k$ is an infinitesimal generator of the underlying Markov chain corresponding to the BMAP, there exists a stationary probability vector $\overline{\boldsymbol{\pi}}_a$ such that $\overline{\boldsymbol{\pi}}_a \mathbf{D} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_a \mathbf{e} = 1$, where $\mathbf{e}$ denotes a column vector with an appropriate order whose all elements are 1. The average arrival rate λ^* of the stationary BMAP is given by $\lambda^* = \overline{\boldsymbol{\pi}}_a \sum_{k=1}^{N_0} k\mathbf{D}_k\mathbf{e}$.

Similarly, the service process BMSP is characterized by the $m_s \times m_s$ rate matrices $\mathbf{L}_k,\ k \geq 0$, where $\mathbf{L}_k$ corresponds to a service of batch size k if $k \geq 1$ and without a service if $k = 0$. The m_s-state of the BMSP is usually referred to as the phase (state) of the UMC corresponding to the BMSP. The (i, j)-th element $[L_k]_{ij}$ of $\mathbf{L}_k$ denotes the phase transition rate of the UMC corresponding to the BMSP from state i to j with a batch service of size k. The matrix $\mathbf{L}_0$ has non-negative off-diagonal and negative diagonal elements, and the matrix $\mathbf{L}_k,\ k \geq 1$, has non-negative elements. The diagonal element $[L_0]_{ii}$ of $\mathbf{L}_0$ represents the mean rate of exponential sojourn time in state i. Again, from practical point of view, the size of servicing batch is to be of finite support. Therefore, we assume that the random batch size of service process to be finite with maximum batch size M_0. This gives $\mathbf{L}_k = \mathbf{0}$, for $k \geq M_0+1$. Since $\mathbf{L} = \sum_{k=0}^{M_0} \mathbf{L}_k$ is an infinitesimal generator of the underlying Markov chain corresponding to the BMSP, there exists a stationary probability vector $\overline{\boldsymbol{\pi}}_s$ such that $\overline{\boldsymbol{\pi}}_s \mathbf{L} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_s \mathbf{e} = 1$. The average service rate μ^* of the stationary BMSP is given by $\mu^* = \overline{\boldsymbol{\pi}}_s \sum_{k=1}^{M_0} k\mathbf{L}_k\mathbf{e}$. The traffic intensity is given by $\rho = \frac{\lambda^*}{\mu^*} < 1$.

3. Analysis of the Model

In this section, we carry out the analysis of the system-length distributions at random, pre-arrival, and post-departure epochs as well as the sojourn-time distribution of an arbitrary customer in an arriving batch.

3.1. System-length distribution at random epoch

We first consider the steady-state system-length distribution at random epoch. For this purpose, we define the state of the system at time t by $Y(t) = (N(t), I(t), J(t))$, where $N(t)$ denotes the number of customers in the system, $I(t)$ the phase of the BMAP and $J(t)$ the phase of the BMSP at time t. Then $\{Y(t)\}_{t\geq 0}$ is a continuous-time Markov chain on the state space $\{(n, i, j) : n \geq 0, 1 \leq i \leq m_a, 1 \leq j \leq m_s\}$. The Toeplitz type block-structured infinitesimal generator $\mathbf{Q}$ for the $BMAP/BMSP/1$ queue has the following structured:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & \mathbf{A}_5 \cdots \\ \widehat{\mathbf{S}}_1 & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 \cdots \\ \widehat{\mathbf{S}}_2 & \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \cdots \\ \widehat{\mathbf{S}}_3 & \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \cdots \\ \widehat{\mathbf{S}}_4 & \mathbf{S}_3 & \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 \cdots \\ \widehat{\mathbf{S}}_5 & \mathbf{S}_4 & \mathbf{S}_3 & \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \quad \ddots \end{pmatrix}, \tag{1}$$

where the matrix $\widehat{\mathbf{S}}_n, n \geq 1$, of order $m_a m_s \times m_a m_s$ decreases the level of the chain by n and it reaches to the level zero of the chain, while the matrix $\mathbf{B}_0$ of order $m_a m_s \times m_a m_s$ remains at the level zero. The matrix $\mathbf{S}_n, n \geq 1$, of order $m_a m_s \times m_a m_s$ decreases the level of the chain by n and it reaches to the respective level of the chain. The matrix $\mathbf{A}_n, n \geq 1$, of order $m_a m_s \times m_a m_s$ increases the level of the chain by n, while the matrix $\mathbf{A}_0$ of order $m_a m_s \times m_a m_s$ remains at the same level. We assume that the service process runs during idle periods of the system without generating any real service completion. Therefore, the block matrices of the generator (1) can be expressed using the Kronecker product $\otimes$ operation as

$$\begin{aligned} \mathbf{B}_0 &= \mathbf{D}_0 \otimes \mathbf{I}_{m_s} + \mathbf{I}_{m_a} \otimes \widehat{\mathbf{L}}_0, \\ \widehat{\mathbf{S}}_n &= \mathbf{I}_{m_a} \otimes \widehat{\mathbf{L}}_n, \quad n \geq 1, \\ \mathbf{S}_n &= \mathbf{I}_{m_a} \otimes \mathbf{L}_n, \quad n \geq 1, \\ \mathbf{A}_0 &= \mathbf{D}_0 \otimes \mathbf{I}_{m_s} + \mathbf{I}_{m_a} \otimes \mathbf{L}_0, \\ \mathbf{A}_n &= \mathbf{D}_n \otimes \mathbf{I}_{m_s}, \quad n \geq 1, \end{aligned}$$

where $\mathbf{I}_r$ is the identity matrix of order r, and $\widehat{\mathbf{L}}_n = \sum_{k=n}^{M_0} \mathbf{L}_k$, $n \geq 0$.

Let $\boldsymbol{\pi}(n) = [\pi_{11}(n), \ldots, \pi_{1m_s}(n), \ldots, \pi_{ij}(n), \ldots, \pi_{m_a 1}(n), \ldots, \pi_{m_a m_s}(n)]$, $n \geq 0$, denote the row vector according to the block structure of the generator $\mathbf{Q}$, where $\pi_{ij}(n)$ represents the steady-state probability that there are n customers in the system with the arrival process being in phase i $(1 \leq i \leq m_a)$ and the service process being in phase j $(1 \leq j \leq m_s)$. Define the stationary probability vector $\mathbf{\Pi}$ of $\mathbf{\Pi Q} = \mathbf{0}$ with $\mathbf{\Pi e} = 1$ in the partitioned form as $\mathbf{\Pi} = [\boldsymbol{\pi}(0), \boldsymbol{\pi}(1), \boldsymbol{\pi}(2), \boldsymbol{\pi}(3), \ldots]$. Then $\mathbf{\Pi Q} = \mathbf{0}$ can be written explicitly as

$$\boldsymbol{\pi}(0)\mathbf{B}_0 + \sum_{n=1}^{M_0} \boldsymbol{\pi}(n)\widehat{\mathbf{S}}_n = \mathbf{0}, \tag{2}$$

$$\boldsymbol{\pi}(0)\mathbf{A}_n + \sum_{k=1}^{n} \boldsymbol{\pi}(k)\mathbf{A}_{n-k} + \sum_{k=1}^{M_0} \boldsymbol{\pi}(n+k)\mathbf{S}_k = \mathbf{0}, \quad n \geq 1. \tag{3}$$

Multiplying (2) by z^0 and (3) by z^n, using $\boldsymbol{\pi}^*(z) = \sum_{n=0}^{\infty} \boldsymbol{\pi}(n) z^n$, after simplification, we obtain

$$\boldsymbol{\pi}^*(z) = \left[\boldsymbol{\pi}(0)\left(\mathbf{I}_{m_a} \otimes \left(\mathbf{L}(z^{-1}) - \widehat{\mathbf{L}}_0\right)\right) + \sum_{n=1}^{M_0-1} \boldsymbol{\pi}(n) \sum_{k=n+1}^{M_0} (\mathbf{I}_{m_a} \otimes \mathbf{L}_k)\left(z^{n-k} - 1\right)\right] \times \left(\frac{\text{adj}\,[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]}{\det\,[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]}\right), \quad (4)$$

where adj$[\cdot]$ and $\det[\cdot]$ are the adjoint matrix and the determinant of a square matrix, respectively.

To obtain the system-length distribution $\boldsymbol{\pi}(n), n \geq 0$, we use the zeros of the characteristic polynomial of the vector-generating function $\boldsymbol{\pi}^*(z)$. Our primary objective is to correctly determine the unknown vectors $\boldsymbol{\pi}(n), 0 \leq n \leq M_0-1$, that occur in (4) above. For this, the idea of the zeros of $\det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]$ in the unit disk is required. We know that if $\rho < 1$ then $\det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})] = 0$ has exactly $(mM_0 - 1)$ roots inside of $|z| = 1$, one root at $z = 1$ and other mN_0 roots outside of $|z| = 1$ (including multiplicity), where $m = m_a \cdot m_s$. In this connection, the interested reader is referred to Gail *et al.* [9] or Samanta *et al.* [28]. We denote the roots whose absolute value is less than one as $\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_{mM_0-1}$ and the roots whose absolute value is greater than one as $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{mN_0}$. We assume that all roots are distinct. Since, each component of $\boldsymbol{\pi}^*(z)$ is convergent in $|z| \leq 1$, therefore the zeros of $\det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]$ whose absolute value is less or equal to one must be the zeros of the numerator of each component of $\boldsymbol{\pi}^*(z)$. This shows that we can determine the unknown vectors $\boldsymbol{\pi}(n), 0 \leq n \leq M_0 - 1$, by considering any one component of $\boldsymbol{\pi}^*(z)$. Therefore, we rewrite the right-hand side of $\boldsymbol{\pi}^*(z)$ in (4) as

$$\boldsymbol{\pi}^*(z) = \left[\frac{F_{11}(z)}{G(z)}, \ldots, \frac{F_{ij}(z)}{G(z)}, \ldots, \frac{F_{m_a m_s}(z)}{G(z)}\right], \quad (5)$$

where $G(z) = \det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]$ and $F_{ij}(z)$ is the ij-th component of the vector

$$\left[\boldsymbol{\pi}(0)\left(\mathbf{I}_{m_a} \otimes \left(\mathbf{L}(z^{-1}) - \widehat{\mathbf{L}}_0\right)\right) + \sum_{n=1}^{M_0-1} \boldsymbol{\pi}(n) \sum_{k=n+1}^{M_0} (\mathbf{I}_{m_a} \otimes \mathbf{L}_k)\left(z^{n-k} - 1\right)\right] \text{adj}\left[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})\right].$$

Now, since each component $\pi^*_{ij}(z)$, $1 \leq i \leq m_a, 1 \leq j \leq m_s$, of $\boldsymbol{\pi}^*(z)$ is convergent in $|z| \leq 1$ and $\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_{mM_0-1}$, are the zeros of $G(z)$, we have

$$F_{ij}(\gamma_k) = 0, \quad 1 \leq i \leq m_a,\ 1 \leq j \leq m_s,\ \ k = 1, 2, \ldots, mM_0 - 1, \quad (6)$$

and using the normalization condition $\boldsymbol{\pi}^*(1)\mathbf{e} = 1$, we have

$$\lim_{z \to 1} \frac{\sum_{i=1}^{m_a} \sum_{j=1}^{m_s} F_{ij}(z)}{G(z)} = \frac{\sum_{i=1}^{m_a} \sum_{j=1}^{m_s} F'_{ij}(1)}{G'(1)} = 1, \quad (7)$$

where $f^{'}(\xi)$ is the first order derivative of $f(z)$ at $z=\xi$.

