Hindawi Publishing Corporation
Journal of Probability and Statistics
Volume 2009, Article ID 716364, 18 pages
doi:10.1155/2009/716364

Research Article

Computational Procedures for a Class of GI/D/k Systems in Discrete Time

Md. Mostafizur Rahman and Attahiru Sule Alfa

Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada R3T 5V6

Correspondence should be addressed to Md. Mostafizur Rahman, mmrahman@ee.umanitoba.ca

Received 30 April 2009; Accepted 31 August 2009

Recommended by Nikolaos E. Limnios

A class of discrete time GI/D/k systems is considered for which the interarrival times have finite support and customers are served in first-in first-out (FIFO) order. The system is formulated as a single server queue with new general independent interarrival times and constant service duration by assuming cyclic assignment of customers to the identical servers. Then the queue length is set up as a quasi-birth-death (QBD) type Markov chain. It is shown that this transformed GI/D/1 system has special structures which make the computation of the matrix **R** simple and efficient, thereby reducing the number of multiplications in each iteration significantly. As a result we were able to keep the computation time very low. Moreover, use of the resulting structural properties makes the computation of the distribution of queue length of the transformed system efficient. The computation of the distribution of waiting time is also shown to be simple by exploiting the special structures.

1. Introduction

In most communication systems studied in discrete time, the interarrival times of packets usually follow independent general distribution. Hence, this arrival of packets in an ATM switch can be represented as independent general distribution. On the other hand, multiple packets (53 byte cells) are transmitted simultaneously through identical transmission lines in an ATM switch. Therefore, an ATM switch can be considered as a multiserver system with deterministic service time as the packets are of same size. Hence, the performance of an ATM switch can be analyzed by studying a GI/D/k system with finite support on interarrival times.

To the best of our knowledge GI/D/k systems in discrete time have never been analyzed with focus on computational efficiency in the literature. In this paper, matrix-geometric method is used to analyze a class of GI/D/k systems in discrete time with focus

on computational efficiency. The class of GI/D/k systems considered in this paper has finite support for the interarrival times and the service is first-in first-out (FIFO) order. The idea used here to analyze the waiting time distribution of this multiserver system is due to Crommelin [1]. This idea is also used in analyzing the distribution of waiting time of multiserver systems with constant service time duration for both continuous and discrete times such as [2–20]. The idea is that the assignment of customers to servers in cyclic order does not change the distribution of waiting time in a multiserver system where customers are served in FIFO order in the identical servers with same constant service time. In this case, only an arbitrary server can be studied to compute the waiting time distribution. Crommelin [1] analyzed the waiting time distribution of an M/D/c system by converting the system to an E_c/D/1 system. The work of Crommelin [1] is followed by some studies on deterministic service time multiserver systems in both continuous time and discrete time. Iversen [15] decomposed an M/D/rk queue with FIFO into E_k/D/r queues with FIFO for analyzing waiting time distribution. Franx [8] developed expression for the waiting time distribution of the M/D/c queue by a full probabilistic analysis requiring neither generating functions nor Laplace transforms. Wittevrongel and Bruneel [19] used transform-based method to analyze multiserver system with constant service time and correlated arrival process. Nishimura [17] studied a MAP/D/N system in continuous time using spectral technique. Takine [18] analyzed a MAP/D/s system and computed the distribution of queue length by first characterizing the sojourn time distribution. Kim and Chaudhry [21] also computed the distribution of queue length of a discrete time GI/D/c queue after computing the distribution of waiting time by using distributional Little's law and transform-based method. Exact elapsed time from the time of last arrival is approximated with time average in [21]. Chaudhry et al. [7] analyzed the distribution of waiting time of a MAP/D/k system in discrete time. Later Alfa [3] gave a simple and more efficient computational scheme for the MAP/D/k system in discrete time. Alfa [2] also carried out algorithmic analysis for a discrete time BMAP/D/k system by exploiting its structural properties. Chaudhry et al. [7] and Alfa [2, 3] used matrix geometric method [22] for analysis. There is some work on the GI/D/c queues such as Wuyts and Bruneel [4] and Gao et al. [10–14] which used transform-based method. There are other algorithms for multiserver queues with constant service times such as GI/D/1 and GI/D/c queues by Chaudhry [23], GI^X/D/c queue by Chaudhry and Kim [5], M^X/D/c queue by Chaudhry et al. [6] and Franx [9], Ph/D/c queues by van Hoorn [16], and an E_k/D/r queue by Xerocostas and Demertzes [20].

In this paper, the cyclic assignment of customers to servers is used to model a class of GI/D/k systems in discrete time as a single server system with the assumption of first-in first-out (FIFO) service order. The modeling as single server system sets up the distribution of queue length as a quasi-birth-death (QBD) which has some structural properties. Analysis of the GI/D/k system is carried out efficiently by exploiting these structural properties in this paper. This paper has three contributions—reductions in the computational complexities of (i) the matrix $\mathbf{R}$, (ii) the distribution of queue length of the transformed system, and (iii) the distribution of waiting time. The first contribution is that the time complexity of the computation of the matrix $\mathbf{R}$ is reduced by decreasing the number of multiplications in each iteration from $O(n^3k^3d^3)$ to $O(n^3kd^2)$ while requiring the same number of iteration as the natural iteration. Here, n is the support of general interarrival times, k is the number of servers, and d is the duration of service. The second contribution is that the distribution of queue length of the transformed system is computed efficiently by exploiting the special structures of the system. The third and most important contribution is that the computation of the distribution of waiting time is simplified. Chaudhry et al. [7] and Alfa [2, 3] computed

the distribution of waiting time using the technique for phase type service time distribution. However, it is shown in this paper that the computation of the distribution of waiting time for deterministic service time distribution in discrete time does not need the complicated steps of phase type service time distribution.

The rest of the paper is organized as follows. Section 2 introduces the GI/D/k system and explains the modeling of this multiserver system into a single server system. Section 3 discusses the special structures of the matrices of our model and exploitation of those structures for efficient computation of the matrix $\mathbf{R}$. The computation of the distributions of queue length and waiting time is explained in Sections 4 and 5, respectively. Some numerical examples are provided in Section 6 to show the suitability of our proposed method to compute the matrix $\mathbf{R}$. Section 7 concludes the paper.

2. The GI/D/k System

The class of GI/D/k system in discrete time considered here has finite support for interarrival time which is of general distribution. Moreover, the customers are served in FIFO service order in the identical servers which have constant service time.

The interarrival times, A, of this multiserver system are general, independent, and identically distributed (iid) with distribution $a_v = \Pr\{A = v\}$, $v = 1, 2, \ldots$.

We let $\lambda = 1/E[A]$ be the mean arrival rate of customers to the system with $0 < \lambda \leq 1$. The interarrival times can be represented as a Markov chain where each state represents the elapsed time using the approach followed in the analysis of discrete time GI/G/1 and GI^X/G/1 systems by Alfa [24]. General independent interarrival times can be represented as an absorbing Markov chain with a transition matrix $\mathbf{P_A} = \begin{bmatrix} \mathbf{T} & \mathbf{t} \\ \mathbf{0}_{1,n} & 1 \end{bmatrix}$ where $\mathbf{T}$ is a square matrix of order n, $\mathbf{t}$ is a column vector of order n, $\mathbf{t} = \mathbf{1}_n - \mathbf{T}\mathbf{1}_n$, $\mathbf{0}_{i,j}$ is an $i \times j$ matrix of zeros, and $\mathbf{1}_i$ is a column vector of ones of order i. The case of infinite interarrival times can be captured for $n = \infty$. However, the interarrival times are finite if data is collected from practical systems. Therefore, general independent interarrival times with finite support ($n < \infty$) are considered in this paper. The probabilities of interarrival times can be represented by a vector $\underline{\mathbf{a}} = [a_1, a_2, \ldots, a_n]$. Here, $\underline{\mathbf{a}}\mathbf{1}_n = 1$ and $a_0 = 0$. This general arrival process can be defined by phase type distribution $(\boldsymbol{\alpha}, \mathbf{T})$ of dimension n in elapsed time representation. The parameters $\boldsymbol{\alpha}$, $\mathbf{T}$, and $\mathbf{t}$ are given as $\boldsymbol{\alpha} = [1, 0, 0 \ldots, 0]$, $\mathbf{T}_{i,j} = (1 - \sum_{j=0}^{i} a_j)/(1 - \sum_{v=0}^{i-1} a_v) = \tilde{a}_i$, for $1 \leq i < n$, $j = i + 1$, $\mathbf{T}_{i,j} = 0$, for all $j \neq i + 1$ and $\mathbf{t} = [\tilde{\tilde{a}}_1 \;\; \tilde{\tilde{a}}_2 \;\; \cdots \;\; \tilde{\tilde{a}}_{n-1} \;\; 1]'$ where $\tilde{\tilde{a}}_i = 1 - \tilde{a}_i$ for $1 \leq i \leq n - 1$ and $\mathbf{W}'$ is the transpose of matrix $\mathbf{W}$.

The general independent arrival process can be represented by two square matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ of order n where $(\mathbf{D}_l)_{ij}$ ($l = 0, 1$ and $1 \leq i, j \leq n$) represents a transition from phase i to phase j with l arrivals. Now, $\mathbf{D}_0$ and $\mathbf{D}_1$ can be represented in terms of phase type distribution $(\boldsymbol{\alpha}, \mathbf{T})$. Here, $\mathbf{D}_0 = \mathbf{T}$ and $\mathbf{D}_1 = \mathbf{t}\boldsymbol{\alpha} = [\mathbf{t}\ \mathbf{0}_{n,n-1}]$. The first column of $\mathbf{D}_1$ is nonzero and this column is the vector $\mathbf{t}$. The matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ are similar to the matrices of zero or one arrival for MAP distributed interarrival times. However, they are special cases because the matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ only capture general and independently distributed interarrival times.

A discrete time Markov chain $\{(N(t), J(t)); t = 0, 1, 2, \ldots; N(t) \in \{0, 1, 2, \ldots\}; J(t) \in \{1, 2, 3, \ldots, n\}\}$ can be considered to represent this general arrival process where $N(t)$ is the number of arrivals up to and including time t and $J(t)$ is the phase of the next arrival at time t.

The matrix $\mathbf{D} = \mathbf{D}_0 + \mathbf{D}_1$ is stochastic. If $\tilde{\pi} = [\tilde{\pi}_1, \tilde{\pi}_2, \ldots, \tilde{\pi}_n]$ is the invariant probability vector for arrival in n different phases then $\tilde{\pi} = \tilde{\pi}\mathbf{D}$ and $\tilde{\pi}\mathbf{1}_n = 1$. The arrival rate is given by $\lambda = \tilde{\pi}\mathbf{D}_1\mathbf{1}_n = \tilde{\pi}\mathbf{t}$.

There are k identical servers in a GI/D/k system. The service times are of constant duration d units ($d \geq 1$). Service time can be represented as phase type distribution in elapsed time form with representation $(\beta, \mathbf{S})$ of dimension d where $\beta = [1, 0, 0, \ldots, 0]$ and $\mathbf{S} = \begin{bmatrix} \mathbf{0}_{d-1,1} & \mathbf{I}_{d-1} \\ 0 & \mathbf{0}_{1,d-1} \end{bmatrix}$, $\mathbf{I}_j$ is an identity matrix of order j. Another column vector $\mathbf{s} = \mathbf{1}_d - \mathbf{S}\mathbf{1}_d$ can be defined for this phase type distribution $(\beta, \mathbf{S})$.

The condition that ensures the stability of GI/D/k system is $\rho < 1$ where $\rho = \lambda d/k$. We assume that the system is stable (i.e., $\rho < 1$).

2.1. The Model of Arrivals to an Arbitrary Server

Let us assume that customers are served in first-in first-out (FIFO) order in this GI/D/k system. It can also be assumed that the servers are assigned with customers in a cyclic manner particularly during the idle times. This assignment of customers to servers does not affect the waiting times experienced by the customers as the servers are identical. Using the same approach of Alfa [2, 3] we can consider the case of the jth ($1 \leq j \leq k$) server. In this approach if the first customer is assigned to the first server then the jth server will be assigned with $j, j + k, j + 2k, \ldots$th customers. In this case we only need to study one server, an arbitrary server j ($1 \leq j \leq k$). This arbitrary server sees the arrival of customers to be according to another independent general process which we call ${}_k$GI. This ${}_k$GI arrival process is a k-fold convolution of original arrival process to the multiserver system. ${}_k$GI can be defined by two square matrices $\mathbf{C}_0$ and $\mathbf{C}_1$ of order nk. The element $(\mathbf{C}_v)_{ij}$ ($v = 0, 1$ and $1 \leq i, j \leq nk$) represents transition from phase i to phase j with v arrivals. Both $\mathbf{C}_0$ and $\mathbf{C}_1$ can be considered to comprise of k^2 square block matrices of order n. In this representation $(\mathbf{C}_0)_{i,j} = \mathbf{D}_0$ for $i = j, 1 \leq i \leq k$, $(\mathbf{C}_0)_{i,j} = \mathbf{D}_1$ for $j = i + 1, 1 \leq i < k$ and $(\mathbf{C}_0)_{i,j} = \mathbf{0}_{n,n}$ for $1 \leq i \leq k$ and $i \neq j$ or $j \neq i + 1$. Similarly $(\mathbf{C}_1)_{i,j} = \mathbf{D}_1$ for $i = k,\ j = 1$ and $(\mathbf{C}_1)_{i,j} = \mathbf{0}_{n,n}$ in all other cases for $1 \leq i,\ j \leq k$. The first column and the last row of $\mathbf{C}_0$ are zero as the first column and last row of $\mathbf{D}_0$ are zero. Only the last n rows of $\mathbf{C}_1$ are nonzero and only the first element of each of these n rows is nonzero.

The matrices $\mathbf{C}_0$ and $\mathbf{C}_1$ exhibit behavior similar to the matrices $\mathbf{D}_0$ and $\mathbf{D}_1$ except that $\mathbf{C}_v$ ($v = 0, 1$) is a new matrix representing the arrival of v supercustomers where a supercustomer is the last customer to form a group of k customers or the arrival of v regular customers to an arbitrary server. On the other hand, D_v ($v = 0, 1$) represents arrival of v regular customers to the GI/D/k system. Now, this new arrival process ${}_k$GI to an arbitrary server can be represented by a discrete time Markov chain $\{(\widehat{N}(t), \widehat{J}(t)); t = 0, 1, 2, \ldots;\ \widehat{N}(t) \in \{0, 1, 2, \ldots\};\ \widehat{J}(t) \in \{1, 2, \ldots, nk\}\}$ where $\widehat{N}(t)$ is the number of supercustomers that have arrived by time t and $\widehat{J}(t)$ is the phase of the next arrival at time t.

The matrix $\mathbf{C} = \mathbf{C}_0 + \mathbf{C}_1$ is stochastic. If δ, a $1 \times nk$ matrix, is the invariant probability vector for arrival in different nk phases then $\delta = \delta\mathbf{C}$ and $\delta\mathbf{1}_{nk} = 1$. δ can also be represented as $\delta = [\delta_1, \delta_2, \delta_3, \ldots, \delta_k]$ where δ_i ($1 \leq i \leq k$) is of size $1 \times n$. The supercustomer arrival rate is given by $\lambda^* = \lambda/k = \delta\mathbf{C}_1\mathbf{1}_{nk} = \delta_k\mathbf{t}$.

2.2. The New Model ${}_k$GI/D/1 System

A model can be developed for a single server queue with arrival ${}_k$GI and deterministic service time of d units following the techniques of Alfa [3]. This single server queue is called

$_k$GI/D/1 system and the resulting model is of QBD process type for which standard methods can be used [22]. Let us assume that L_t, J_t, and S_t represent the number of supercustomers, the arrival phase of the next supercustomer, and the elapsed time of the supercustomer in service, respectively, at time t in this QBD model. The state space can be represented by $\{L_t, J_t, S_t\}$ for $L_t \geq 1$ and the state space is $\{0, J_t\}$ for $L_t = 0$. The total state space is $\{(0, j) \cup (i, j, l) : i \geq 1, 1 \leq j \leq nk, 1 \leq l \leq d\}$. The transition matrix for this $_k$GI/D/1 system is

$$\mathbf{P} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & & & \\ \mathbf{E} & \mathbf{A}_1 & \mathbf{A}_0 & & & \\ & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & & \\ & & \mathbf{A}_2 & \mathbf{A}_1 & \mathbf{A}_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix}, \tag{2.1}$$

where $\mathbf{B}_0 = \mathbf{C}_0$, $\mathbf{B}_1 = \mathbf{C}_1 \otimes \beta$, $\mathbf{E} = \mathbf{C}_0 \otimes \mathbf{s}$, $\mathbf{A}_2 = \mathbf{C}_0 \otimes (\mathbf{s}\beta)$, $\mathbf{A}_1 = \mathbf{C}_0 \otimes \mathbf{S} + \mathbf{C}_1 \otimes (\mathbf{s}\beta)$, and $\mathbf{A}_0 = \mathbf{C}_1 \otimes \mathbf{S}$.

Let $\mathbf{x} = [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots]$ be the stationary distribution of $\mathbf{P}$ where $\mathbf{x} = \mathbf{x}\mathbf{P}$, $\mathbf{x}_0\mathbf{1}_{nk} + \sum_{i=1}^{\infty} \mathbf{x}_i\mathbf{1}_{nkd} = 1$ and $\mathbf{x}_j$ $(j \geq 0)$ is the probability vector representing j supercustomers in the $_k$GI/D/1 system. During the computation of the stationary distribution of queue length $\mathbf{x}$, a matrix $\mathbf{R}$ is computed first which is followed by the computations of $\mathbf{x}_0$ and $\mathbf{x}_1$. Then $\mathbf{x}_i (i \geq 2)$ is computed using the matrix-geometric result $\mathbf{x}_i = \mathbf{x}_{i-1}\mathbf{R} = \mathbf{x}_1\mathbf{R}^{i-1}$. Here, the matrix $\mathbf{R}$ is of order $nkd \times nkd$ which is the minimal nonnegative solution of $\mathbf{R} = \mathbf{A}_0 + \mathbf{R}\mathbf{A}_1 + \mathbf{R}^2\mathbf{A}_2$. The entry $\mathbf{R}_{u,v}$ $(1 \leq u, v \leq nkd)$ of $\mathbf{R}$ represents the expected number of visits into $(m + 1, v)$, starting from (m, u), before the first return to level m $(m \geq 1)$. Here, m is the number of supercustomers in the system and u and v are any of the nkd phases of the system.

3. Structures of the Matrices

This section explains the structures of the block-matrices $\mathbf{B}_0$, $\mathbf{B}_1$, $\mathbf{E}$, $\mathbf{A}_2$, $\mathbf{A}_1$, and $\mathbf{A}_0$ and how their structures are exploited for efficient computation of the matrix $\mathbf{R}$. The structures of these matrices are also fully exploited in computing the stationary distribution of the queue length in Section 4.