Equations (6) and (7) give mM_0 linearly independent simultaneous equations in mM_0 unknowns, $\pi_{ij}(n)$'s $(0 \le n \le M_0-1, 1 \le i \le m_a, 1 \le j \le m_s)$. Solving these mM_0 equations, we determine the M_0 unknown vectors $\boldsymbol{\pi}(n)$, $0 \le n \le M_0-1$. Now, after substituting the value of $\boldsymbol{\pi}(n)$, $0 \le n \le M_0-1$, in (5) and letting $\overline{\boldsymbol{\pi}} = \boldsymbol{\pi}^*(1)$, we have

$$\overline{\boldsymbol{\pi}} = \left[\frac{F_{11}^{'}(1)}{G^{'}(1)}, \ldots, \frac{F_{ij}^{'}(1)}{G^{'}(1)}, \ldots, \frac{F_{m_a m_s}^{'}(1)}{G^{'}(1)}\right].$$

Having found $\boldsymbol{\pi}(n)$, $0 \le n \le M_0-1$ accurately, we now give our attention to calculate the remaining state probabilities $\boldsymbol{\pi}(n)$, $n \ge M_0$. After substituting the value of $\boldsymbol{\pi}(n)$, $0 \le n \le M_0-1$, in (4), the $\boldsymbol{\pi}^*(z)$ is a rational function with completely known polynomials both in the numerator and the denominator, where the degree of numerator is less than to the degree of denominator. To determine the remaining probability vectors $\boldsymbol{\pi}(n)$, $n \ge M_0$, we proceed to find the partial fractions of $\boldsymbol{\pi}^*(z)$ involving the zeros of $\det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]$ whose absolute value is greater than one. Note that the zeros of $\det[\mathbf{D}(z) \oplus \mathbf{L}(z^{-1})]$ whose absolute value is less or equal to one do not play any role in making partial fractions. Now, applying the partial fraction method on the ij-th component $\pi_{ij}^*(z)$ of $\boldsymbol{\pi}^*(z)$, we have

$$\pi_{ij}^*(z) = \sum_{k=1}^{mN_0} \frac{C_{k,ij}}{\alpha_k - z}, 1 \le i \le m_a, \;\; 1 \le j \le m_s, \tag{8}$$

where

$$C_{k,ij} = -\frac{F_{ij}(\alpha_k)}{G^{'}(\alpha_k)}, \quad 1 \le i \le m_a, \;\; 1 \le j \le m_s, \;\; k = 1, 2, \ldots, mN_0.$$

Now, collecting the coefficients of z^n from both the sides of (8), we have

$$\pi_{ij}(n) = \sum_{k=1}^{mN_0} \frac{C_{k,ij}}{\alpha_k^{n+1}}, \quad 1 \le i \le m_a, \;\; 1 \le j \le m_s, \;\; n \ge 0,$$

and hence

$$\boldsymbol{\pi}(n) = \left[\sum_{k=1}^{mN_0} \frac{C_{k,11}}{\alpha_k^{n+1}}, \ldots, \sum_{k=1}^{mN_0} \frac{C_{k,ij}}{\alpha_k^{n+1}}, \ldots, \sum_{k=1}^{mN_0} \frac{C_{k,m_a m_s}}{\alpha_k^{n+1}}\right], \quad n \ge 0. \tag{9}$$

The mean system-length can be obtained from (9) as $L_s = \sum_{n=0}^{\infty} n\boldsymbol{\pi}(n)\mathbf{e}$. From the Little's law, we also have mean sojourn time W_s as $W_s = \frac{L_s}{\lambda^*}$.

For a comparative study between the roots method and the matrix-geometric method, we now proceed to find out the random epoch probabilities $\boldsymbol{\pi}(n)$, $n \ge 0$, using the matrix-geometric method given in Neuts [24]. For this purpose, reblocking the Toeplitz type block-structure infinitesimal generator $\mathbf{Q}$ given above in the QBD form as

$$\mathbf{Q} = \begin{pmatrix} \mathbf{V} & \mathbf{Z} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots \\ \mathbf{X} & \mathbf{Y} & \mathbf{Z} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots \\ \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{Z} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{Z} & \mathbf{0} & \mathbf{0} & \mathbf{0}\cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{Z} & \mathbf{0} & \mathbf{0}\cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots\ddots \end{pmatrix}, \tag{10}$$

where the block matrices $\mathbf{V}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ each of order $\kappa = m_a m_s(\phi + 1)$ with $\phi = max(N_0, M_0)$ are given by

$$\mathbf{Z} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_\phi & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{\phi-1} & \mathbf{A}_\phi & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}_3 & \mathbf{A}_4 & \mathbf{A}_5 & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{A}_3 & \mathbf{A}_4 & \cdots & \cdots & \mathbf{A}_\phi & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \cdots & \cdots & \mathbf{A}_{\phi-1} & \mathbf{A}_\phi & \mathbf{0} \end{pmatrix},$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \cdots & \mathbf{A}_{\phi-2} & \mathbf{A}_{\phi-1} & \mathbf{A}_\phi \\ \widehat{\mathbf{S}}_1 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{\phi-3} & \mathbf{A}_{\phi-2} & \mathbf{A}_{\phi-1} \\ \widehat{\mathbf{S}}_2 & \mathbf{S}_1 & \mathbf{A}_0 & \cdots & \cdots & \mathbf{A}_{\phi-4} & \mathbf{A}_{\phi-3} & \mathbf{A}_{\phi-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \widehat{\mathbf{S}}_{\phi-2} & \mathbf{S}_{\phi-3} & \mathbf{S}_{\phi-4} & \cdots & \cdots & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \\ \widehat{\mathbf{S}}_{\phi-1} & \mathbf{S}_{\phi-2} & \mathbf{S}_{\phi-3} & \cdots & \cdots & \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 \\ \widehat{\mathbf{S}}_\phi & \mathbf{S}_{\phi-1} & \mathbf{S}_{\phi-2} & \cdots & \cdots & \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 \end{pmatrix},$$

$$\mathbf{X} = \begin{pmatrix} \mathbf{0} & \mathbf{S}_\phi & \mathbf{S}_{\phi-1} & \cdots & \cdots & \mathbf{S}_3 & \mathbf{S}_2 & \mathbf{S}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_\phi & \cdots & \cdots & \mathbf{S}_4 & \mathbf{S}_3 & \mathbf{S}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{S}_5 & \mathbf{S}_4 & \mathbf{S}_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{S}_\phi & \mathbf{S}_{\phi-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{S}_\phi \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

$$\mathbf{Y} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \cdots & \mathbf{A}_{\phi-2} & \mathbf{A}_{\phi-1} & \mathbf{A}_{\phi} \\ \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 & \cdots & \cdots & \mathbf{A}_{\phi-3} & \mathbf{A}_{\phi-2} & \mathbf{A}_{\phi-1} \\ \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 & \cdots & \cdots & \mathbf{A}_{\phi-4} & \mathbf{A}_{\phi-3} & \mathbf{A}_{\phi-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{S}_{\phi-2} & \mathbf{S}_{\phi-3} & \mathbf{S}_{\phi-4} & \cdots & \cdots & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{S}_{\phi-1} & \mathbf{S}_{\phi-2} & \mathbf{S}_{\phi-3} & \cdots & \cdots & \mathbf{S}_1 & \mathbf{A}_0 & \mathbf{A}_1 \\ \mathbf{S}_{\phi} & \mathbf{S}_{\phi-1} & \mathbf{S}_{\phi-2} & \cdots & \cdots & \mathbf{S}_2 & \mathbf{S}_1 & \mathbf{A}_0 \end{pmatrix}.$$

In order to efficiently solve the system of linear algebraic equations $\mathbf{\Pi Q} = \mathbf{0}$ with $\mathbf{\Pi e} = 1$, we reblock the vector $\mathbf{\Pi}$ as $\mathbf{\Pi} = [\mathbf{\Upsilon}(0), \mathbf{\Upsilon}(1), \mathbf{\Upsilon}(2), \ldots]$, where $\mathbf{\Upsilon}(n) = [\boldsymbol{\pi}(n(\phi+1)+0), \ldots, \boldsymbol{\pi}(n(\phi+1)+k), \ldots, \boldsymbol{\pi}(n(\phi+1)+\phi)]$, $n \geq 0$. Therefore, $\mathbf{\Pi Q} = \mathbf{0}$ can be written explicitly for QBD form as

$$\mathbf{\Upsilon}(0)\mathbf{V} + \mathbf{\Upsilon}(1)\mathbf{X} = \mathbf{0}, \tag{11}$$
$$\mathbf{\Upsilon}(n-1)\mathbf{Z} + \mathbf{\Upsilon}(n)\mathbf{Y} + \mathbf{\Upsilon}(n+1)\mathbf{X} = \mathbf{0}, \quad n \geq 1. \tag{12}$$

Applying the matrix-geometric method given in Neuts [24] on (11) and (12), we have

$$\mathbf{\Upsilon}(n) = \mathbf{\Upsilon}(0)\mathbf{R}^n, \quad n \geq 0, \quad \text{with} \quad \mathbf{R}^0 = \mathbf{I}_\kappa, \tag{13}$$

where the square matrix $\mathbf{R}$ of order κ is the minimal non-negative solution to the matrix-quadratic equation

$$\mathbf{Z} + \mathbf{RY} + \mathbf{R}^2\mathbf{X} = \mathbf{0},$$

and all eigenvalues of $\mathbf{R}$ lie inside the unit disk. This matrix $\mathbf{R}$ is evaluated numerically by a simple iterative scheme as follows:

$$\mathbf{R}[n+1] = -\mathbf{A} - \mathbf{R}^2[n]\mathbf{C}, \quad n \geq 0, \quad \text{with} \quad \mathbf{R}[0] = \mathbf{0},$$

where $\mathbf{A} = \mathbf{ZY}^{-1}$, $\mathbf{C} = \mathbf{XY}^{-1}$ and $\mathbf{R}[n]$ is the value of $\mathbf{R}$ at the n-th iteration.

To obtain $\mathbf{\Upsilon}(0)$, we solve the system of linear κ equations

$$\mathbf{\Upsilon}(0)(\mathbf{V} + \mathbf{RX}) = \mathbf{0},$$

from which one equation is replaced by the normalization condition $\mathbf{\Upsilon}(0)(\mathbf{I}_\kappa - \mathbf{R})^{-1}\mathbf{e} = 1$. Once we know $\mathbf{R}$ and $\mathbf{\Upsilon}(0)$, we can use the result (13) to get $\boldsymbol{\pi}(n)$, $n \geq 0$.

3.2. System-length distribution at pre-arrival epoch

Let $\boldsymbol{\pi}^-(n) = [\pi^-_{11}(n), \ldots, \pi^-_{1m_s}(n), \ldots, \pi^-_{ij}(n), \ldots, \pi^-_{m_a1}(n), \ldots, \pi^-_{m_am_s}(n)]$, $n \geq 0$, denote the row vector according to the block structure of the generator $\mathbf{Q}$, where $\pi^-_{ij}(n)$ represents the pre-arrival epoch probability that an arbitrary customer of an arriving batch finds n customers (including the customers in front of him in his batch) in the system with the arrival process being in phase i and the service process being in phase j. Therefore, we have

$$\boldsymbol{\pi}^{-}(n)=\sum_{r=0}^{n}\boldsymbol{\pi}(r)\Big(\mathbf{H}_{n+1-r}\otimes\mathbf{I}_{m_s}\Big),\quad n\geq 0,$$

where $\mathbf{H}_k=\frac{1}{\lambda^*}\sum_{n=k}^{N_0}\mathbf{D}_n, k\geq 1$, is a matrix of order $m_a\times m_a$ whose (i,j)-th element $[H_k]_{ij}$ represents the probability that the position of an arbitrary customer in an arriving batch is k with batch arrival phase changes from state i to j. For more details, the interested reader is referred to Samanta [27].

3.3. System-length distribution at post-departure epoch

Let $\boldsymbol{\pi}^{+}(n)=[\pi^{+}_{11}(n),\ldots,\pi^{+}_{1m_s}(n),\ldots,\pi^{+}_{ij}(n),\ldots,\pi^{+}_{m_a1}(n),\ldots,\pi^{+}_{m_am_s}(n)],\quad n\geq 0$, denote the row vector according to the block structure of the generator $\mathbf{Q}$, where $\pi^{+}_{ij}(n)$ represents the post-departure epoch probability that there are n customers in the system immediately after service completion of a batch with the arrival process being in phase i and the service process being in phase j. Hence, using the "rate-in and rate-out" argument; for more details, see Kim *et al.* [16], we have

$$\boldsymbol{\pi}^{+}(n)=\frac{\sum\limits_{k=n+1}^{n+M_0}\boldsymbol{\pi}(k)\Big(\mathbf{I}_{m_a}\otimes\mathbf{L}_{k-n}\Big)}{\sum\limits_{n=0}^{\infty}\sum\limits_{k=n+1}^{n+M_0}\boldsymbol{\pi}(k)\Big(\mathbf{I}_{m_a}\otimes\mathbf{L}_{k-n}\Big)\mathbf{e}},\quad n\geq 0.$$

3.4. Sojourn-time distribution

In this section, we obtain the sojourn-time distribution of an arbitrary customer in an arriving batch. The sojourn time means that the total time spent by a customer in the system (from its arrival until departure). For this, let $\mathcal{N}(x)$ denote the number of customers served in the time interval $(0,x]$ and $\mathcal{J}(x)$ be the phase (state) of the underlying Markov chain corresponding to the BMSP at time x with state-space $\{i:1\leq i\leq m_s\}$. Then $\{(\mathcal{N}(x),\mathcal{J}(x))\}$ is a two-dimensional Markov process of BMSP with state-space $\{(n,i):n\geq 0, 1\leq i\leq m_s\}$. Let $\{\mathbf{P}(n,x), n\geq 0, x\geq 0\}$ be an $m_s\times m_s$ matrix whose (i,j)-th element is the conditional probability defined as

$$P_{ij}(n,x)=Pr\{\mathcal{N}(x)=n,\mathcal{J}(x)=j|\mathcal{N}(0)=0,\mathcal{J}(0)=i\},\quad 1\leq i,j\leq m_s.$$

Using the property of BMSP and probability arguments, we have the following system of matrix differential-difference equation

$$\frac{d}{dx}\mathbf{P}(n,x)=\sum_{k=0}^{n}\mathbf{P}(k,x)\mathbf{L}_{n-k},\quad n\geq 0,\tag{14}$$

with $\mathbf{P}(0,0)=\mathbf{I}_{m_s}$ and $\mathbf{P}(n,0)=\mathbf{0}$, for $n\geq 1$.