Here, $\mathbf{B}_0$ and $\mathbf{B}_1$ are matrices of order $nk \times nk$ and $nk \times nkd$, respectively. The first column and the last row of the matrix $\mathbf{B}_0$ are zero. On the other hand, only the last n rows of the matrix $\mathbf{B}_1$ are nonzero and among these last n rows only the first column is nonzero which is $\mathbf{t}$. The matrix $\mathbf{B}_1$ can be represented as $\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}_{n(k-1),nd} & \mathbf{0}_{n(k-1),nd(k-1)} \\ \widehat{\mathbf{L}} & \mathbf{0}_{n,nd(k-1)} \end{bmatrix}$ where $\widehat{\mathbf{L}} = [\mathbf{t} \quad \mathbf{0}_{n,nd-1}]$.

The matrix $\mathbf{E}$ is of order $nkd \times nk$ which can be represented as

$$\mathbf{E} = \begin{bmatrix} \widetilde{\mathbf{L}} & \widetilde{\mathbf{M}} & & & \\ & \widetilde{\mathbf{L}} & \widetilde{\mathbf{M}} & & \\ & & \ddots & \ddots & \\ & & & \widetilde{\mathbf{L}} & \widetilde{\mathbf{M}} \\ & & & & \widetilde{\mathbf{L}} \end{bmatrix}, \tag{3.1}$$

where

$$\widetilde{\mathbf{L}} = \begin{bmatrix} \widetilde{a}_1\mathbf{s} & & & \\ & \widetilde{a}_2\mathbf{s} & & \\ & & \ddots & \\ & & & \widetilde{a}_{n-1}\mathbf{s} \end{bmatrix}, \qquad \widetilde{\widetilde{\mathbf{M}}} = \begin{bmatrix} \widetilde{\widetilde{a}}_1\mathbf{s} \\ \widetilde{\widetilde{a}}_2\mathbf{s} \\ \vdots \\ \widetilde{\widetilde{a}}_{n-1}\mathbf{s} \\ \mathbf{s} \end{bmatrix}. \tag{3.2}$$

$\widetilde{\mathbf{L}} = \mathbf{D}_0 \otimes \mathbf{s}$ is a matrix of order $nd \times n$. Moreover, $\widetilde{\mathbf{L}}$ has $(n-1)$ nonzero columns and each column contains only one nonzero element. $\widetilde{\mathbf{L}}$ has $(n-1)$ nonzero elements in $(jd, j+1)$th $(1 \le j \le n-1)$ positions. $\widetilde{\widetilde{\mathbf{M}}} = \mathbf{D}_1 \otimes \mathbf{s}$ is also a matrix of order $nd \times n$ matrix. Furthermore, only the first column of $\widetilde{\widetilde{\mathbf{M}}}$ is nonzero which contains n nonzero elements. $\widetilde{\widetilde{\mathbf{M}}}$ has nonzero elements in $(jd, 1)$th $(1 \le j \le n)$ positions.

The matrices $\mathbf{A}_2$, $\mathbf{A}_1$, and $\mathbf{A}_0$ are of size $nkd \times nkd$. The matrix $\mathbf{A}_2$ can be represented as

$$\mathbf{A}_2 = \begin{bmatrix} \widetilde{\mathbf{L}}_1 & \widetilde{\widetilde{\mathbf{M}}}_1 & & & \\ & \widetilde{\mathbf{L}}_1 & \widetilde{\widetilde{\mathbf{M}}}_1 & & \\ & & \ddots & \ddots & \\ & & & \widetilde{\mathbf{L}}_1 & \widetilde{\widetilde{\mathbf{M}}}_1 \\ & & & & \widetilde{\mathbf{L}}_1 \end{bmatrix}, \tag{3.3}$$

where $\widetilde{\mathbf{L}}_1$ and $\widetilde{\widetilde{\mathbf{M}}}_1$ are both matrices of order $nd \times nd$, $\widetilde{\mathbf{L}}_1 = \mathbf{D}_0 \otimes (\mathbf{s}\beta)$ and $\widetilde{\widetilde{\mathbf{M}}}_1 = \mathbf{D}_1 \otimes (\mathbf{s}\beta)$. Hence,

$$\widetilde{\mathbf{L}}_1 = \begin{bmatrix} \widetilde{a}_1(\mathbf{s}\beta) & & & \\ & \widetilde{a}_2(\mathbf{s}\beta) & & \\ & & \ddots & \\ & & & \widetilde{a}_{n-1}(\mathbf{s}\beta) \end{bmatrix}, \qquad \widetilde{\widetilde{\mathbf{M}}}_1 = \begin{bmatrix} \widetilde{\widetilde{a}}_1(\mathbf{s}\beta) \\ \widetilde{\widetilde{a}}_2(\mathbf{s}\beta) \\ \vdots \\ \widetilde{\widetilde{a}}_{n-1}(\mathbf{s}\beta) \\ (\mathbf{s}\beta) \end{bmatrix}. \tag{3.4}$$

There are $(n-1)$ nonzero columns in $\widetilde{\mathbf{L}}_1$ and each column contains only one nonzero element. $\widetilde{\mathbf{L}}_1$ has these nonzero elements in $(jd, jd+1)$th $(1 \le j \le n-1)$ positions. Only the first column of $\widetilde{\widetilde{\mathbf{M}}}_1$ is nonzero with n nonzero elements in $(jd, 1)$th $(1 \le j \le n)$ positions.

The matrix $\mathbf{A}_1$ can be represented as

$$\mathbf{A}_1 = \begin{bmatrix} \widetilde{\mathbf{L}}_2 & \widetilde{\mathbf{M}}_2 & & & \\ & \widetilde{\mathbf{L}}_2 & \widetilde{\mathbf{M}}_2 & & \\ & & \ddots & \ddots & \\ & & & \widetilde{\mathbf{L}}_2 & \widetilde{\mathbf{M}}_2 \\ \widetilde{\mathbf{M}}_1 & & & & \widetilde{\mathbf{L}}_2 \end{bmatrix}, \tag{3.5}$$

where

$$\widetilde{\mathbf{L}}_2 = \begin{bmatrix} \widetilde{a}_1\mathbf{S} & & & \\ & \widetilde{a}_2\mathbf{S} & & \\ & & \ddots & \\ & & & \widetilde{a}_{n-1}\mathbf{S} \end{bmatrix}, \qquad \widetilde{\mathbf{M}}_2 = \begin{bmatrix} \widetilde{\widetilde{a}}_1\mathbf{S} \\ \widetilde{\widetilde{a}}_2\mathbf{S} \\ \vdots \\ \widetilde{\widetilde{a}}_{n-1}\mathbf{S} \\ \mathbf{S} \end{bmatrix}. \tag{3.6}$$

Here, $\widetilde{\mathbf{L}}_2$ and $\widetilde{\mathbf{M}}_2$ are both matrices of order $nd \times nd$. $\widetilde{\mathbf{L}}_2 = \mathbf{D}_0 \otimes \mathbf{S}$ and $\widetilde{\mathbf{M}}_2 = \mathbf{D}_1 \otimes \mathbf{S}$. There are $(n-1)(d-1)$ nonzero columns, $(n-1)(d-1)$ nonzero rows, and $(n-1)(d-1)$ nonzero elements in $\widetilde{\mathbf{L}}_2$. Moreover, each nonzero row or each nonzero column of $\widetilde{\mathbf{L}}_2$ contains only one nonzero element. On the other hand, there are $(d-1)$ nonzero columns in $\widetilde{\mathbf{M}}_2$ from column 2 to d each having n nonzero elements.

The matrix $\mathbf{A}_0$ can be represented as $\mathbf{A}_0 = \begin{bmatrix} \\ \widetilde{\mathbf{M}}_2 \end{bmatrix}$. $\mathbf{A}_0$ has only $n(d-1)$ nonzero rows which are in its last nd rows and $(d-1)$ nonzero columns from column 2 to column d.

3.1. Computation of the Matrix R

The matrix $\mathbf{R}$ can be computed using any of the three linearly convergent formulae from Neuts [22]—$\mathbf{R}(i+1) := \mathbf{A}_0 + \mathbf{R}(i)\mathbf{A}_1 + \mathbf{R}^2(i)\mathbf{A}_2$, $\mathbf{R}(i+1) := (\mathbf{A}_0 + \mathbf{R}^2(i)\mathbf{A}_2)(\mathbf{I}_{nkd} - \mathbf{A}_1)^{-1}$, and $\mathbf{R}(i+1) := \mathbf{A}_0(\mathbf{I}_{nkd} - \mathbf{A}_1 - \mathbf{R}(i)\mathbf{A}_2)^{-1}$ which are known as natural, traditional, and $\mathbf{U}$-based iterations, respectively. Here, $\mathbf{R}(i)$ is the ith iteration value of $\mathbf{R}$, $\mathbf{R}(0) := \mathbf{0}_{nkd,nkd}$, $\mathbf{U}$ is a square matrix of the same order as the matrix $\mathbf{R}$ and $\mathbf{U} = \mathbf{A}_1 + \mathbf{R}\mathbf{A}_2$. The matrix $\mathbf{U}$ is the minimal nonnegative solution of $\mathbf{U} = \mathbf{A}_1 + \mathbf{A}_0(\mathbf{I} - \mathbf{U})^{-1}\mathbf{A}_2$. The entry $\mathbf{U}_{u,v}$ $(1 \le u, v \le nkd)$ represents the taboo probability of starting from (m,u) for $m \ge 1$ the eventual visit of the Markov chain to level m by visiting (m,v) under the taboo of level $(m-1)$. Any of these three methods terminate in the ith $(i \ge 1)$ iteration for $\|\mathbf{R}(i) - \mathbf{R}(i-1)\|_\infty < \varepsilon$ where ε is a very small positive constant. In each step, natural itearation requires two matrix multiplications whereas traditional iteration requires three matrix multiplications and one matrix inversion. On the other hand, $\mathbf{U}$-based iteration requires two matrix multiplications and one matrix inversion in each step. Traditional iteration requires less number of iterations than natural iteration does while the number of iterations required for $\mathbf{U}$-based iteration is less than the numbers of iterations for other two methods. Alfa and Xue [25] developed efficient inversion

technique for **U**-based iteration for the matrix **R** in analyzing the queue length of a GI/G/1 system in discrete time.