Let $\mathbf{W}(x) = [W_{11}(x), \ldots, W_{1m_s}(x), \ldots, W_{ij}(x), \ldots, W_{m_a1}(x), \ldots, W_{m_am_s}(x)]$, $x \geq 0$, denote the row vector according to the block structure of the generator $\mathbf{Q}$, where $W_{ij}(x)$ represents the stationary joint probability that the sojourn time is at most a time x with the arrival phase being in i and the service phase being in j at time x, given that arbitrary customer arrived at time $x = 0$. Suppose that an arbitrary customer sees the system with n, $n \geq 0$, customers ahead of him upon arrival. If an arbitrary customer completes his service in the time interval $(x, x+dx]$, then k, $0 \leq k \leq n$, customers are served in the interval $(0, x]$ and a batch of size at least $(n+1-k)$ customers are served during dx time unit. Thus, the elementary probability vector $d\mathbf{W}(x)$ that an arbitrary customer completes his service in the time interval $(x, x+dx]$ is given by

$$d\mathbf{W}(x) = \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \mathbf{P}(k,x) \widehat{\mathbf{L}}_{n+1-k} dx \right) \quad x > 0.$$

Hence, the vector probability density function $\mathbf{w}(x) = \dfrac{d\mathbf{W}(x)}{dx}$ is given by

$$\mathbf{w}(x) = \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \mathbf{P}(k,x) \widehat{\mathbf{L}}_{n+1-k} \right) \quad x > 0. \tag{15}$$

The evaluation of the matrix $\mathbf{P}(n, x)$, $n \geq 0$, occurs in (15) can be carried out along the lines proposed by Lucantoni [21] for BMAP, which is also same for the BMSP. Hence, applying the uniformization argument to the matrices $\mathbf{P}(n, x)$ as presented by Lucantoni [21], we have

$$\mathbf{P}(n,x) = \sum_{k=0}^{\infty} e^{-\theta x} \frac{(\theta x)^k}{k!} \mathbf{U}_n^{(k)}, \quad n \geq 0, \;\; x \geq 0, \tag{16}$$

where $\theta = \max_i [-L_0]_{ii}$, $1 \leq i \leq m_s$ and $\mathbf{U}_n^{(k)}$ are given by

$$\mathbf{U}_n^{(k+1)} = \mathbf{U}_n^{(k)} + \theta^{-1} \sum_{r=0}^{n} \mathbf{U}_r^{(k)} \mathbf{L}_{n-r}, \quad n \geq 0, \;\; k \geq 0,$$

with $\mathbf{U}_0^{(0)} = \mathbf{I}_{m_s}$, $\mathbf{U}_n^{(0)} = \mathbf{0}$, $n \geq 1$.

Hence, the cumulative sojourn-time distribution of an arbitrary customer in an arriving batch is given by

$$\begin{aligned} \mathbf{W}(x) &= \int_0^x \mathbf{w}(t) dt \\ &= \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \int_0^x \mathbf{P}(k,t) dt \widehat{\mathbf{L}}_{n+1-k} \right), \quad x \geq 0. \end{aligned} \tag{17}$$

Using (16) in (17), we obtain

$$\mathbf{W}(x) = \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \sum_{r=0}^{\infty} J_r(x) \mathbf{U}_k^{(r)} \widehat{\mathbf{L}}_{n+1-k} \right), \quad x \geq 0,$$

where

$$J_r(x) = \int_0^x \frac{e^{-\theta t}(\theta t)^r}{r!} dt$$

can be calculated by the iterative scheme

$$\begin{aligned} J_0(x) &= \frac{1}{\theta}\left(1 - e^{-\theta x}\right), \\ J_r(x) &= J_{r-1}(x) - \frac{(\theta x)^r e^{-\theta x}}{r!\theta}, \quad r \geq 1. \end{aligned}$$

In order to determine the mean sojourn time of an arbitrary customer, let us define the L.-S.T. of $\mathbf{w}(x)$ and $\mathbf{P}(n, x)$ as

$$\begin{aligned} \widetilde{\mathbf{w}}(s) &= \int_0^{\infty} e^{-sx} \mathbf{w}(x) dx, \quad \mathrm{Re}(s) \geq 0, \\ \widetilde{\mathbf{P}}(n, s) &= \int_0^{\infty} e^{-sx} \mathbf{P}(n, x) dx, \quad n \geq 0. \end{aligned}$$

Now, multiplying (15) by e^{-sx} and integrating w.r.t. x over 0 to ∞, we obtain

$$\widetilde{\mathbf{w}}(s) = \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \widetilde{\mathbf{P}}(k, s) \widehat{\mathbf{L}}_{n+1-k} \right), \quad \mathrm{Re}(s) \geq 0, \tag{18}$$

where $\widetilde{\mathbf{P}}(k, s)$, $k \geq 0$, can be obtained by taking L.-S.T. of (14) as

$$\begin{aligned} \widetilde{\mathbf{P}}(0, s) &= (s\mathbf{I}_{m_s} - \mathbf{L}_0)^{-1}, \\ \widetilde{\mathbf{P}}(k, s) &= \sum_{r=0}^{k-1} \widetilde{\mathbf{P}}(r, s) \mathbf{L}_{k-r} \widetilde{\mathbf{P}}(0, s), \quad k \geq 1. \end{aligned}$$

Differentiating (18) w.r.t. s and setting $s = 0$, we obtain the mean sojourn time $W = -\frac{d\widetilde{\mathbf{w}}(s)\mathbf{e}}{ds}|_{s=0}$ of an arbitrary customer, and it is given by

$$W = \sum_{n=0}^{\infty} \boldsymbol{\pi}^{-}(n) \sum_{k=0}^{n} \left(\mathbf{I}_{m_a} \otimes \widetilde{\mathbf{P}}^{(1)}(k, 0) \widehat{\mathbf{L}}_{n+1-k} \right) \mathbf{e},$$

where the first order derivative $\widetilde{\mathbf{P}}^{(1)}(k, 0)$ of $\widetilde{\mathbf{P}}(k, s)$ at $s = 0$ is given by

$$\begin{aligned} \widetilde{\mathbf{P}}^{(1)}(0, 0) &= -(-\mathbf{L}_0)^{-2}, \\ \widetilde{\mathbf{P}}^{(1)}(k, 0) &= \sum_{i=0}^{k-1} \left[\widetilde{\mathbf{P}}^{(1)}(i, 0) \mathbf{L}_{k-i} - \widetilde{\mathbf{P}}(i, 0) \mathbf{L}_{k-i} (-\mathbf{L}_0)^{-1} \right] (-\mathbf{L}_0)^{-1}, \quad k \geq 1. \end{aligned}$$

4. Numerical Results

We have done the numerical work based upon the analytical procedure discussed in this paper. To justify our analytical results, we have created outputs based on the diversified inputs but only a few of them are appended here in the forms of tables and graphs. All the calculations are performed using MAPLE software on a PC having configurations as Intel(R) Core i7-6500U processor @ 2.50 GHz with 8 GB DDR2 RAM in Windows 10 environment. All the numerical results were carried out in high precision, but they are reported here in 6 decimal places. The analytical results are not affected by considering the maximum batch size of the arriving batch either less or greater than equal to the maximum service batch size. To show this impact, we have taken two numerical examples with (i) maximum batch size of the arriving batch is less than maximum service batch size, i.e., $N_0 < M_0$, (ii) maximum batch size of the arriving batch is greater than maximum service batch size, i.e., $N_0 > M_0$. The numerical results for these two cases have been presented in Tables 1 - 8.

Example 1: We have presented the system-length distributions at various time epochs as well as the sojourn-time distribution of an arbitrary customer in an arriving batch in Tables 1 - 4. For case (i), we choose the following rate matrices $\mathbf{D}_n$, $n \geq 0$, of order $m_a = 2$ of the arrival process BMAP with maximum arrival batch size $N_0 = 10$:

$$\mathbf{D}_0 = \begin{bmatrix} -0.346 & 0.069 \\ 0.230 & -0.376 \end{bmatrix}, \mathbf{D}_3 = \begin{bmatrix} 0.005 & 0.007 \\ 0.051 & 0.018 \end{bmatrix}, \mathbf{D}_5 = \begin{bmatrix} 0.014 & 0.022 \\ 0.002 & 0.054 \end{bmatrix},$$
$$\mathbf{D}_7 = \begin{bmatrix} 0.011 & 0.032 \\ 0.008 & 0.004 \end{bmatrix}, \mathbf{D}_9 = \begin{bmatrix} 0.006 & 0.080 \\ 0.002 & 0.003 \end{bmatrix}, \mathbf{D}_{10} = \begin{bmatrix} 0.080 & 0.020 \\ 0.003 & 0.001 \end{bmatrix},$$

including $\mathbf{D}_k = \mathbf{0},\ k \in \mathbb{N} - \{3, 5, 7, 9, 10\}$, where $\mathbb{N}$ is the set of natural numbers. Hence, $\overline{\boldsymbol{\pi}}_a \mathbf{D} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_a \mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_a = \begin{bmatrix} 0.562738 & 0.437262 \end{bmatrix}$ with $\lambda^* = 1.576076$.
Again, we choose the following rate matrices $\mathbf{L}_n$, $n \geq 0$, of order $m_s = 3$ of the service process BMSP with maximum service batch size $M_0 = 18$:

$$\mathbf{L}_0 = \begin{bmatrix} -0.414 & 0.069 & 0.014 \\ 0.122 & -0.224 & 0.000 \\ 0.230 & 0.000 & -0.378 \end{bmatrix}, \mathbf{L}_4 = \begin{bmatrix} 0.000 & 0.005 & 0.007 \\ 0.015 & 0.000 & 0.034 \\ 0.000 & 0.051 & 0.018 \end{bmatrix},$$
$$\mathbf{L}_8 = \begin{bmatrix} 0.010 & 0.014 & 0.022 \\ 0.003 & 0.011 & 0.008 \\ 0.001 & 0.002 & 0.054 \end{bmatrix}, \mathbf{L}_{10} = \begin{bmatrix} 0.011 & 0.032 & 0.014 \\ 0.003 & 0.005 & 0.009 \\ 0.008 & 0.004 & 0.001 \end{bmatrix},$$
$$\mathbf{L}_{13} = \begin{bmatrix} 0.006 & 0.010 & 0.080 \\ 0.001 & 0.000 & 0.002 \\ 0.002 & 0.003 & 0.000 \end{bmatrix}, \mathbf{L}_{18} = \begin{bmatrix} 0.020 & 0.080 & 0.020 \\ 0.001 & 0.003 & 0.007 \\ 0.003 & 0.000 & 0.001 \end{bmatrix},$$

including $\mathbf{L}_k = \mathbf{0},\ k \in \mathbb{N} - \{4, 8, 10, 13, 18\}$. Hence, $\overline{\boldsymbol{\pi}}_s\mathbf{L} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_s\mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_s = \begin{bmatrix} 0.332192 & 0.414449 & 0.253359 \end{bmatrix}$ with $\mu^* = 2.035615$. Thus, we have $\rho = 0.774251$. Further, we have $\overline{\boldsymbol{\pi}}_a \otimes \overline{\boldsymbol{\pi}}_s = \begin{bmatrix} 0.186937, 0.233226, 0.142575, 0.145255, 0.181223, 0.110784 \end{bmatrix}$. The characteristic equation $G(z) = 0$ has $m(N_0 + M_0) = 168$ roots in total. Out of these 168 roots, we have $mM_0 - 1 = 107$ roots inside of $|z| = 1$, one root at $z = 1$ and other $mN_0 = 60$ roots outside of $|z| = 1$. Using these 107 inside roots $\gamma_1, \gamma_2, \ldots, \gamma_{107}$ in (6) and $z = 1$ in (7), and then solving these mM_0 linearly independent simultaneous equations, we get the vectors $\boldsymbol{\pi}(n), 0 \le n \le M_0 - 1$. Further, other 60 roots $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_{60}$ outside the unit disk are used in (8) for partial fractions.