There are several efficient methods for computing the matrix **R** for a QBD such as logarithmic reduction algorithm of Latouche and Ramaswami [26], cyclic reduction technique of Bini and Meini [27], and invariant subspace approach of Akar and Sohraby [28]. These quadratically convergent algorithms compute a matrix **G** which is the minimal nonnegative solution to $\mathbf{G} = \mathbf{A}_2 + \mathbf{A}_1\mathbf{G} + \mathbf{A}_0\mathbf{G}^2$. Here, **G** is an $m \times m$ matrix for $m \times m$ matrices $\mathbf{A}_2$, $\mathbf{A}_1$, and $\mathbf{A}_0$ where $G_{i,j}$ $(1 \le i, j \le m)$ represents the probability that starting from state $(N+1, i)$ the Markov chain eventually visits the level N $(N \ge 1)$ and does so by visiting the state (N, j). The matrices **R**, **U**, and **G** are related and these relations can be found in [26]. Similar to the computation of the matrix **R**, the matrix **G** can be computed using three different iterations—natural, traditional, and **U**-based. Generally a zero matrix is used for the initial value for iterations of the matrix **G** (i.e., $\mathbf{G} = \mathbf{0}_{m,m}$). Latouche and Ramaswami [26] showed that if $S_G(\varepsilon)$ iterative steps are needed to converge for a very small positive constant ε such that $\|\mathbf{G}(S_G(\varepsilon)) - \mathbf{G}(S_G(\varepsilon) - 1)\|_\infty < \varepsilon$ for the $(l+1)$-st natural iteration $\mathbf{G}(l+1) = \mathbf{A}_2 + \mathbf{A}_1\mathbf{G}(l) + \mathbf{A}_0\mathbf{G}^2(l)$, then logarithmic reduction requires not more than $\log_2 S_G(\varepsilon)$ iterations and overall complexity of their algorithm is $O(m^3 \log_2 S_G(\varepsilon))$. They also developed logarithmic reduction technique for the matrix **R**. However, the quadratic convergent algorithms [26–28] are not considered here for computing the matrix **R** as these algorithms involve inversion of matrices which cannot exploit the special structures of the matrices $\mathbf{A}_2$, $\mathbf{A}_1$, and $\mathbf{A}_0$.

The complexity of computing the matrix **R** can be reduced by using the structure of $\mathbf{A}_0$. The first $nd(k-1)$ rows are zero in $\mathbf{A}_0$. Therefore, the first $nd(k-1)$ rows are zero in **R**. Among the last nd rows of $\mathbf{A}_0$, n rows are zero which are rows $nd(k-1)+d$, $nd(k-1)+2d$, $nd(k-1)+3d, \ldots, nkd$. The corresponding rows are zero for **R**. The matrix **R** can be represented as follows:

$$\mathbf{R} = \begin{bmatrix} & & & & \\ \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 & \cdots & \mathbf{R}_k \end{bmatrix}. \tag{3.7}$$

Here, $\mathbf{R}_j$ $(1 \le j \le k)$ is a matrix of order $nd \times nd$ which contains $d, 2d, 3d, \ldots, nd$th rows as zero rows.

The computations of $\mathbf{R}(i)\mathbf{A}_2$ and $\mathbf{R}(i)\mathbf{A}_1$ require $(2k-1)$ and $2k$ block-matrix multiplications, respectively, where each block matrix is of order $nd \times nd$. Similarly, the computation of $\mathbf{R}^2(i)\mathbf{A}_2$ requires multiplication of $\mathbf{R}(i)$ and $\mathbf{R}(i)\mathbf{A}_2$ which involves k block-matrix multiplications. Inversions of both the matrices $(\mathbf{I}_{nkd} - \mathbf{A}_1)$ and $(\mathbf{I}_{nkd} - \mathbf{A}_1 - \mathbf{R}(i)\mathbf{A}_2)$ generate k^2 block matrices each of order $nd \times nd$. Inversion of the matrix $(\mathbf{I}_{nkd} - \mathbf{A}_1)$ needs to be computed only once for traditional iteration whereas inversion of the matrix $(\mathbf{I}_{nkd} - \mathbf{A}_1 - \mathbf{R}(i)\mathbf{A}_2)$ needs to be computed in each iteration for **U**-based iteration. On the other hand, the multiplication of $(\mathbf{A}_0 + \mathbf{R}^2(i)\mathbf{A}_2)$ and $(\mathbf{I}_{nkd} - \mathbf{A}_1)^{-1}$ in traditional iteration and the multiplication of $\mathbf{A}_0$ and $(\mathbf{I}_{nkd} - \mathbf{A}_1 - \mathbf{R}(i)\mathbf{A}_2)^{-1}$ in **U**-based iteration involve k^2 and k block-matrix multiplications, respectively, where each block matrix is of order $nd \times nd$. Each iteration of natural method requires substantially less multiplications than each step of traditional and **U**-based methods do. Therefore, we use natural iteration for developing our algorithm for computing the matrix **R** in this paper.

The matrix $\mathbf{R}_j$ $(1 \le j \le k)$ of (3.7) can be computed using the natural iteration with the following equation:

$$\mathbf{R}_j(i+1) := \begin{cases} \widetilde{\mathbf{M}}_2 + \mathbf{R}_1(i)\widetilde{\mathbf{L}}_2 + \mathbf{R}_k(i)\widetilde{\mathbf{M}}_1 + \mathbf{R}_k(i)\mathbf{R}_1(i)\widetilde{\mathbf{L}}_1, & j = 1, \\ \mathbf{R}_{j-1}(i)\widetilde{\mathbf{M}}_2 + \mathbf{R}_j(i)\widetilde{\mathbf{L}}_2 + \mathbf{R}_k(i)\mathbf{R}_{j-1}(i)\widetilde{\mathbf{M}}_1 + \mathbf{R}_k(i)\mathbf{R}_j(i)\widetilde{\mathbf{L}}_1, & 1 < j \le k, \end{cases} \tag{3.8}$$

where $\mathbf{R}_j(i)$ $(i \ge 0, 1 \le j \le k)$ is the ith iteration value of $\mathbf{R}_j$ with the initial value $\mathbf{R}_j(0) := \mathbf{0}_{nd,nd}(1 \le j \le k)$. The terminating condition for this iteration is same as the terminating condition for natural, traditional, and $\mathbf{U}$-based methods but needs to consider only $n(d-1)$ nonzero rows of $(\mathbf{R}(i) - \mathbf{R}(i-1))$.

The number of multiplications required to compute $\mathbf{R}_j(i)\widetilde{\mathbf{L}}_2$ $(1 \le j \le k)$ is $n(n-1)(d-1)^2$. Similarly, the number of multiplications required to compute both $\mathbf{R}_j(i)\widetilde{\mathbf{L}}_1$ and $\mathbf{R}_j(i)\widetilde{\mathbf{M}}_1$ for $1 \le j \le k$ is $n(n-1)(d-1)$. In the same way, the number of multiplications required to compute $\mathbf{R}_{j-1}(i)\widetilde{\mathbf{M}}_2(1 < j \le k)$ is $n(n-1)(d-1)^2$. On the other hand, the numbers of multiplications required to compute $\mathbf{R}_k(i)\mathbf{R}_{j-1}(i)\widetilde{\mathbf{M}}_1(1 < j \le k)$ by block-matrix multiplication of $\mathbf{R}_k(i)$ and $\mathbf{R}_{j-1}(i)\widetilde{\mathbf{M}}_1$ $(1 < j \le k)$ and $\mathbf{R}_k(i)\mathbf{R}_j(i)\widetilde{\mathbf{L}}_1(1 \le j \le k)$ by block-matrix multiplication of $\mathbf{R}_k(i)$ and $\mathbf{R}_j(i)\widetilde{\mathbf{L}}_1$ $(1 \le j \le k)$ are $n^2(d-1)^2$ and $n^2(n-1)(d-1)^2$, respectively. Hence, the numbers of multiplications required for computing $\mathbf{R}_1(i+1)$ from $\mathbf{R}_1(i)$ and $\mathbf{R}_j(i+1)$ $(2 \le j \le k)$ from $\mathbf{R}_j(i)$ in one iteration are $n(n-1)(d-1)(d+1) + n^2(n-1)(d-1)^2$ and $n^3(d-1)^2 + 2nd(n-1)(d-1)$, respectively. Therefore, the total number of multiplications required to compute all k block matrices $\mathbf{R}_j(i+1)$'s $(1 \le j \le k)$ of the matrix $\mathbf{R}(i+1)$ in one iteration is $n^2(n-1)(d-1)^2 + n(n-1)(d-1)(d+1) + (k-1)(n^3(d-1)^2 + 2nd(n-1)(d-1)) = O(n^3kd^2)$. By exploiting the special structures of the matrices the computational complexity is decreased in each iteration from $O(n^3k^3d^3)$ to $O(n^3kd^2)$ which is substantial reduction in computations. Moreover, the memory requirement for our method is also less than that of natural, traditional, $\mathbf{U}$-based methods and quadratic convergent algorithms.

4. Stationary Distribution of $\mathbf{P}$ for $_k$GI/D/1 System

Two methods for computing stationary distribution $\mathbf{x}$ for the Markov chain represented by the matrix $\mathbf{P}$ are briefly presented here which are followed by Section 4.1. Section 4.1 explains how the second method can be made efficient by utilizing the special structures of the matrices.

In the first method an invariant probability vector $\mathbf{z} = [\mathbf{z}_0, \mathbf{z}_1]$ needs to be computed for a stochastic matrix $\widetilde{\mathbf{P}} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 \\ \mathbf{E} & \mathbf{A}_1 + \mathbf{R}\mathbf{A}_2 \end{bmatrix}$. The row vector $\mathbf{z}$ is computed from $\mathbf{z}\widetilde{\mathbf{P}} = \mathbf{z}$ and $\mathbf{z}_0\mathbf{1}_{nk} + \mathbf{z}_1\mathbf{1}_{nkd} = 1$. Then $\mathbf{x}_0$ and $\mathbf{x}_1$ can be expressed as $\mathbf{x}_0 = v^{-1}\mathbf{z}_0$ and $\mathbf{x}_1 = v^{-1}\mathbf{z}_1$ where $v = \mathbf{z}_0\mathbf{1}_{nk} + \mathbf{z}_1(\mathbf{I}_{nkd} - \mathbf{R})^{-1}\mathbf{1}_{nkd}$.

In the second method, we need to compute $\widetilde{\mathbf{h}}$, the invariant probability vector of a stochastic matrix $\mathbf{H}$ defined as $\mathbf{H} = \mathbf{A}_1 + R\mathbf{A}_2 + \mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{B}_1$. Next a positive real number h is computed as follows $h = \widetilde{\mathbf{h}}(\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{1}_{nk} + (\mathbf{I}_{nkd} - \mathbf{R})^{-1}\mathbf{1}_{nkd})$. On the other hand, $\widetilde{\mathbf{h}}$ is computed using the relations $\widetilde{\mathbf{h}} = \widetilde{\mathbf{h}}\mathbf{H}$ and $\widetilde{\mathbf{h}}\mathbf{1}_{nkd} = 1$. $\mathbf{x}_0$ and $\mathbf{x}_1$ can be computed as $\mathbf{x}_1 = h^{-1}\widetilde{\mathbf{h}}$, $\mathbf{x}_0 = \mathbf{x}_1\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$.

4.1. Exploiting Structures for Stationary Distribution

The stationary distribution of $\mathbf{x}$ can be computed by first computing the probability vector $\mathbf{x}_1$ which can be made efficient by exploiting the special structures of the matrices $\mathbf{H}$, $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{1}_{nk}$ and $(\mathbf{I}_{nkd} - \mathbf{R})^{-1}\mathbf{1}_{nkd}$. The special structures of the matrices $(\mathbf{I}_{nkd} - \mathbf{R})^{-1}$, $(\mathbf{I}_{nkd} - \mathbf{R})^{-1}\mathbf{1}_{nkd}$, $(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$, $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$, $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{1}_{nk}$, $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{B}_1$, and $\mathbf{H}$ are due to the special structures of the matrices $\mathbf{B}_0$, $\mathbf{B}_1$, $\mathbf{E}$, and $\mathbf{R}$. Here, how the computations of these matrices can be carried out efficiently is explained.

Computing $(\mathbf{I}_{nkd} - \mathbf{R})^{-1} = \begin{bmatrix} \mathbf{I}_{nd(k-1)} & \mathbf{0}_{nd(k-1),nd} \\ \Phi & \Upsilon \end{bmatrix}$ requires $O(n^3kd^3)$ multiplications by using block-matrix inversion [29] where $\Upsilon = \tilde{\mathbf{R}}_k^{-1}$, $\Phi = \Upsilon\Psi$, $\Psi = [\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \ldots, \mathbf{R}_{k-1}]$, and $\tilde{\mathbf{R}}_k = \mathbf{I}_{nd} - \mathbf{R}_k$. Φ can be expressed as $\Phi = [\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_{k-1}]$ where $\Phi_j = \Upsilon\mathbf{R}_j$ $(1 \le j \le k-1)$.

$(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ can be represented as

$$(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 & \cdots & \mathbf{V}_{k-1} & \mathbf{V}_k \\ & \mathbf{V}_1 & \cdots & \mathbf{V}_{k-2} & \mathbf{V}_{k-1} \\ & & \ddots & \vdots & \vdots \\ & & & \mathbf{V}_1 & \mathbf{V}_2 \\ & & & & \mathbf{V}_1 \end{bmatrix}, \tag{4.1}$$

where $\mathbf{V}_1 = \tilde{\mathbf{D}}_0^{-1}$, $\mathbf{V}_i = (\tilde{\mathbf{D}}_0^{-1}\mathbf{D}_1)^{i-1}\tilde{\mathbf{D}}_0^{-1}$ $(1 < i \le k)$, and $\tilde{\mathbf{D}}_0 = \mathbf{I}_n - \mathbf{D}_0$. The computation of $(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ requires $O(n^2k + nk^2)$ instead of $O(n^3k^3)$ multiplications by utilizing the special structure of $(\mathbf{I}_{nk} - \mathbf{B}_0)$.

The matrices $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ and $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{B}_1$ can be represented as

$$\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1} = \begin{bmatrix} \tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2 & \ldots & \tilde{\mathbf{V}}_{k-1} & \tilde{\mathbf{V}}_k \\ & \tilde{\mathbf{V}}_1 & \ldots & \tilde{\mathbf{V}}_{k-2} & \tilde{\mathbf{V}}_{k-1} \\ & & \ddots & \vdots & \vdots \\ & & & \tilde{\mathbf{V}}_1 & \tilde{\mathbf{V}}_2 \\ & & & & \tilde{\mathbf{V}}_1 \end{bmatrix}, \quad \mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{B}_1 = \begin{bmatrix} \tilde{\tilde{\mathbf{V}}}_1 \\ \tilde{\tilde{\mathbf{V}}}_2 \\ \vdots \\ \tilde{\tilde{\mathbf{V}}}_{k-1} \\ \tilde{\tilde{\mathbf{V}}}_k \end{bmatrix}, \tag{4.2}$$

where $\tilde{\mathbf{V}}_1 = \tilde{\mathbf{L}}\mathbf{V}_1$, $\tilde{\mathbf{V}}_i = \tilde{\mathbf{L}}\mathbf{V}_i + \widetilde{\mathbf{M}}\mathbf{V}_{i-1}$, for $1 < i \le k$, and $\tilde{\tilde{\mathbf{V}}}_i = \tilde{\mathbf{V}}_{k-i+1}\hat{\mathbf{L}}$ $(1 \le i \le k)$. Here, $\tilde{\mathbf{V}}_i$ and $\tilde{\tilde{\mathbf{V}}}_i$ for $1 \le i \le k$ are matrices of order $nd \times n$ and $nd \times nd$, respectively. Here, multiplying $\mathbf{E}$ by $(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ and $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ by $\mathbf{B}_1$ both require $O(n^2k)$ multiplications due to utilizing special structures of matrices. If there were no special structures of $\mathbf{E}$, $(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ and $\mathbf{B}_1$ then the multiplying $\mathbf{E}$ by $(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ and $\mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}$ by $\mathbf{B}_1$ would require $O(n^3k^3d)$ and $O(n^3k^2d^3)$ multiplications, respectively.

The matrix $\mathbf{H}$ of can be represented in the following form:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & & & & \\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & \mathbf{H}_{2,3} & & & \\ \mathbf{H}_{3,1} & & \mathbf{H}_{3,3} & \mathbf{H}_{3,4} & & \\ \vdots & & & \ddots & \ddots & \\ \mathbf{H}_{k-1,1} & & & & \mathbf{H}_{k-1,k-1} & \mathbf{H}_{k-1,k} \\ \mathbf{H}_{k,1} & \mathbf{H}_{k,2} & \mathbf{H}_{k,3} & \cdots & \mathbf{H}_{k,k-1} & \mathbf{H}_{k,k} \end{bmatrix}, \tag{4.3}$$

where $\mathbf{H}_{i,j}$ $(1 \le i, j \le k)$ is a matrix of order $nd \times nd$. Here,

$$\mathbf{H}_{i,j} = \begin{cases} \widetilde{\mathbf{L}}_2 + \widetilde{\widetilde{\mathbf{V}}}_1, & i = j = 1, \\ \widetilde{\mathbf{M}}_2, & 1 \le i \le k-1,\ j = i+1, \\ \widetilde{\mathbf{L}}_2, & 1 < i = j < k, \\ \widetilde{\widetilde{\mathbf{V}}}_i, & 1 < i < k,\ j = 1, \\ \mathbf{R}_1\widetilde{\mathbf{L}}_1 + \widetilde{\widetilde{\mathbf{V}}}_k, & i = k,\ j = 1, \\ \mathbf{R}_{i-1}\widetilde{\mathbf{M}}_1 + \mathbf{R}_i\widetilde{\mathbf{L}}_1, & i = k,\ 1 < j < k, \\ \mathbf{R}_{k-1}\widetilde{\mathbf{M}}_1 + \mathbf{R}_k\widetilde{\mathbf{L}}_1 + \widetilde{\mathbf{L}}_2, & i = j = k. \end{cases} \tag{4.4}$$