Example 2: We have presented the system-length distributions at various time epochs as well as the sojourn-time distribution of an arbitrary customer in an arriving batch in Tables 5 - 8. For case (ii), we choose the following rate matrices $\mathbf{D}_n$, $n \ge 0$, of order $m_a = 3$ of the arrival process BMAP with maximum arrival batch size $N_0 = 15$:

$$\mathbf{D}_0 = \begin{bmatrix} -0.415 & 0.059 & 0.024 \\ 0.222 & -0.328 & 0.000 \\ 0.450 & 0.000 & -0.547 \end{bmatrix}, \mathbf{D}_1 = \begin{bmatrix} 0.000 & 0.006 & 0.008 \\ 0.025 & 0.000 & 0.024 \\ 0.000 & 0.031 & 0.028 \end{bmatrix},$$

$$\mathbf{D}_2 = \begin{bmatrix} 0.020 & 0.024 & 0.012 \\ 0.003 & 0.021 & 0.006 \\ 0.001 & 0.002 & 0.014 \end{bmatrix}, \mathbf{D}_5 = \begin{bmatrix} 0.011 & 0.022 & 0.013 \\ 0.003 & 0.004 & 0.008 \\ 0.007 & 0.003 & 0.002 \end{bmatrix},$$

$$\mathbf{D}_{10} = \begin{bmatrix} 0.005 & 0.011 & 0.080 \\ 0.001 & 0.000 & 0.002 \\ 0.002 & 0.004 & 0.000 \end{bmatrix}, \mathbf{D}_{15} = \begin{bmatrix} 0.020 & 0.070 & 0.030 \\ 0.001 & 0.003 & 0.005 \\ 0.002 & 0.000 & 0.001 \end{bmatrix},$$

including $\mathbf{D}_k = \mathbf{0},\ k \in \mathbb{N} - \{1, 2, 5, 10, 15\}$. Hence, $\overline{\boldsymbol{\pi}}_a\mathbf{D} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_a\mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_a = \begin{bmatrix} 0.479067 & 0.331818 & 0.189115 \end{bmatrix}$ with $\lambda^* = 1.657368$.

Again, we choose the following rate matrices $\mathbf{L}_n$, $n \ge 0$, of order $m_s = 2$ of the service process BMSP with maximum service batch size $M_0 = 8$:

$$\mathbf{L}_0 = \begin{bmatrix} -0.792 & 0.069 \\ 0.230 & -0.910 \end{bmatrix}, \mathbf{L}_1 = \begin{bmatrix} 0.070 & 0.089 \\ 0.061 & 0.088 \end{bmatrix}, \mathbf{L}_2 = \begin{bmatrix} 0.065 & 0.042 \\ 0.062 & 0.034 \end{bmatrix},$$

$$\mathbf{L}_4 = \begin{bmatrix} 0.091 & 0.072 \\ 0.089 & 0.095 \end{bmatrix}, \mathbf{L}_6 = \begin{bmatrix} 0.085 & 0.075 \\ 0.084 & 0.063 \end{bmatrix}, \mathbf{L}_8 = \begin{bmatrix} 0.078 & 0.056 \\ 0.037 & 0.067 \end{bmatrix},$$

including $\mathbf{L}_k = \mathbf{0},\ k \in \mathbb{N} - \{1, 2, 4, 6, 8\}$. Hence, $\overline{\boldsymbol{\pi}}_s\mathbf{L} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_s\mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_s = \begin{bmatrix} 0.582816 & 0.417184 \end{bmatrix}$ with $\mu^* = 2.946029$. Thus, we have $\rho = 0.562577$. Further, we have $\overline{\boldsymbol{\pi}}_a \otimes \overline{\boldsymbol{\pi}}_s = \begin{bmatrix} 0.279208 & 0.199859 & 0.193389 & 0.138429 & 0.110219 & 0.078896 \end{bmatrix}$. The characteristic equation $G(z) = 0$ has $m(N_0 + M_0) = 138$ roots in total. Out of these 138 roots, we have $mM_0 - 1 = 47$ roots inside of $|z| = 1$, one root at $z = 1$ and other $mN_0 = 90$ roots outside of $|z| = 1$. Using these 47 inside roots $\gamma_1, \gamma_2, \dots, \gamma_{47}$ in (6) and $z = 1$ in (7), and then solving these mM_0 linearly independent simultaneous equations, we get the vectors $\boldsymbol{\pi}(n), 0 \le n \le M_0 - 1$. Further, other 90 roots $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_{90}$ outside the unit disk are used in (8) for partial fractions. It is found that the mean sojourn time W_s using Little's rule given in Tables 1 and 5 match with the results obtained from the sojourn-time distribution given in Tables 4 and 8, respectively. Moreover, one can observe from Tables 1 and 5 that $\sum_{n=0}^{\infty} \boldsymbol{\pi}(n) = \boldsymbol{\pi}_a \otimes \boldsymbol{\pi}_s$, where $\sum_{n=0}^{\infty} \boldsymbol{\pi}(n)$ is calculated through roots whereas $\boldsymbol{\pi}_a \otimes \boldsymbol{\pi}_s$ is independent of the roots. This fact confirms the correctness of our analytical and numerical results.

Example 3: For the purpose of comparative study between the roots method and the MGM, we choose the rate matrices $\mathbf{D}_n$, $n \ge 0$, of order $m_a = 3$ of the arrival process BMAP with maximum arrival batch size $N_0 = 10$ such that each entry of $\mathbf{D}_n$, $n \ge 0$, is a function of δ, $(\delta > 0)$, and they are given by

$$\mathbf{D}_0 = \begin{bmatrix} -7\delta^2 & \frac{\delta^2}{2} & \frac{\delta^2}{3} \\ \frac{\delta}{6} & -7\delta & \frac{\delta}{4} \\ \frac{\delta^3}{3} & \frac{\delta^3}{6} & -7\delta^3 \end{bmatrix}, \mathbf{D}_1 = \begin{bmatrix} \frac{\delta^2}{3} & \frac{\delta^2}{4} & \frac{\delta^2}{6} \\ \frac{\delta}{2} & \frac{\delta}{6} & \frac{\delta}{3} \\ \frac{\delta^3}{2} & \frac{\delta^3}{3} & \frac{\delta^3}{6} \end{bmatrix}, \mathbf{D}_2 = \begin{bmatrix} \frac{\delta^2}{2} & \frac{\delta^2}{6} & \frac{\delta^2}{3} \\ \frac{\delta}{4} & \frac{\delta}{2} & \frac{\delta}{6} \\ \frac{\delta^3}{4} & \frac{\delta^3}{3} & \frac{\delta^3}{6} \end{bmatrix},$$

$$\mathbf{D}_4 = \begin{bmatrix} \frac{\delta^2}{4} & \frac{\delta^2}{2} & \frac{\delta^2}{6} \\ \frac{\delta}{2} & \frac{\delta}{3} & \frac{\delta}{4} \\ \frac{\delta^3}{3} & \frac{\delta^3}{6} & \frac{\delta^3}{2} \end{bmatrix}, \mathbf{D}_5 = \begin{bmatrix} \frac{\delta^2}{4} & \frac{\delta^2}{3} & \frac{\delta^2}{4} \\ \frac{\delta}{6} & \frac{\delta}{3} & \frac{\delta}{4} \\ \frac{\delta^3}{4} & \frac{\delta^3}{2} & \frac{\delta^3}{3} \end{bmatrix}, \mathbf{D}_7 = \begin{bmatrix} \frac{\delta^2}{2} & \frac{\delta^2}{6} & \frac{\delta^2}{4} \\ \frac{\delta}{3} & \frac{\delta}{4} & \frac{\delta}{2} \\ \frac{\delta^3}{4} & \frac{\delta^3}{4} & \frac{\delta^3}{6} \end{bmatrix},$$

$$\mathbf{D}_8 = \begin{bmatrix} \frac{\delta^2}{3} & \frac{\delta^2}{4} & \frac{\delta^2}{2} \\ \frac{\delta}{2} & \frac{\delta}{6} & \frac{\delta}{4} \\ \frac{\delta^3}{2} & \frac{\delta^3}{6} & \frac{\delta^3}{4} \end{bmatrix}, \mathbf{D}_{10} = \begin{bmatrix} \frac{\delta^2}{6} & \frac{\delta^2}{3} & \frac{\delta^2}{6} \\ \frac{\delta}{6} & \frac{\delta}{3} & \frac{\delta^2}{3} \\ \frac{\delta^3}{3} & \frac{\delta^3}{2} & \frac{\delta^3}{4} \end{bmatrix},$$

including $\mathbf{D}_k = \mathbf{0},\ k \in \mathbb{N} - \{1, 2, 4, 5, 7, 8, 10\}$.

Similarly, we choose the rate matrices $\mathbf{L}_n$, $n \ge 0$, of order $m_s = 4$ of the service process BMSP with maximum service batch size $M_0 = 15$ such that each entry of $\mathbf{L}_n$, $n \ge 0$, is a function of η, $(\eta > 0)$, and they are given by

$$\mathbf{L}_0 = \begin{bmatrix} -\frac{16}{\eta} & \frac{1}{\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} \\ \frac{1}{\eta} & -\frac{19}{\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} \\ \frac{1}{4\eta} & \frac{1}{\eta} & -\frac{18}{\eta} & \frac{1}{2\eta} \\ \frac{1}{5\eta} & \frac{1}{2\eta} & \frac{1}{4\eta} & -\frac{21}{\eta} \end{bmatrix}, \mathbf{L}_1 = \begin{bmatrix} \frac{1}{\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{5\eta} \\ \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{5\eta} & \frac{1}{\eta} \\ \frac{1}{2\eta} & \frac{1}{5\eta} & \frac{1}{\eta} & \frac{1}{2\eta} \\ \frac{1}{4\eta} & \frac{1}{\eta} & \frac{1}{2\eta} & \frac{1}{5\eta} \end{bmatrix},$$

$$\mathbf{L}_2 = \begin{bmatrix} \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} \\ \frac{1}{\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} & \frac{1}{2\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{5\eta} \\ \frac{1}{\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{3\eta} \end{bmatrix}, \mathbf{L}_3 = \begin{bmatrix} \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{\eta} \\ \frac{1}{2\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} \\ \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{5\eta} & \frac{1}{2\eta} \\ \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{5\eta} \end{bmatrix},$$

$$\mathbf{L}_5 = \begin{bmatrix} \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{4\eta} \\ \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{3\eta} & \frac{1}{5\eta} \\ \frac{1}{2\eta} & \frac{1}{3\eta} & \frac{1}{\eta} & \frac{1}{5\eta} \\ \frac{1}{3\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} \end{bmatrix}, \mathbf{L}_6 = \begin{bmatrix} \frac{1}{3\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{2\eta} & \frac{1}{3\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{3\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{\eta} \end{bmatrix},$$

$$\mathbf{L}_8 = \begin{bmatrix} \frac{1}{\eta} & \frac{1}{4\eta} & \frac{1}{2\eta} & \frac{1}{5\eta} \\ \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{5\eta} & \frac{1}{2\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{\eta} & \frac{1}{2\eta} \\ \frac{1}{4\eta} & \frac{1}{\eta} & \frac{1}{2\eta} & \frac{1}{5\eta} \end{bmatrix}, \mathbf{L}_9 = \begin{bmatrix} \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} & \frac{1}{4\eta} \\ \frac{1}{2\eta} & \frac{1}{5\eta} & \frac{1}{2\eta} & \frac{1}{4\eta} \\ \frac{1}{\eta} & \frac{1}{3\eta} & \frac{1}{2\eta} & \frac{1}{\eta} \end{bmatrix},$$

$$\mathbf{L}_{12} = \begin{bmatrix} \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{\eta} \\ \frac{1}{2\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} \\ \frac{1}{\eta} & \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{5\eta} \end{bmatrix}, \mathbf{L}_{13} = \begin{bmatrix} \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} & \frac{1}{4\eta} \\ \frac{1}{2\eta} & \frac{1}{\eta} & \frac{1}{3\eta} & \frac{1}{5\eta} \\ \frac{1}{5\eta} & \frac{1}{3\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} \\ \frac{1}{3\eta} & \frac{1}{5\eta} & \frac{1}{\eta} & \frac{1}{5\eta} \end{bmatrix},$$

$$\mathbf{L}_{15} = \begin{bmatrix} \frac{1}{4\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{2\eta} & \frac{1}{4\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{4\eta} & \frac{1}{3\eta} & \frac{1}{5\eta} & \frac{1}{3\eta} \\ \frac{1}{\eta} & \frac{1}{5\eta} & \frac{1}{4\eta} & \frac{1}{3\eta} \end{bmatrix},$$

including $\mathbf{L}_k = \mathbf{0},\ k \in \mathbb{N} - \{1, 2, 3, 5, 6, 8, 9, 12, 13, 15\}$.

All the stationary probabilities at random epoch carried out by the MGM also match with those obtained using the method of roots. We found during the computational work that both the methods give the same results, but from computation time point of view one method slightly differ from the other. We have recorded the computation times to calculate $\boldsymbol{\pi}(n)$, $n \geq 0$, for different traffic intensity (ρ) by the roots method and the MGM when $m_a = 3$ and $m_s = 4$. These computation times are given in Table 9. Figure 1 graphically displays the results given in Table 9. It is observed from Figure 1 that the computation time required in the roots method is higher as compared to the MGM for different ρ. Further, in both the methods, we found the similar effect for different lower orders of input matrices $\mathbf{D}_k$ and $\mathbf{L}_k$, $k \geq 0$ during the computational work. Based on computation time, Gupta *et al.* [11], and Bank and Samanta [5] have shown that the roots method is numerically more efficient than the matrix-analytic method developed by Neuts [25] for $BMAP/G/1$ and $BMAP/G^{(a,Y)}/1$ queue, respectively. However, we have found in this work that the MGM is numerically more efficient than the roots method. Hence, it concludes that all the methods give the same results but computation time depends on respective problems. Authors suggest the reader and practitioner to use the MGM instead of the roots method for the problem considered in this paper.