We can recursively define $\mathbf{U}_k = \widetilde{\mathbf{U}}_k = \mathbf{H}_{k-1,k}(\mathbf{I} - \mathbf{H}_{k,k})^{-1}$, $\mathbf{U}_i = \mathbf{H}_{i-1,i}(\mathbf{I} - \mathbf{H}_{i,i} - \widetilde{\mathbf{U}}_{i+1}\mathbf{H}_{k,i})^{-1}$ for $2 \le i \le k-1$, $\widetilde{\mathbf{U}}_i = \mathbf{U}_i\widetilde{\mathbf{U}}_{i+1}$ for $2 \le i \le k-1$, $\widetilde{\widetilde{\mathbf{U}}}_2 = \mathbf{U}_2$, $\widetilde{\widetilde{\mathbf{U}}}_i = \widetilde{\widetilde{\mathbf{U}}}_{i-1}\mathbf{U}_i$ for $3 \le i \le k$, and $\mathbf{U}_1 = \mathbf{H}_{1,1} + \sum_{i=2}^{k} \widetilde{\widetilde{\mathbf{U}}}_i\mathbf{H}_{i,1}$.

Let us define another invariant probability vector $\tilde{\mathbf{u}} = [\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2, \tilde{\mathbf{u}}_3, \ldots, \tilde{\mathbf{u}}_k]$ of the stochastic matrix $\mathbf{H}$ where $\tilde{\mathbf{u}}_i$ $(1 \le i \le k)$ is a row vector having nd scalar elements. Here, $\tilde{\mathbf{u}}$ is computed from $\tilde{\mathbf{u}}_1\mathbf{U}_1 = \tilde{\mathbf{u}}_1$, $\tilde{\mathbf{u}}_1\mathbf{1}_{nd} = 1$ and $\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_{i-1}\mathbf{U}_i$ for $2 \le i \le k$. Next a positive real number u is computed as $u = \tilde{\mathbf{u}}(\eta + \zeta)$ where $\eta = \mathbf{E}(\mathbf{I}_{nk} - \mathbf{B}_0)^{-1}\mathbf{1}_{nk}$ and $\zeta = (\mathbf{I}_{nkd} - \mathbf{R})^{-1}\mathbf{1}_{nkd}$. η and ζ are both of size $nkd \times 1$. The column vectors η and ζ can be expressed as $\eta = [\eta_1, \eta_2, \eta_3, \ldots, \eta_{k-1}, \eta_k]$ and $\zeta = [\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_{k-1}, \zeta_k]$, respectively. Both η_i and $\zeta_i (1 \le i \le k)$ are of size $nd \times 1$. $\eta_k = \widetilde{\mathbf{V}}_1\mathbf{1}_n$ and $\eta_j = \eta_{j+1} + \widetilde{\mathbf{V}}_{k-j+1}\mathbf{1}_n$ for $(1 \le j \le k-1)$. There are only $(n-1)$ nonzero elements in η_k and n nonzero elements in η_j for $1 \le j \le k-1$. On the other hand, $\zeta_i = \mathbf{1}_{nd}$ for $1 \le i \le k-1$ and $\zeta_k = (\sum_{j=1}^{k-1} \mathbf{\Phi}_j + \mathbf{\Upsilon})\mathbf{1}_{nd}$.

$\mathbf{x}_1$ can be computed as $\mathbf{x}_1 = u^{-1}\tilde{\mathbf{u}}$ using $O(n^3kd^3)$ multiplications instead of $O(n^3k^3d^3)$ multiplications using the special structure of $\mathbf{H}$. Let us partition $\mathbf{x}_0$ and $\mathbf{x}_1$ as $\mathbf{x}_0 = [\kappa_1, \kappa_2, \kappa_3, \ldots, \kappa_k]$ and $\mathbf{x}_1 = [\nu_1, \nu_2, \nu_3, \ldots, \nu_k]$, respectively, for efficient computation of $\mathbf{x}_0$. Here, κ_i and ν_i for $1 \le i \le k$ are of size $1 \times n$ and $1 \times nd$, respectively. κ_i $(1 \le i \le k)$ is computed as $\kappa_i = \sum_{j=1}^{i} \nu_{i-j+1}\widetilde{\mathbf{V}}_j$ and these vector-matrix multiplications can be made efficient by considering only nonzero rows of $\widetilde{\mathbf{V}}_j (1 \le j \le k)$.

$\mathbf{x}_i$ $(i \ge 2)$ can be computed as $\mathbf{x}_i = \mathbf{x}_{i-1}\mathbf{R}$ for $(i \ge 2)$ using $n(d-1) \times nkd = n^2kd(d-1)$ multiplications instead of $n^2k^2d^2$ multiplications as $\mathbf{R}$ has only $n(d-1)$ nonzero rows.

5. Waiting-Time Distribution of $_k$GI/D/1 System

The waiting time distribution of an arbitrary customer in $_k$GI/D/1 system is the same as the waiting time distribution of a customer in GI/D/k system. Let $w(r)$ be the probability that a customer (i.e., the kth customer of a supercustomer group) has to wait r units of time. $w(r)$ is computed after obtaining the probability vectors $\mathbf{y}_i$'s where $\mathbf{y}_i (i \geq 0)$ is the stationary probability vector that a supercustomer sees i supercustomers ahead of him. $\mathbf{y}_0$ can be expressed as $\mathbf{y}_0 = [\mathbf{y}_{0,1}, \mathbf{y}_{0,2}, \ldots, \mathbf{y}_{0,nk}]$. Similarly, $\mathbf{y}_i$ $(i \geq 1)$ can be expressed as $\mathbf{y}_i = [\mathbf{y}_{i,1}, \mathbf{y}_{i,2}, \ldots, \mathbf{y}_{i,nk}]$ and $\mathbf{y}_{i,v}$ can be expressed as $\mathbf{y}_{i,v} = [\mathbf{y}_{i,v,1}, \mathbf{y}_{i,v,2}, \ldots, \mathbf{y}_{i,v,d}]$.

Two approaches for computing the distribution of waiting time are presented in Sections 5.1 and 5.2, respectively. The first method uses the approach for phase type service time distribution and this approach is used in the analysis of Alfa [3] and Chaudhry et al. [7]. In the second method we develop a computationally simpler approach. This is a simplified version of Method 1. It simplifies the computations of the probability vectors $\mathbf{y}_i$'s for $i \geq 0$ and $w(r)$, the probability mass function of the distribution of waiting time.

5.1. Method 1

The expressions for the stationary distribution of an entering supercustomer to find different number of supercustomers ahead of him in the system can be written as

$$\mathbf{y}_i = \begin{cases} \dfrac{1}{\lambda^*}[\mathbf{x}_0\mathbf{C}_1 + \mathbf{x}_1(\mathbf{C}_1 \otimes \mathbf{s})], & i = 0, \\ \dfrac{1}{\lambda^*}\left[\mathbf{x}_i(\mathbf{C}_1 \otimes \mathbf{S}) + \mathbf{x}_{i+1}\left(\mathbf{C}_1 \otimes (\mathbf{s}\beta)\right)\right], & i \geq 1. \end{cases} \tag{5.1}$$

In this method, the probability that the waiting time is r units of time is computed as

$$w(r) = \begin{cases} \mathbf{y}_0\mathbf{1}_{nk}, & r = 0, \\ \displaystyle\sum_{i=1}^{r} \mathbf{y}_i(\mathbf{1}_{nk} \otimes \mathbf{I}_d)\boldsymbol{\Omega}^{(i)}(r)\mathbf{1}_d, & r \geq 1, \end{cases} \tag{5.2}$$

where $\boldsymbol{\Omega}^{(i)}(r)$ is a square matrix of order d that represents the probability distribution of r unit of work in system at the arrival of a supercustomer who finds that i supercustomers are ahead of him. For example, the element $(\boldsymbol{\Omega}^{(i)}(r))_{u,v}$ represents the probability that the service of the arbitrary supercustomer, who arrives to find i supercustomers in the system with remaining workload of r units begins in phase v $(1 \leq v \leq d)$, given that service of the supercustomer who was in service at the arrival of the arbitrary supercustomer was in phase

u $(1 \le u \le d)$. The matrix $\mathbf{\Omega}^{(i)}(r)$ is computed in the following way:

$$\mathbf{\Omega}^{(i)}(r) = \begin{cases} \mathbf{I}_d, & i = r = 0, \\ (\mathbf{s}\beta)^r, & 1 \le i = r, \\ \mathbf{0}_{d,d}, & i = 0,\ r \ge 1, \\ \mathbf{0}_{d,d}, & i \ge 1,\ r = 0, \\ (\mathbf{S})^{r-1}(\mathbf{s}\beta), & i = 1,\ r \ge 1, \\ (\mathbf{s}\beta)\mathbf{\Omega}^{(i-1)}(r-1) + \mathbf{S}\mathbf{\Omega}^{(i)}(r-1), & 2 \le i \le r. \end{cases} \tag{5.3}$$

5.2. Method 2

This method simplifies the computation of $\mathbf{y}_0$ and $\mathbf{y}_i$ $(i \ge 1)$. Moreover, this method does not need to compute the matrix $\mathbf{\Omega}^{(i)}(r)$ $(i, r \ge 0)$ which simplifies the computation of the probability mass function of the distribution of waiting time.