Example 4: We have demonstrated the numerical stability (especially when some of the roots gets close) of the root finding method based on the mathematical software package Maple. For this purpose, we choose the rate matrices $\mathbf{D}_n$, $n \geq 0$, of order $m_a = 3$ of the arrival process BMAP with maximum arrival batch size $N_0 = 10$ and they are given by

$$\mathbf{D}_0 = \begin{bmatrix} -0.588700 & 0.042050 & 0.028033 \\ 0.048333 & -2.030000 & 0.072500 \\ 0.008130 & 0.004065 & -0.170723 \end{bmatrix}, \ \mathbf{D}_1 = \begin{bmatrix} 0.028033 & 0.021025 & 0.014017 \\ 0.145000 & 0.048333 & 0.096667 \\ 0.012195 & 0.008129 & 0.004065 \end{bmatrix},$$

$$\mathbf{D}_2 = \begin{bmatrix} 0.042050 & 0.014017 & 0.028033 \\ 0.072500 & 0.145000 & 0.048333 \\ 0.006097 & 0.008129 & 0.004064 \end{bmatrix}, \ \mathbf{D}_3 = \begin{bmatrix} 0.021025 & 0.042050 & 0.014017 \\ 0.145000 & 0.096667 & 0.072500 \\ 0.008130 & 0.004065 & 0.012194 \end{bmatrix},$$

$$\mathbf{D}_4 = \begin{bmatrix} 0.021025 & 0.028033 & 0.021025 \\ 0.048333 & 0.096667 & 0.072500 \\ 0.006097 & 0.012195 & 0.008130 \end{bmatrix}, \ \mathbf{D}_7 = \begin{bmatrix} 0.042050 & 0.014017 & 0.021025 \\ 0.096667 & 0.072500 & 0.145000 \\ 0.006097 & 0.006097 & 0.004065 \end{bmatrix},$$

$$\mathbf{D}_8 = \begin{bmatrix} 0.028033 & 0.021025 & 0.042050 \\ 0.145000 & 0.048333 & 0.072500 \\ 0.012195 & 0.004065 & 0.006097 \end{bmatrix}, \ \mathbf{D}_{10} = \begin{bmatrix} 0.014017 & 0.028033 & 0.014017 \\ 0.048333 & 0.096667 & 0.096667 \\ 0.008130 & 0.012195 & 0.006097 \end{bmatrix},$$

including $\mathbf{D}_k = \mathbf{0}, \ k \in \mathbb{N} - \{1, 2, 3, 4, 7, 8, 10\}$. Hence, $\overline{\boldsymbol{\pi}}_a \mathbf{D} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_a \mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_a = \begin{bmatrix} 0.241387 & 0.064296 & 0.694317 \end{bmatrix}$ with $\lambda^* = 1.776913$.
Similarly, we choose the rate matrices $\mathbf{L}_n$, $n \geq 0$, of order $m_s = 2$ of the service process BMSP with maximum service batch size $M_0 = 10$ and they are given by

$$\mathbf{L}_0 = \begin{bmatrix} -1.747578 & 0.194176 \\ 0.009426 & -0.339337 \end{bmatrix}, \ \mathbf{L}_1 = \begin{bmatrix} 0.194175 & 0.048544 \\ 0.037704 & 0.012568 \end{bmatrix},$$

$$\mathbf{L}_2 = \begin{bmatrix} 0.097087 & 0.038835 \\ 0.007541 & 0.037704 \end{bmatrix}, \ \mathbf{L}_3 = \begin{bmatrix} 0.097087 & 0.064725 \\ 0.009426 & 0.018852 \end{bmatrix},$$

$$\mathbf{L}_4 = \begin{bmatrix} 0.048545 & 0.038835 \\ 0.007541 & 0.012568 \end{bmatrix}, \ \mathbf{L}_5 = \begin{bmatrix} 0.194175 & 0.038835 \\ 0.012568 & 0.009426 \end{bmatrix},$$

$$\mathbf{L}_6 = \begin{bmatrix} 0.048545 & 0.064725 \\ 0.018852 & 0.007541 \end{bmatrix}, \ \mathbf{L}_7 = \begin{bmatrix} 0.064725 & 0.038835 \\ 0.037704 & 0.012568 \end{bmatrix},$$

$$\mathbf{L}_8 = \begin{bmatrix} 0.194175 & 0.064725 \\ 0.007541 & 0.009426 \end{bmatrix}, \ \mathbf{L}_9 = \begin{bmatrix} 0.064725 & 0.038835 \\ 0.037704 & 0.007541 \end{bmatrix},$$

$$\mathbf{L}_{10} = \begin{bmatrix} 0.048544 & 0.064725 \\ 0.012568 & 0.012568 \end{bmatrix},$$

including $\mathbf{L}_k = \mathbf{0}, \ k \in \mathbb{N} - \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. Hence, $\overline{\boldsymbol{\pi}}_s \mathbf{L} = \mathbf{0}$ and $\overline{\boldsymbol{\pi}}_s \mathbf{e} = 1$ yield $\overline{\boldsymbol{\pi}}_s = \begin{bmatrix} 0.222028 & 0.777972 \end{bmatrix}$ with $\mu^* = 3.126994$. Thus, we have $\rho = 0.568249$.
The characteristic equation $G(z) = 0$ has $m(N_0 + M_0) = 120$ roots in total. Out of these 120 roots, we have $mM_0 - 1 = 59$ roots $\gamma_1, \gamma_2, \ldots, \gamma_{59}$ inside of $|z| = 1$, one root at $z = 1$ and other $mN_0 = 60$ roots $\alpha_1, \alpha_2, \ldots, \alpha_{60}$ outside of $|z| = 1$. The accuracy of these roots is

also verified by back substituting into the characteristic equation $G(z) = 0$. All the roots of $G(z) = 0$ and accuracy of these roots are reported in Table 10 and Table 11 up to 30 decimal places. Three roots γ_1, γ_2 and γ_3 inside the unit disc are very closed. The first 7 decimals of roots γ_1, γ_2 and γ_3 are same. One may remark here that we can obtain close roots however close they may get using the mathematical software package Maple. Therefore, Maple identifies these roots as distinct roots and hence calculates these roots accurately. For visual illustration purpose, we have also plotted all 120 roots of $G(z) = 0$ in Figure 2. Finally, we have not found any effect in numerical results carried out by the roots method and all the results are perfectly matched with those obtained using the MGM.

Table 1. System-length distribution at random epoch.

n	$\pi_{11}(n)$	$\pi_{12}(n)$	$\pi_{13}(n)$	$\pi_{21}(n)$	$\pi_{22}(n)$	$\pi_{23}(n)$	$\boldsymbol{\pi}(n)\mathbf{e}$
0	0.025960	0.035912	0.028303	0.015495	0.021967	0.017660	0.145297
1	0.001635	0.002079	0.001827	0.001239	0.001609	0.001610	0.009999
2	0.001669	0.002143	0.001993	0.000950	0.001249	0.001165	0.009169
3	0.003893	0.004840	0.003154	0.002118	0.002670	0.001854	0.018528
4	0.001436	0.001819	0.001558	0.000899	0.001175	0.001047	0.007935
5	0.003972	0.004860	0.003066	0.004372	0.005432	0.003591	0.025293
6	0.002156	0.002627	0.002237	0.001412	0.001734	0.001541	0.011707
7	0.003403	0.004169	0.002743	0.003155	0.003903	0.002580	0.019952
8	0.002233	0.002752	0.001890	0.001618	0.002009	0.001427	0.011928
9	0.004304	0.005224	0.003054	0.005981	0.007336	0.004524	0.030423
10	0.008003	0.009759	0.005609	0.004077	0.004957	0.002923	0.035328
20	0.003462	0.004192	0.002227	0.002288	0.002765	0.001586	0.016521
50	0.001212	0.001494	0.000802	0.000981	0.001208	0.000650	0.006346
100	0.000306	0.000378	0.000202	0.000248	0.000306	0.000164	0.001604
150	0.000077	0.000096	0.000051	0.000063	0.000077	0.000041	0.000405
200	0.000020	0.000024	0.000013	0.000016	0.000020	0.000010	0.000102
300	0.000001	0.000002	0.000000	0.000001	0.000001	0.000000	0.000005
350	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.186937	0.233226	0.142575	0.145255	0.181223	0.110784	1.000000
$L_s = 33.125126, W_s = 21.017467$							

Table 2. System-length distribution at pre-arrival epoch.

n	$\pi^-_{11}(n)$	$\pi^-_{12}(n)$	$\pi^-_{13}(n)$	$\pi^-_{21}(n)$	$\pi^-_{22}(n)$	$\pi^-_{23}(n)$	$\boldsymbol{\pi}^-(n)\mathbf{e}$
0	0.002560	0.003563	0.002823	0.003438	0.004783	0.003788	0.020955
1	0.002732	0.003783	0.003025	0.003668	0.005078	0.004056	0.022342
2	0.002894	0.003993	0.003220	0.003887	0.005360	0.004319	0.023673
3	0.002686	0.003637	0.002869	0.004100	0.005579	0.004408	0.023278
4	0.002784	0.003761	0.002969	0.004271	0.005797	0.004593	0.024176
5	0.002973	0.003952	0.003027	0.003987	0.005292	0.004067	0.023298
6	0.003094	0.004096	0.003168	0.004172	0.005512	0.004258	0.024300
7	0.003167	0.004140	0.003133	0.004041	0.005257	0.003963	0.023700
8	0.003190	0.004166	0.003155	0.004120	0.005351	0.004025	0.024007
9	0.003556	0.004590	0.003347	0.003403	0.004248	0.002968	0.022112
10	0.002769	0.003404	0.002247	0.003663	0.004497	0.002946	0.019526
20	0.002328	0.002818	0.001658	0.003034	0.003673	0.002138	0.015649
50	0.001055	0.001299	0.000698	0.001346	0.001657	0.000890	0.006946
100	0.000266	0.000329	0.000176	0.000340	0.000420	0.000224	0.001755
150	0.000067	0.000083	0.000044	0.000086	0.000106	0.000057	0.000443
200	0.000017	0.000021	0.000011	0.000022	0.000027	0.000014	0.000112
300	0.000001	0.000001	0.000000	0.000001	0.000002	0.000000	0.000005
350	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.145114	0.181047	0.110677	0.187078	0.233401	0.142682	1.000000

Table 3. System-length distribution at post-departure epoch.

n	$\pi^+_{11}(n)$	$\pi^+_{12}(n)$	$\pi^+_{13}(n)$	$\pi^+_{21}(n)$	$\pi^+_{22}(n)$	$\pi^+_{23}(n)$	$\boldsymbol{\pi}^+(n)\mathbf{e}$
0	0.003672	0.008713	0.009515	0.002309	0.005749	0.006286	0.036244
1	0.003387	0.008385	0.011099	0.003574	0.008793	0.012784	0.048022
2	0.003939	0.009785	0.013749	0.002565	0.006396	0.008690	0.045125
3	0.002810	0.006885	0.009078	0.002289	0.005670	0.007579	0.034311
4	0.002784	0.006863	0.009269	0.002311	0.005660	0.007490	0.034377
5	0.003151	0.007468	0.010000	0.003035	0.007266	0.009838	0.040758
6	0.003889	0.009038	0.013084	0.002778	0.006533	0.009618	0.044941
7	0.002783	0.006825	0.009560	0.002161	0.005384	0.007152	0.033866
8	0.002686	0.006473	0.008707	0.002231	0.005356	0.007198	0.032651
9	0.002843	0.006816	0.009081	0.002286	0.005530	0.007123	0.033679
10	0.002701	0.006475	0.008365	0.002216	0.005289	0.007039	0.032084
20	0.002082	0.004911	0.006656	0.001705	0.004007	0.005482	0.024843
50	0.000918	0.002145	0.002949	0.000744	0.001739	0.002390	0.010885
100	0.000232	0.000542	0.000745	0.000188	0.000439	0.000604	0.002749
150	0.000059	0.000137	0.000188	0.000047	0.000111	0.000153	0.000695
200	0.000015	0.000035	0.000048	0.000012	0.000028	0.000039	0.000176
300	0.000000	0.000002	0.000003	0.000000	0.000002	0.000002	0.000009
350	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.133480	0.315799	0.429937	0.107836	0.255765	0.348950	1.000000

Table 4. Sjourn-time distribution.