The special structures of $\mathbf{C}_1$, $\mathbf{s}$, $\mathbf{S}$, and β can be exploited to reduce the number of operations to compute $\mathbf{y}_0$ and $\mathbf{y}_i$ $(i \ge 1)$. Only the first columns of the matrices $\mathbf{C}_1$ and $\mathbf{C}_1 \otimes s$ are nonzero. Therefore, the first element of $\mathbf{y}_0$ is nonzero and the remaining $(nk-1)$ elements of $\mathbf{y}_0$ are equal to zero. This phenomenon is quite intuitive as the arriving supercustomer does not find anybody ahead of him, he can start his service in phase 1 instantly and the arrival process for next supercustomer starts at phase 1 of the nk phases. The first element $\mathbf{y}_{0,1}$ of $\mathbf{y}_0$ is computed as

$$\mathbf{y}_{0,1} = \frac{1}{\lambda^*}\sum_{j=1}^{n}\left(\mathbf{x}_{0,(k-1)n+j}\mathbf{t}_j + \mathbf{x}_{1,(k-1)n+j,d}\mathbf{t}_j\right). \tag{5.4}$$

$(d-1)$ columns of $(\mathbf{C}_1 \otimes \mathbf{S})$ are nonzero which are column 2 to column d. On the other hand, only the first column of $(\mathbf{C}_1 \otimes (s\beta))$ is nonzero. Therefore, the first d elements of $\mathbf{y}_i (i \ge 1)$ are nonzero. It is quite intuitive that the first d elements of $\mathbf{y}_i$ $(i \ge 1)$ are nonzero (i.e., $\mathbf{y}_{i,1}$ is nonzero and $\mathbf{y}_{i,v} = \mathbf{0}_{1,d}$ $(2 \le v \le nk)$) as the arriving supercustomer finds i supercustomers ahead of him, the arrival process of next supercustomer starts at phase 1 of nk phases and the ongoing service is in the start of phase j $(1 \le j \le d)$. The d nonzero elements $\mathbf{y}_{i,1,j}$'s $(i \ge 1,\ 1 \le j \le d)$ of the vector $\mathbf{y}_i$ can be computed using

$$\mathbf{y}_{i,1,j} = \begin{cases} \dfrac{1}{\lambda^*}\displaystyle\sum_{l=1}^{n}\mathbf{x}_{i+1,(k-1)n+l,d}\mathbf{t}_l, & j = 1, \\ \dfrac{1}{\lambda^*}\displaystyle\sum_{l=1}^{n}\mathbf{x}_{i,(k-1)n+l,j-1}\mathbf{t}_l, & 2 \le j \le d. \end{cases} \tag{5.5}$$

The special structures of $\mathbf{1}_{nk} \otimes \mathbf{I}_d$ and $\mathbf{y}_i$ for $i \ge 1$ result in

$$\mathbf{y}_i(\mathbf{1}_{nk} \otimes \mathbf{I}_d) = \mathbf{y}_{i,1}. \tag{5.6}$$

If a supercustomer sees i supercustomers ahead of him then he has to wait or $id+1, id+2, \ldots, id+d$ amount of time for $d > 1$ (or i units of time for $d = 1$) before getting service.

First we consider the case of $d = 1$. In this case, $\mathbf{S} = [0]$, $\mathbf{s} = [1]$, and $\beta = [1]$. For $d = 1$, (5.3) can be written as

$$\boldsymbol{\Omega}^{(i)}(r) = \begin{cases} 1, & i = r = 0, \\ 1, & 1 \le i = r, \\ 0, & i = 0, r \ge 1, \\ 0, & i = 1, r \ge 1, \\ \boldsymbol{\Omega}^{(i-1)}(r-1), & 2 \le i \le r. \end{cases} \tag{5.7}$$

For $i < r$ we can assume $i + i_1 = r$ and $i_1 > 0$. In this case, $\boldsymbol{\Omega}^{(i)}(r) = \boldsymbol{\Omega}^{(i-1)}(r-1) = \cdots = \boldsymbol{\Omega}^{(0)}(i_1) = 0$.

For $i = r$, we have $\boldsymbol{\Omega}^{(i)}(r) = 1$ from (5.7).

For $i > r$ we can assume $i = r + i_1$ and $i_1 > 0$. In this case, $\boldsymbol{\Omega}^{(i)}(r) = \boldsymbol{\Omega}^{(i-1)}(r-1) = \cdots = \boldsymbol{\Omega}^{(i_1)}(0) = 0$. Therefore, we can express for $d = 1$,

$$\boldsymbol{\Omega}^{(i)}(r)\mathbf{1}_d = \begin{cases} 1, & i = r, \\ 0, & \text{otherwise.} \end{cases} \tag{5.8}$$

Now, we consider the case $d \ge 2$. Let us define $\mathbf{e}_j$ $(1 \le j \le d)$ as the jth column of an identity matrix of order d. It is found that for $i \ge 1$ and $d \ge 2$

$$\begin{aligned} \boldsymbol{\Omega}^{(i)}\big((i-1)d - j\big) &= \mathbf{0}_{d,d} \quad \text{for } j = 0, 1, 2, \ldots, (i-1)d, \\ \boldsymbol{\Omega}^{(i)}\big((i-1)d + j\big) &= \mathbf{S}^{j-1}(\mathbf{s}\beta) = \begin{bmatrix} \mathbf{e}_{d-j+1} & \mathbf{0}_{d,d-1} \end{bmatrix} \quad \text{for } j = 1, 2, \ldots, d, \\ \boldsymbol{\Omega}^{(i)}\big(id + j\big) &= \mathbf{0}_{d,d} \quad \text{for } j \ge 1, \end{aligned} \tag{5.9}$$

which are proved by induction in [30]. If the matrix $\boldsymbol{\Omega}^{(i)}(r)$ is nonzero then it contains nonzero elements in the first column. The nonzero first column in nonzero $\boldsymbol{\Omega}^{(i)}(r)$ is due to elapsed time representation of service time where service starts at phase 1.

Now, $[\mathbf{e}_{d-j+1} \quad \mathbf{0}_{d,d-1}]\mathbf{1}_d = [\mathbf{e}_{d-j+1}]$. Therefore, we can express for $d \ge 2$,

$$\boldsymbol{\Omega}^{(i)}(r)\mathbf{1}_d = \begin{cases} \left[\mathbf{e}_{d-(r \bmod d)}\right], & i = \left\lceil \dfrac{r}{d} \right\rceil, \\ \mathbf{0}_{d,1}, & \text{otherwise.} \end{cases} \tag{5.10}$$

Alfa [2, 3] pointed out that $\left(\boldsymbol{\Omega}^{(i)}(r)\right)_{u,v} = 0$ for $i \ge 1$, $2 \le v \le d$. However, he did not provide any explicit expression for $\boldsymbol{\Omega}^{(i)}(r)\mathbf{1}_d$ and his method needs to compute the first column of the matrix $\boldsymbol{\Omega}^{(i)}(r)$ $(i \ge 1)$ recursively as our method does not need to compute. Use of (5.10) to compute $\boldsymbol{\Omega}^{(i)}(r)\mathbf{1}_d$ can significantly reduce the complexity of computing

Table 1: Number of iterations required for different methods for interarrival times having probability vector $\underline{\mathbf{a}}$ = [0.1 0.2 0.2 0.5].

ε	Natural **R**	Traditional **R**	**U**-based **R**	Natural **G**	Traditional **G**	**U**-based **G**	LR **R**	LR **G**	Structure **R**
10^{-6}	157	35	5	194	40	5	3	2	157
10^{-7}	186	41	6	222	46	5	3	2	186
10^{-8}	214	47	6	250	52	5	3	2	214
10^{-9}	243	53	6	279	58	6	3	2	243

the distribution of waiting time for any multiserver queueing system in discrete time with deterministic service time such as BMAP/D/k system of Alfa [2] and MAP/D/k systems of Alfa [3] and Chaudhry et al. [7]. Equation (5.10) reduces the usage of memory as there is no need to compute and store the matrix $\mathbf{\Omega}^{(i)}(r)$ for different values of i and r.

Now, if an arriving customer has to wait r units of time ($r \geq 0$) then there are $\lceil r/d \rceil$ supercustomers ahead of him and the customer, who is in service in the server, is in phase $(d - (r \mod d))$ of service time. Therefore, the probability of a customer waiting r ($r \geq 0$) units of time can be expressed by using value of $\mathbf{\Omega}^{(i)}(r)\mathbf{1}_d$ of (5.8) and (5.10) in (5.2)

$$w(r) = \begin{cases} \mathbf{y}_{0,1}, & r = 0, \\ \mathbf{y}_{r,1,1}, & r \geq 1,\ d = 1, \\ \mathbf{y}_{\lceil r/d \rceil,1,d-(r \bmod d)}, & r \geq 1,\ d \geq 2. \end{cases} \tag{5.11}$$

6. Numerical Examples

This section compares our proposed method of computing the matrix **R** with natural, traditional, and U-based methods and Logarithmic Reduction technique [26] for the matrices **R** and **G**. Three different matrices **R**, **G,** and **U** are computed in each of these methods. The values of the convergence constant ε used are 10^{-6}, 10^{-7}, 10^{-8}, and 10^{-9}. Three identical servers (i.e., $k = 3$) are considered with duration of service as eight time units (i.e., $d = 8$) and the support for interarrival times is assumed to be four (i.e., $n = 4$). Three different probability vectors [0.1, 0.2, 0.2, 0.5], [0.1, 0.2, 0.3, 0.4], and [0.1, 0.3, 0.4, 0.2] are used for interarrival times for numerical examples. A program is written in C programming language for different methods to compute the matrices **R**, **G**, and **U**. The program was executed in SunOS 5.10 [31] on an i386 processor which operates at 2660 MHz and has an i387 compatible floating point processor. The number of iterations and time in seconds is recorded in Tables 1–6 for each method from the execution of the program. "LR" is used to denote logarithmic reduction technique in those tables.