x	$W_{11}(x)$	$W_{12}(x)$	$W_{13}(x)$	$W_{21}(x)$	$W_{22}(x)$	$W_{23}(x)$	$\boldsymbol{W}(x)\mathbf{e}$
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.5	0.001425	0.003731	0.004172	0.001893	0.004953	0.005579	0.021753
1.0	0.002846	0.007466	0.008331	0.003776	0.009897	0.011122	0.043438
1.5	0.004257	0.011184	0.012459	0.005641	0.014806	0.016605	0.064952
2.0	0.005652	0.014870	0.016544	0.007481	0.019661	0.022012	0.086219
2.5	0.007028	0.018511	0.020574	0.009292	0.024446	0.027332	0.107182
3.0	0.008383	0.022099	0.024542	0.011072	0.029152	0.032556	0.127803
3.5	0.009715	0.025628	0.028442	0.012817	0.033771	0.037680	0.148053
4.0	0.011022	0.029094	0.032272	0.014528	0.038298	0.042699	0.167913
4.5	0.012305	0.032494	0.036028	0.016203	0.042732	0.047611	0.187372
5.0	0.013561	0.035825	0.039710	0.017842	0.047069	0.052416	0.206424
10.5	0.025712	0.067990	0.075303	0.033572	0.088658	0.098511	0.389746
20.0	0.040613	0.107303	0.118945	0.052681	0.139049	0.154480	0.613071
50.5	0.060534	0.159765	0.177285	0.078113	0.206019	0.228957	0.910673
100.0	0.065963	0.174058	0.193184	0.085039	0.224251	0.249238	0.991732
150.5	0.066468	0.175388	0.194662	0.085683	0.225946	0.251124	0.999271
200.0	0.066512	0.175504	0.194792	0.085739	0.226095	0.251289	0.999932
250.5	0.066516	0.175515	0.194804	0.085744	0.226109	0.251305	0.999994
285.0	0.066517	0.175516	0.194805	0.085745	0.226110	0.251306	0.999999
300.5	0.066517	0.175516	0.194805	0.085745	0.226110	0.251306	0.999999
$W = 21.017467$							

Table 5. System-length distribution at random epoch.

n	$\pi_{11}(n)$	$\pi_{12}(n)$	$\pi_{21}(n)$	$\pi_{22}(n)$	$\pi_{31}(n)$	$\pi_{32}(n)$	$\boldsymbol{\pi}(n)\mathbf{e}$
0	0.117391	0.087516	0.068588	0.051248	0.023642	0.018123	0.366507
1	0.009346	0.006687	0.005810	0.004290	0.004639	0.003333	0.034106
2	0.010036	0.006886	0.007872	0.005362	0.004388	0.003145	0.037688
3	0.006211	0.004441	0.004165	0.002975	0.002226	0.001631	0.021649
4	0.006432	0.004701	0.003827	0.002867	0.002895	0.002158	0.022879
5	0.008388	0.005777	0.006767	0.004631	0.003738	0.002647	0.031947
6	0.006145	0.004442	0.003698	0.002727	0.002923	0.002171	0.022106
7	0.006130	0.004394	0.004658	0.003375	0.002310	0.001711	0.022578
8	0.005520	0.003916	0.003428	0.002500	0.002509	0.001760	0.019633
9	0.006470	0.004774	0.004694	0.003395	0.002896	0.002365	0.024595
10	0.008954	0.005917	0.004867	0.003413	0.009904	0.006649	0.039703
20	0.003296	0.002215	0.002681	0.001806	0.001957	0.001306	0.013262
50	0.000343	0.000232	0.000286	0.000194	0.000194	0.000131	0.001381
100	0.000008	0.000005	0.000007	0.000005	0.000005	0.000003	0.000033
155	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.279208	0.199859	0.193389	0.138429	0.110219	0.078896	1.000000
$L_s = 9.994732, W_s = 6.030486$							

Table 6. System-length distribution at pre-arrival epoch.

n	$\pi_{11}^{-}(n)$	$\pi_{12}^{-}(n)$	$\pi_{21}^{-}(n)$	$\pi_{22}^{-}(n)$	$\pi_{31}^{-}(n)$	$\pi_{32}^{-}(n)$	$\boldsymbol{\pi}^{-}(n)\mathbf{e}$
0	0.005503	0.004109	0.011150	0.008326	0.012633	0.009434	0.051155
1	0.004934	0.003671	0.011243	0.008360	0.011764	0.008748	0.048719
2	0.003819	0.002809	0.009569	0.007054	0.011457	0.008450	0.043157
3	0.003883	0.002859	0.009858	0.007261	0.011802	0.008702	0.044364
4	0.003996	0.002951	0.010195	0.007521	0.012273	0.009056	0.045991
5	0.003297	0.002404	0.009083	0.006647	0.011788	0.008639	0.041857
6	0.003319	0.002425	0.009317	0.006819	0.012104	0.008876	0.042859
7	0.003365	0.002463	0.009495	0.006958	0.012451	0.009131	0.043862
8	0.003426	0.002505	0.009755	0.007138	0.012798	0.009374	0.044997
9	0.003562	0.002608	0.010106	0.007407	0.013291	0.009745	0.046719
10	0.003380	0.002445	0.009971	0.007231	0.008390	0.006008	0.017469
20	0.001570	0.001088	0.004368	0.003030	0.004384	0.003028	0.037425
50	0.000176	0.000120	0.000493	0.000335	0.000477	0.000324	0.001925
100	0.000004	0.000003	0.000012	0.000008	0.000011	0.000008	0.000045
155	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.089037	0.063733	0.241517	0.172880	0.252261	0.180571	1.000000

Table 7. System-length distribution at post-departure epoch.

n	$\pi_{11}^{+}(n)$	$\pi_{12}^{+}(n)$	$\pi_{21}^{+}(n)$	$\pi_{22}^{+}(n)$	$\pi_{31}^{+}(n)$	$\pi_{32}^{+}(n)$	$\boldsymbol{\pi}^{+}(n)\mathbf{e}$
0	0.012118	0.011283	0.007926	0.007256	0.005658	0.005317	0.049558
1	0.012209	0.011768	0.009274	0.008999	0.005182	0.005089	0.052520
2	0.010725	0.009967	0.006527	0.006122	0.006343	0.005913	0.045595
3	0.010876	0.009916	0.008040	0.007188	0.004863	0.004440	0.045323
4	0.011018	0.010361	0.007158	0.006947	0.007309	0.006615	0.049407
5	0.009750	0.009036	0.007145	0.006494	0.004510	0.004240	0.041174
6	0.009961	0.009317	0.006569	0.006277	0.006863	0.006279	0.045267
7	0.010147	0.009363	0.008758	0.008085	0.005087	0.004706	0.046145
8	0.009247	0.008420	0.006405	0.005947	0.006145	0.005116	0.052520
9	0.010602	0.010038	0.009370	0.008449	0.007163	0.007366	0.052987
10	0.007422	0.007034	0.005751	0.005512	0.003807	0.003593	0.033119
20	0.003755	0.003511	0.003071	0.002864	0.002071	0.001943	0.017215
50	0.000407	0.000382	0.000340	0.000318	0.000230	0.000215	0.001891
100	0.000010	0.000009	0.000008	0.000007	0.000005	0.000005	0.000044
155	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Sum	0.220740	0.206288	0.173893	0.162351	0.122421	0.114307	1.000000

Table 8. Sjourn-time distribution.

x	$W_{11}(x)$	$W_{12}(x)$	$W_{21}(x)$	$W_{22}(x)$	$W_{31}(x)$	$W_{32}(x)$	$\boldsymbol{W}(x)\mathbf{e}$
0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.5	0.005708	0.005147	0.013739	0.012397	0.016009	0.014466	0.067466
1.0	0.011091	0.009997	0.027174	0.024510	0.031546	0.028479	0.132798
1.5	0.016175	0.014574	0.040231	0.036277	0.046441	0.041896	0.195596
2.0	0.020974	0.018892	0.052838	0.047631	0.060584	0.054623	0.255543
2.5	0.025495	0.022959	0.064928	0.058513	0.073910	0.066606	0.312411
3.0	0.029744	0.026780	0.076446	0.068876	0.086390	0.077821	0.366057
3.5	0.033727	0.030359	0.087356	0.078686	0.098020	0.088267	0.416414
4.0	0.037449	0.033704	0.097633	0.087924	0.108814	0.097959	0.463482
4.5	0.040918	0.036820	0.107269	0.096582	0.118800	0.106922	0.507311
5.0	0.044143	0.039716	0.116265	0.104663	0.128014	0.115191	0.547993
6.0	0.049900	0.044884	0.132397	0.119147	0.144298	0.129801	0.620428
7.5	0.056980	0.051237	0.152306	0.137014	0.164037	0.147506	0.709080
9.0	0.062485	0.056176	0.167801	0.150915	0.179192	0.161097	0.777666
10.5	0.066729	0.059982	0.179738	0.161621	0.190780	0.171489	0.830339
20.0	0.078002	0.070092	0.211352	0.189969	0.221322	0.198876	0.969613
50.5	0.080458	0.072294	0.218218	0.196127	0.227957	0.204825	0.999879
60.0	0.080466	0.072301	0.218241	0.196147	0.227979	0.204845	0.999978
90.5	0.080468	0.072303	0.218246	0.196152	0.227983	0.204849	1.000000
100.0	0.080468	0.072303	0.218246	0.196152	0.227983	0.204849	1.000000
$W = 6.030486$							

Table 9. Computation time to get $\boldsymbol{\pi}(n)$, $n \geq 0$, for different ρ.

δ	η	ρ	Time (Second)	
			Roots method	MGM
0.35	4.0	0.105963	19054.76	1275.79
0.40	5.5	0.204190	18873.11	1399.18
0.45	6.5	0.322783	18920.46	1815.48
0.48	7.0	0.406492	19109.17	2135.43
0.51	7.5	0.503497	19187.34	2556.84
0.54	8.0	0.614643	19067.51	3012.96
0.57	8.5	0.740728	19145.84	4544.60
0.59	8.8	0.830315	19221.37	5241.85
0.61	9.0	0.916410	19404.07	6342.23

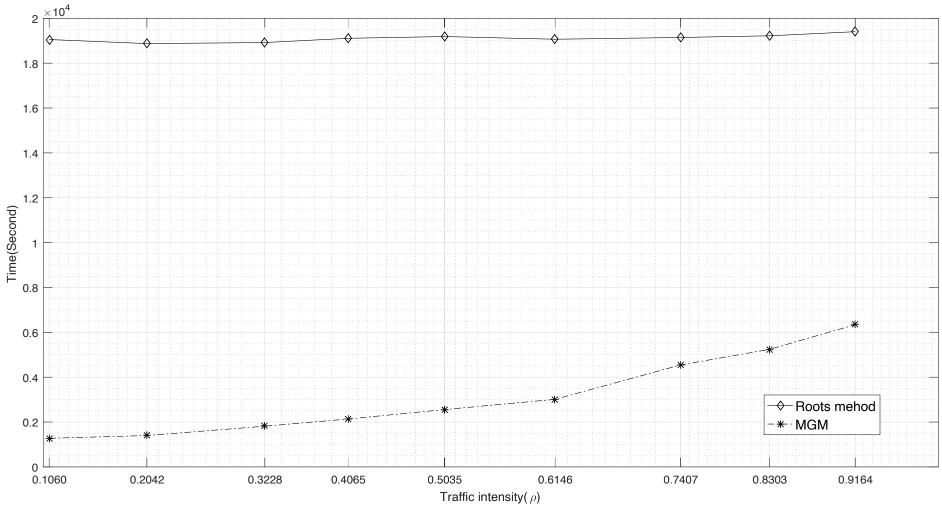

Figure 1. Computation time to get $\boldsymbol{\pi}(n)$, $n \geq 0$, for different ρ.

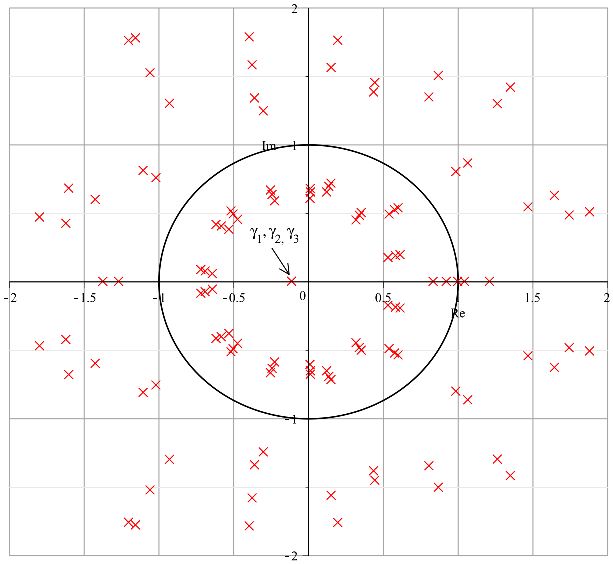

Figure 2. Location of all roots of $G(z) = 0$.

Table 10. The roots of $G(z) = 0$ in $|z| \leq 1$ and their accuracy.

γ_i	Roots	$\lvert G(\gamma_i)\rvert$
γ_1	$-0.110400923414066008759219539984$	$8.710000000000000000000000000000 \times 10^{-523}$
γ_2	$-0.110400926619440682553371298224$	$1.432900000000000000000000000000 \times 10^{-523}$
γ_3	$-0.110400937630677526927125064109$	$8.187000000000000000000000000000 \times 10^{-523}$
γ_4	$0.320400084706996544771508990594 - 0.448479642118275453126191570797i$	$5.846366392897386652806509325200 \times 10^{-521}$
γ_5	$0.320400084706996544771508990594 + 0.448479642118275453126191570797i$	$5.846366392897386652806509325200 \times 10^{-521}$
γ_6	$0.534289368244009405053690490891 - 0.174280143043974380915847249993i$	$2.373785162983373270203501468228 \times 10^{-520}$
γ_7	$0.534289368244009405053690490891 + 0.174280143043974380915847249993i$	$2.373785162983373270203501468228 \times 10^{-520}$
γ_8	$0.342905443731415114392731765870 - 0.483087168881854594723545070216$	$2.965889411289638762750906457704 \times 10^{-520}$
γ_9	$0.342905443731415114392731765870 + 0.483087168881854594723545070216$	$2.965889411289638762750906457704 \times 10^{-520}$
γ_{10}	$0.012948559373001676316748782166 - 0.606361669433420426886292320274i$	$2.778160542517296641479044551327 \times 10^{-520}$
γ_{11}	$0.012948559373001676316748782166 + 0.606361669433420426886292320274i$	$2.778160542517296641479044551327 \times 10^{-520}$
γ_{12}	$0.584540238643891927378272258587 - 0.188137175756215312036505015909i$	$4.873704956190926418480916666996 \times 10^{-520}$
γ_{13}	$0.584540238643891927378272258587 + 0.188137175756215312036505015909i$	$4.873704956190926418480916666996 \times 10^{-520}$
γ_{14}	$0.355534381073907957424412863835 - 0.502403835658666519663103500442i$	$1.478276022940235610178880719663 \times 10^{-520}$
γ_{15}	$0.355534381073907957424412863835 + 0.502403835658666519663103500442i$	$1.478276022940235610178880719663 \times 10^{-520}$
γ_{16}	$-0.223434443737494889747625346811 - 0.588495978175506378581370598181i$	$5.636922919465903074625929111898 \times 10^{-520}$
γ_{17}	$-0.223434443737494889747625346811 + 0.588495978175506378581370598181i$	$5.636922919465903074625929111898 \times 10^{-520}$
γ_{18}	$-0.640071681385930657412212340735 - 0.056965621936319161263743857009i$	$1.118209729880758996787562196920 \times 10^{-519}$
γ_{19}	$-0.640071681385930657412212340735 + 0.056965621936319161263743857009i$	$1.118209729880758996787562196920 \times 10^{-519}$
γ_{20}	$0.615127658043940124937050815342 - 0.194644212087934559518578277393i$	$2.268540059156990068601507141902 \times 10^{-519}$
γ_{21}	$0.615127658043940124937050815342 + 0.194644212087934559518578277393i$	$2.268540059156990068601507141902 \times 10^{-519}$
γ_{22}	$-0.529273257260324729581546137657 - 0.380528498435929597816870865005i$	$1.019306627075484092283009434320 \times 10^{-519}$
γ_{23}	$-0.529273257260324729581546137657 + 0.380528498435929597816870865005i$	$1.019306627075484092283009434320 \times 10^{-519}$
γ_{24}	$0.016358368718113427481296631814 - 0.652842813031811305219082768141i$	$1.325634187851233716061141821111 \times 10^{-519}$
γ_{25}	$0.016358368718113427481296631814 + 0.652842813031811305219082768141i$	$1.325634187851233716061141821111 \times 10^{-519}$
γ_{26}	$-0.471596637789278595535930119479 - 0.452638323206608187953603830622i$	$7.785788335165553420158100924347 \times 10^{-520}$
γ_{27}	$-0.471596637789278595535930119479 + 0.452638323206608187953603830622i$	$7.785788335165553420158100924347 \times 10^{-520}$
γ_{28}	$0.124503176708051754930169679936 - 0.652865015896841998465172600065i$	$3.002043470704579920607140165009 \times 10^{-519}$
γ_{29}	$0.124503176708051754930169679936 + 0.652865015896841998465172600065i$	$3.002043470704579920607140165009 \times 10^{-519}$
γ_{30}	$0.015084753071278079545718095598 - 0.678067815269462670357808134190i$	$3.816895236969440323284286425145 \times 10^{-519}$
γ_{31}	$0.015084753071278079545718095598 + 0.678067815269462670357808134190i$	$3.816895236969440323284286425145 \times 10^{-519}$
γ_{32}	$-0.240653849473855393028212011443 - 0.634718428692994117058897512030i$	$5.778867449595984633618545064892 \times 10^{-519}$
γ_{33}	$-0.240653849473855393028212011443 + 0.634718428692994117058897512030i$	$5.778867449595984633618545064892 \times 10^{-519}$
γ_{34}	$-0.687833860398420622671893260326 - 0.075543394088017605551868100684i$	$4.064556679393215202947029909553 \times 10^{-519}$
γ_{35}	$-0.687833860398420622671893260326 + 0.075543394088017605551868100684i$	$4.064556679393215202947029909553 \times 10^{-519}$
γ_{36}	$-0.501255415530992723045502944800 - 0.492043283759255705164782861517i$	$1.779843468538736414237794948599 \times 10^{-518}$
γ_{37}	$-0.501255415530992723045502944800 + 0.492043283759255705164782861517i$	$1.779843468538736414237794948599 \times 10^{-518}$
γ_{38}	$-0.580174182760349065804893157044 - 0.403840289649339939073437419866i$	$2.131930639115635018786985157267 \times 10^{-518}$
γ_{39}	$-0.580174182760349065804893157044 + 0.403840289649339939073437419866i$	$2.131930639115635018786985157267 \times 10^{-518}$
γ_{40}	$0.138469042798308448991519733374 - 0.694280774023043750478549138476i$	$1.713420555497102701897050868450 \times 10^{-519}$
γ_{41}	$0.138469042798308448991519733374 + 0.694280774023043750478549138476i$	$1.713420555497102701897050868450 \times 10^{-519}$
γ_{42}	$-0.251416877563965724773465905037 - 0.667030019861702725544667135491i$	$8.898770982557085674850197844482 \times 10^{-519}$
γ_{43}	$-0.251416877563965724773465905037 + 0.667030019861702725544667135491i$	$8.898770982557085674850197844482 \times 10^{-519}$
γ_{44}	$-0.716786564835662238755589951188 - 0.086714798775763102496995562276i$	$1.748009739675382930044378287587 \times 10^{-518}$
γ_{45}	$-0.716786564835662238755589951188 + 0.086714798775763102496995562276i$	$1.748009739675382930044378287587 \times 10^{-518}$
γ_{46}	$-0.514978630616171132393271122059 - 0.514480743247183818002472355331i$	$2.430680478984434515014623959482 \times 10^{-518}$
γ_{47}	$-0.514978630616171132393271122059 + 0.514480743247183818002472355331i$	$2.430680478984434515014623959482 \times 10^{-518}$
γ_{48}	$0.541354201205154565368734693881 - 0.491397475595220674574637133772i$	$4.651760759110468269138666247723 \times 10^{-518}$
γ_{49}	$0.541354201205154565368734693881 + 0.491397475595220674574637133772i$	$4.651760759110468269138666247723 \times 10^{-518}$
γ_{50}	$0.152926921894043504472507911658 - 0.717352723559405107009099227323i$	$1.390969809161938640241822774457 \times 10^{-518}$
γ_{51}	$0.152926921894043504472507911658 + 0.717352723559405107009099227323i$	$1.390969809161938640241822774457 \times 10^{-518}$
γ_{52}	$-0.615311495969519174302685320755 - 0.416266215481631873556918712574i$	$1.545894774070990900443706491259 \times 10^{-517}$
γ_{53}	$-0.615311495969519174302685320755 + 0.416266215481631873556918712574i$	$1.545894774070990900443706491259 \times 10^{-517}$
γ_{54}	$0.578617995650982326004796359876 - 0.522951414239860490188882322652i$	$4.158773972470732870088570729015 \times 10^{-517}$
γ_{55}	$0.578617995650982326004796359876 + 0.522951414239860490188882322652i$	$4.158773972470732870088570729015 \times 10^{-517}$
γ_{56}	$0.601688973920937729418538173776 - 0.537233410818626848870880775336i$	$1.792840458246354948093019116693 \times 10^{-516}$
γ_{57}	$0.601688973920937729418538173776 + 0.537233410818626848870880775336i$	$1.792840458246354948093019116693 \times 10^{-516}$
γ_{58}	$0.836113067283192371891396436700$	$7.239693000000000000000000000000 \times 10^{-515}$
γ_{59}	$0.923850396491094858988057241037$	$1.040653584000000000000000000000 \times 10^{-512}$
z	$1.000000000000000000000000000000$	$0.000000000000000000000000000000$

Table 11. The roots of $G(z) = 0$ in $|z| > 1$ and their accuracy.

α_i	Roots	$\|G(\alpha_i)\|$
α_1	$1.04503636957342933197559629925$	$2.74450801539200000000000000000 \times 10^{-509}$
α_2	$1.21276451032548703712248019111 9$	$3.02807578007761900000000000000 \times 10^{-505}$
α_3	$-1.26663270802758564565660682022 4$	$3.79243847969338500000000000000 \times 10^{-504}$
α_4	$-1.01686711474670589086554837470 4 + 0.75672847921766898270432832137 2$	$8.44223427511808881147170712372 4 \times 10^{-504}$
α_5	$-1.01686711474670589086554837470 4 - 0.75672847921766898270432832137 2$	$8.44223427511808881147170712372 4 \times 10^{-504}$
α_6	$0.98761745268448464327302915327 0 + 0.80188406608974678741454692384 0i$	$3.07986199779377607943187381168 7 \times 10^{-505}$
α_7	$0.98761745268448464327302915327 0 - 0.80188406608974678741454692384 0i$	$3.07986199779377607943187381168 7 \times 10^{-505}$
α_8	$-0.29930110307614064672459541926 7 + 1.24505216789948804892538318013 0i$	$3.16688072985733010012233732513 3 \times 10^{-504}$
α_9	$-0.29930110307614064672459541926 7 - 1.24505216789948804892538318013 0i$	$3.16688072985733010012233732513 3 \times 10^{-504}$
α_{10}	$-1.10239449096765203494276641300 4 + 0.81045436156616999702955257847 2i$	$1.31970911427580177217554834440 1 \times 10^{-501}$
α_{11}	$-1.10239449096765203494276641300 4 - 0.81045436156616999702955257847 2i$	$1.31970911427580177217554834440 1 \times 10^{-501}$
α_{12}	$-1.37203365459312814166409495259 2$	$7.96867354503915653800000000000 \times 10^{-502}$
α_{13}	$1.06750748173616794206035079297 1 + 0.86468245441944105851755131620 6i$	$2.91908607608438338695923578252 7 \times 10^{-502}$
α_{14}	$1.06750748173616794206035079297 1 - 0.86468245441944105851755131620 6i$	$2.91908607608438338695923578252 7 \times 10^{-502}$
α_{15}	$-0.35800731767935759271413335573 9 + 1.33934119631395854363390531468 9i$	$1.47520028506438597430701484010 1 \times 10^{-501}$
α_{16}	$-0.35800731767935759271413335573 9 - 1.33934119631395854363390531468 9i$	$1.47520028506438597430701484010 1 \times 10^{-501}$
α_{17}	$0.43800352221106812944529419880 6 + 1.38305205298853478724354390756 6i$	$3.53697023995907420437104320050 3 \times 10^{-500}$
α_{18}	$0.43800352221106812944529419880 6 - 1.38305205298853478724354390756 6i$	$3.53697023995907420437104320050 3 \times 10^{-500}$
α_{19}	$0.44626620989655218363997135156 9 + 1.45185557348197068406922848409 0i$	$1.63478475677429401407865314019 4 \times 10^{-498}$
α_{20}	$0.44626620989655218363997135156 9 - 1.45185557348197068406922848409 0i$	$1.63478475677429401407865314019 4 \times 10^{-498}$
α_{21}	$-1.42273656104976573954492144525 3 + 0.59846960117845132224126974907 4i$	$1.71806369600510840489975422097 6 \times 10^{-497}$
α_{22}	$-1.42273656104976573954492144525 3 - 0.59846960117845132224126974907 4i$	$1.71806369600510840489975422097 6 \times 10^{-497}$
α_{23}	$1.46997518259508558406407458391 7 + 0.54368484460937313698467152761 5i$	$9.98680840235548577485426962966 1 \times 10^{-497}$
α_{24}	$1.46997518259508558406407458391 7 - 0.54368484460937313698467152761 5i$	$9.98680840235548577485426962966 1 \times 10^{-497}$
α_{25}	$0.15423271459165792482309585257 8 + 1.56211568397756971010302697503 9i$	$7.14716947196322971951543790124 6 \times 10^{-497}$
α_{26}	$0.15423271459165792482309585257 8 - 1.56211568397756971010302697503 9i$	$7.14716947196322971951543790124 6 \times 10^{-497}$
α_{27}	$0.80776234183565119814635883512 5 + 1.34656359358446959717247942584 0i$	$5.11921600919756630956214908515 9 \times 10^{-497}$
α_{28}	$0.80776234183565119814635883512 5 - 1.34656359358446959717247942584 0i$	$5.11921600919756630956214908515 9 \times 10^{-497}$
α_{29}	$-0.92647781372346619608234434238 4 + 1.29996208618752291065875053087 4i$	$2.42505915791821204316342353956 3 \times 10^{-496}$
α_{30}	$-0.92647781372346619608234434238 4 - 1.29996208618752291065875053087 4i$	$2.42505915791821204316342353956 3 \times 10^{-496}$
α_{31}	$-0.37404466199551297085624996276 2 + 1.58126871263344606627680382547 0i$	$8.50433628691553061901992945514 4 \times 10^{-496}$
α_{32}	$-0.37404466199551297085624996276 2 - 1.58126871263344606627680382547 0i$	$8.50433628691553061901992945514 4 \times 10^{-496}$
α_{33}	$-1.61926086167381119011516617005 6 + 0.42416096904312874083677773792 3i$	$1.65424484549970607811075098711 1 \times 10^{-494}$
α_{34}	$-1.61926086167381119011516617005 6 - 0.42416096904312874083677773792 3i$	$1.65424484549970607811075098711 1 \times 10^{-494}$
α_{35}	$0.87075102701897787718255451770 6 + 1.50345169410320037147907435547 4i$	$4.30234771786622577357848993973 8 \times 10^{-493}$
α_{36}	$0.87075102701897787718255451770 6 - 1.50345169410320037147907435547 4i$	$4.30234771786622577357848993973 8 \times 10^{-493}$
α_{37}	$-1.59995629122353151603943092522 9 + 0.68054727682269480414926008680 3i$	$3.94927331126442632020193175063 9 \times 10^{-493}$
α_{38}	$-1.59995629122353151603943092522 9 - 0.68054727682269480414926008680 3i$	$3.94927331126442632020193175063 9 \times 10^{-493}$
α_{39}	$1.64683702992901672985041266176 8 + 0.62806679988828960318671885037 8i$	$4.68136845199659191874699263861 6 \times 10^{-492}$
α_{40}	$1.64683702992901672985041266176 8 - 0.62806679988828960318671885037 8i$	$4.68136845199659191874699263861 6 \times 10^{-492}$
α_{41}	$0.19835515025133233017831402251 9 + 1.76069222417112952948223627659 6i$	$2.00524700201092920084131436845 6 \times 10^{-492}$
α_{42}	$0.19835515025133233017831402251 9 - 1.76069222417112952948223627659 6i$	$2.00524700201092920084131436845 6 \times 10^{-492}$
α_{43}	$1.74542678761996476599869340807 4 + 0.48450531086260257496352630951 6i$	$2.75921547859374689501711150150 8 \times 10^{-491}$
α_{44}	$1.74542678761996476599869340807 4 - 0.48450531086260257496352630951 6i$	$2.75921547859374689501711150150 8 \times 10^{-491}$
α_{45}	$1.26530514772314700395912852595 1 + 1.29797607476015249046755940763 6i$	$3.49127360930809421454500389604 9 \times 10^{-491}$
α_{46}	$1.26530514772314700395912852595 1 - 1.29797607476015249046755940763 6i$	$3.49127360930809421454500389604 9 \times 10^{-490}$
α_{47}	$-0.39349569014063929172343214249 8 + 1.78557761353849486081646468732 4i$	$1.12929760840238893748417254302 3 \times 10^{-490}$
α_{48}	$-0.39349569014063929172343214249 8 - 1.78557761353849486081646468732 4i$	$1.12929760840238893748417254302 3 \times 10^{-490}$
α_{49}	$-1.05645657080746180500908881901 6 + 1.52297347077364220988712315065 5i$	$6.43249667469844560487778202209 7 \times 10^{-490}$
α_{50}	$-1.05645657080746180500908881901 6 - 1.52297347077364220988712315065 5i$	$6.43249667469844560487778202209 7 \times 10^{-490}$
α_{51}	$-1.79526831845126509747062155195 6 + 0.46990341685617789095857181221 2i$	$1.06835212802161921274663211180 0 \times 10^{-490}$
α_{52}	$-1.79526831845126509747062155195 6 - 0.46990341685617789095857181221 2i$	$1.06835212802161921274663211180 0 \times 10^{-490}$
α_{53}	$1.88181959861084849803322246496 6 + 0.50809566841869064696273712851 8i$	$2.14246833270689918803090659096 6 \times 10^{-488}$
α_{54}	$1.88181959861084849803322246496 6 - 0.50809566841869064696273712851 8i$	$2.14246833270689918803090659096 6 \times 10^{-488}$
α_{55}	$1.35223237822443724500670990431 4 + 1.41823889864913495713301110496 8i$	$2.12606178567849248859953121840 9 \times 10^{-487}$
α_{56}	$1.35223237822443724500670990431 4 - 1.41823889864913495713301110496 8i$	$2.12606178567849248859953121840 9 \times 10^{-487}$
α_{57}	$-1.15373893888774089528471929353 0 + 1.77839221875773126873443877813 7i$	$2.80256144115665282513996214652 6 \times 10^{-484}$
α_{58}	$-1.15373893888774089528471929353 0 - 1.77839221875773126873443877813 7i$	$2.80256144115665282513996214652 6 \times 10^{-484}$
α_{59}	$-1.19999615766156162493161793448 9 + 1.75912196876379702519734788590 3i$	$4.37046030643819598027066460399 5 \times 10^{-484}$
α_{60}	$-1.19999615766156162493161793448 9 - 1.75912196876379702519734788590 3i$	$4.37046030643819598027066460399 5 \times 10^{-484}$

5. Conclusion

The main contribution of this paper is the derivation of analytical results to evaluate the steady-state system-length distributions at random, pre-arrival and post-departure epochs of the $BMAP/BMSP/1$ queue. Analysis is based on the roots of the associated characteristic equation of the vector-generating function of system-length distribution at random epoch. Further, we have demonstrated a comprehensive analysis of the system-length distribution at random epoch using the matrix-geometric method. We have also obtained the sojourn-time distribution of an arbitrary customer in an arriving batch. We have carried out some relevant performance measures which will be beneficial to practitioners for modeling of complex communication systems such as traffic modeling of IP networks. We have performed extensive computational works with the diversified inputs and they are appended in the forms of tables and graphs. Finally, a comparative study is also carried out numerically to compare the roots method with that of the matrix-geometric method.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

References

[1] Abate, J., Choudhury, G. L., & Whitt, W. (1994). Asymptotics for steady-state tail probabilities in structured Markov queueing models. *Stochastic Models*, 10(1), 99-143.

[2] Alexander, O. (2013). Performance of Communication Systems: A Model-Based Approach with Matrix-geometric Methods. Springer Science & Business Media, Berlin.

[3] Alfa, A. S., Xue, J., & Ye, Q. (2000). Perturbation theory for the asymptotic decay rates in the queues with Markovian arrival process and/or Markovian service process. *Queueing Systems*, 36(4), 287-301.

[4] Banik, A. D. (2015). Single server queues with a batch markovian arrival process and bulk renewal or non-renewal service. *Journal of Systems Science and Systems Engineering*, 24(3), 337-363.

[5] Bank, B., & Samanta, S. K. (2020). Analytical and computational studies of the $BMAP/G^{(a,Y)}/1$ queue. *Communications in Statistics - Theory and Methods*, doi:10.1080/03610926.2019.1708941.

[6] Benzi, M., Bini, D., Kressner, D., Munthe-Kaas, H., & Van Loan, C. (2015). Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications. Springer, Cetraro.

[7] Buchholz, P., & Kriege, J. (2017). Fitting correlated arrival and service times and related queueing performance. *Queueing Systems*, 85(3-4), 337-359.

[8] Chakravarthy, S. (2001). The batch Markovian arrival process: a review and future work, in: A. Krishnamoorthy (Ed.). Advances in Probability Theory and Stochastic Processes, Notable Publications, 21-49. New Jersey, USA.

[9] Gail, H. R., Hantler, S. L., & Taylor, B. A. (1996). Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains. *Advances in Applied Probability*, 28(1), 114-165.

[10] Gold, H., & Tran-Gia, P. (1993). Performance analysis of a batch service queue arising out of manufacturing system modelling. *Queueing Systems*, 14(3-4), 413-426.

[11] Gupta, U. C., Singh, G., & Chaudhry, M. L. (2016). An alternative method for computing system-length distributions of $BMAP/R/1$ and $BMAP/D/1$ queues using roots. *Performance Evaluation*, 95, 60-79.

[12] He, Q. M. (2013). Fundamentals of Matrix-Analytic Methods. Springer, New York.

[13] Heyman, D. P., & Lucantoni D. (2003). Modeling multiple IP traffic streams with rate limits. *IEEE/ACM Transactions on Networking*, 11(6), 948-958.

[14] Horváth, G. (2016). Analysis of generalized QBD queues with matrix-geometrically distributed batch arrivals and services. *Queueing Systems*, 82, 353-380.

[15] Horváth, A., Horváth, G., & Telek, M. (2010). A joint moments based analysis of networks of $MAP/MAP/1$ queues. *Performance Evaluation*, 67(9), 759-778.

[16] Kim, N. K., Chang, S. H., & Chae, K. C. (2002). On the relationships among queue lengths at arrival, departure, and random epochs in the discrete-time queue with D-BMAP arrivals. *Operations Research Letters*, 30(1), 25-32.

[17] Kim, C., Dudin, S., Dudin, A., & Samouylov, K. (2019). Analysis of a semi-open queuing network with a state dependent marked Markovian arrival process, customers retrials and impatience. *Mathematics*, 7(8), 1-19.

[18] Klemm, A., Lindemann, C., & Lohmann, M. (2001). *Traffic modeling and characterization for UMTS networks*. In GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270), 1741-1746.

[19] Klemm, A., Lindemann, C., & Lohmann, M. (2003). Modeling IP traffic using the batch Markovian arrival process. *Performance Evaluation*, 54(2):,149-173.

[20] Liu, B., Cui, L., Wen, Y., & Shen, J. (2015). A cold standby repairable system with working vacations and vacation interruption following Markovian arrival process. *Reliability Engineering & System Safety*, 142: 1-8.

[21] Lucantoni, D. M. (1991). New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7(1), 1-46.

[22] Lucantoni, D. M. (1993). The $BMAP/G/1$ queue: a tutorial, in: L. Donatiello, R. Nelson (Eds.), Models and Techniques for Performance Evaluation of Computer and Communications Systems, 330-358. *Springer-Verlag*.

[23] Neuts, M. F. (1979). A versatile Markovian point process. *Journal of Applied Probability*, 16(4), 764-779.

[24] Neuts, M. F. (1981). Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach. The John Hopkins University Press, Baltimore.

[25] Neuts, M. F. (1989). Structured Stochastic Matrices of $M/G/1$ type and their Applications. NY: CRC Press, Marcel Dekker, New York.

[26] Ozawa, T. (2006). Sojourn time distributions in the queue defined by a general QBD process. *Queueing Systems*, 53(4), 203-211.

[27] Samanta, S. K. (2020). Waiting-time analysis of $D\text{-}BMAP/G/1$ queueing system. *Annals of Operations Research*, 284(1), 401-413.

[28] Samanta, S. K., Chaudhry, M. L., & Pacheco, A., (2016). Analysis of $BMAP/MSP/1$ queue. *Methodology and Computing in Applied Probability*, 18, 419-440.

[29] Sandhya, R., Sundar, V., Rama, G., Ramshankar, R., & Ramanarayanan, R. (2015). $BMAP/BMSP/1$ queue with randomly varying environment. *ISOR Journal of Engineering*, 5(4), 1-12.

[30] Shanthikumar, J. G., Ding, S., & Zhang, M. T. (2007). Queueing theory for semiconductor manufacturing systems: a survey and open problems. *IEEE Transactions on Automation Science and Engineering*, 4(4), 513-522.

[31] Vishnevskii, V. M., & Dudin, A. N. (2017). Queueing systems with correlated arrival flows and their applications to modeling telecommunication networks. *Automation and Remote Control*, 78(8), 1361-1403.

[32] Vishnevskii, V. M., Semenova, O. V., & Sharov, S. Y. (2013). Modeling and analysis of a hybrid communication channel based on free-space optical and radio-frequency technologies. *Automation and Remote Control*, 74, 521-528.

[33] Vishnevsky, V., Krishnamoorthy, A., Kozyrev, D., & Larionov, A. (2016). Review of methodology and design of broadband wireless networks with linear topology. *Indian Journal of Pure and Applied Mathematics*, 47(2). 329-342.

[34] Wang, Y. C., Cai, D. L., Chiang, L. H., & Hu, C. W. (2014). Elucidating the short term loss behavior of Markovian-modulated batch-service queueing model with discrete-time batch Markovian arrival process. *Mathematical Problems in Engineering*, 2014, 1-10.

[35] Wang, Y. C., Chou, J. H., & Wang, S. Y. (2011). Loss pattern of $DBMAP/DMSP/1/K$ queue and its application in wireless local communications. *Applied Mathematical Modelling*, 35(4), 1782-1797.

[36] Zhang, Q., Heindl, A., & Smirni, E. (2005). Characterizing the $BMAP/MAP/1$ departure process via the ETAQA truncation. *Stochastic Models*, 21(2-3), 821-846.