Tables 1 and 2 show the number of iterations and time required to compute the matrices **R**, **G**, and **U** for the interarrival times having probability vector [0.1, 0.2, 0.2, 0.5]. The means of interarrival time and arrival rate for this probability vector are 3.1 time slots and 0.3226/time slot, respectively, and $\rho = 0.8602$. Tables 3 and 4 show the number of iterations and time required for the interarrival times having probability vector [0.1, 0.2, 0.3, 0.4]. The means of interarrival time and arrival rate for the probability vector [0.1, 0.2, 0.3, 0.4] are 3.0 time slots and 0.3333/time slot, respectively, and $\rho = 0.8889$. Our methods of computing the matrix **R** are found to require least time for these two examples among the methods used

Table 2: Time in seconds required for different methods to execute for interarrival times having probability vector $\underline{\mathbf{a}}$ = [0.1 0.2 0.2 0.5].

	Natural	Traditional	**U**-based	Natural	Traditional	**U**-based	LR	LR	Structure
ε	**R**	**R**	**R**	**G**	**G**	**G**	**R**	**G**	**R**
10^{-6}	3.1825	1.1115	0.2081	3.8766	1.2642	0.2075	0.3497	0.2579	0.0724
10^{-7}	3.8069	1.3013	0.2451	4.4946	1.4159	0.2070	0.3530	0.2751	0.0776
10^{-8}	4.3452	1.4641	0.2454	5.0430	1.6042	0.2070	0.3654	0.2590	0.0844
10^{-9}	4.9589	1.6619	0.2633	5.6928	1.7840	0.2460	0.3510	0.2591	0.0895

Table 3: Number of iterations required for different methods for interarrival times having probability vector $\underline{\mathbf{a}}$ = [0.1, 0.2, 0.3, 0.4].

	Natural	Traditional	**U**-based	Natural	Traditional	**U**-based	LR	LR	Structure
ε	**R**	**R**	**R**	**G**	**G**	**G**	**R**	**G**	**R**
10^{-6}	207	48	6	253	54	5	3	2	207
10^{-7}	245	56	6	291	62	5	3	2	245
10^{-8}	283	64	7	328	70	6	3	2	283
10^{-9}	322	72	7	366	79	6	3	2	322

Table 4: Time in seconds required for different methods to execute for interarrival times having probability vector $\underline{\mathbf{a}}$ = [0.1, 0.2, 0.3, 0.4].

	Natural	Traditional	**U**-based	Natural	Traditional	**U**-based	LR	LR	Structure
ε	**R**	**R**	**R**	**G**	**G**	**G**	**R**	**G**	**R**
10^{-6}	4.2406	1.4866	0.2500	5.1100	1.6555	0.2065	0.3534	0.2591	0.0831
10^{-7}	5.0294	1.7283	0.2455	6.1509	1.9375	0.2199	0.3534	0.2586	0.0904
10^{-8}	5.7546	1.9552	0.2809	6.5859	2.1162	0.2428	0.3548	0.2564	0.1003
10^{-9}	6.5022	2.1919	0.2791	7.3162	2.3785	0.2478	0.3533	0.2586	0.1077

Table 5: Number of iterations required for different methods for interarrival times having probability vector $\underline{\mathbf{a}}$ = [0.1, 0.3, 0.4, 0.2].

	Natural	Traditional	**U**-based	Natural	Traditional	**U**-based	LR	LR	Structure
ε	**R**	**R**	**R**	**G**	**G**	**G**	**R**	**G**	**R**
10^{-6}	1864	545	25	2272	614	22	5	4	1864
10^{-7}	2286	656	28	2661	721	26	5	4	2285
10^{-8}	2707	766	32	3051	828	29	6	4	2706
10^{-9}	3128	876	36	3441	934	33	6	5	3127

here though logarithmic reduction technique and U-based methods require less number of iterations. Tables 5 and 6 show the number of iterations and time required for the interarrival times having probability vector [0.1, 0.3, 0.4, 0.2]. The means of interarrival time and arrival rate for this probability vector [0.1, 0.3, 0.4, 0.2] of interarrival times are 2.7 time slots and 0.3704/time slot, respectively, and $\rho = 0.9877$. Table 6 shows that for the interarrival times probability vector [0.1, 0.3, 0.4, 0.2], Logarithmic Reduction techniques for the matrices **G** and **R** require less time to converge than our method of exploiting the structure of the matrices $\mathbf{A}_2$, $\mathbf{A}_1$, and $\mathbf{A}_0$. The relative advantage of our method of exploiting the structures to Logarithmic Reduction techniques decreases with respect to time with increasing value of ρ.

Table 6: Time in seconds required for different methods to execute for interarrival times having probability vector $\underline{\mathbf{a}} = [0.1, 0.3, 0.4, 0.2]$.

ε	Natural **R**	Traditional **R**	**U**-based **R**	Natural **G**	Traditional **G**	**U**-based **G**	LR **R**	LR **G**	Structure **R**
10^{-6}	37.5583	16.2450	0.9116	45.2144	18.2392	0.7928	0.5395	0.4461	0.4579
10^{-7}	46.3076	19.6171	1.0278	53.3722	21.4709	0.9453	0.5457	0.4585	0.5526
10^{-8}	54.5964	22.9092	1.1562	61.0474	24.5655	1.0368	0.6334	0.4479	0.6488
10^{-9}	63.9370	26.9160	1.3035	70.1627	28.3890	1.2020	0.6346	0.5453	0.7593

7. Conclusion

In this paper, a class of GI/D/k systems in discrete time is analyzed by converting it to a single server queue problem with a convoluted arrival process. Then the stationary distribution of the length of the queue of the transformed system is analyzed using QBD approach. The special structures of the matrices make the computation of the matrix **R** and the distribution of queue length of the transformed system efficient. Numerical examples show that our proposed method for computing the matrix **R** can be as efficient as quadratic convergent algorithms such as logarithmic reduction technique. It is also shown that for deterministic service time the waiting time distribution does not need any cumbersome computation like phase type service time distribution.

Acknowledgment

This research is partially supported by grants from NSERC to A. S. Alfa.

References

[1] C. D. Crommelin, "Delay probability formulae when the holding times are constant," *Post Ofiice Electrical Engineers Journal*, vol. 25, pp. 41–50, 1932.

[2] A. S. Alfa, "Algorithmic analysis of the BMAP/D/k system in discrete time," *Advances in Applied Probability*, vol. 35, no. 4, pp. 1131–1152, 2003.

[3] A. S. Alfa, "Waiting time distribution of the MAP/D/k system in discrete time—a more efficient algorithm," *Operations Research Letters*, vol. 31, no. 4, pp. 263–267, 2003.

[4] I. Wuyts and H. Bruneel, "Analysis of discrete-time multiserver queueing models with constant service times," *Operations Research Letters*, vol. 15, no. 5, pp. 231–236, 1994.

[5] M. L. Chaudhry and N. K. Kim, "A complete and simple solution for a discrete-time multi-server queue with bulk arrivals and deterministic service times," *Operations Research Letters*, vol. 31, no. 2, pp. 101–107, 2003.

[6] M. L. Chaudhry, J. G. C. Templeton, and J. Medhi, "Computational results of multiserver bulk-arrival queues with constant service time M^X/D/c," *Operations Research*, vol. 40, no. 2, pp. 229–238, 1992.

[7] M. L. Chaudhry, B. K. Yoon, and K. C. Chae, "Waiting-time distribution of a discrete-time multiserver queue with correlated arrivals and deterministic service times: D − MAP/D/k system," *Operations Research Letters*, vol. 30, no. 3, pp. 174–180, 2002.

[8] G. J. Franx, "A simple solution for the M/D/c waiting time distribution," *Operations Research Letters*, vol. 29, no. 5, pp. 221–229, 2001.

[9] G. J. Franx, "The M^X/D/c batch arrival queue," *Probability in the Engineering and Informational Sciences*, vol. 19, no. 3, pp. 345–349, 2005.

[10] P. Gao, S. Wittevrongel, and H. Bruneel, "Delay analysis for a discrete-time GI-D-c queue with arbitrary-length service times," in *Proceedings of the 1st European Performance Engineering Workshop (EPEW '04)*, pp. 184–195, Toledo, Spain, September 2004.

[11] P. Gao, S. Wittevrongel, and H. Bruneel, "Analysis of discrete-time multiserver queues with constant service times and correlated arrivals," in *Proceedings of the 12th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA '05)*, pp. 1–8, Riga, Latvia, May 2005.
[12] P. Gao, S. Wittevrongel, and H. Bruneel, "Delay and partial system contents for a discrete-time G-D-c queue," *4OR: A Quarterly Journal of Operations Research*, vol. 6, no. 3, pp. 279–290, 2008.
[13] P. Gao, S. Wittevrongel, J. Walraevens, and H. Bruneel, "Analytic study of multiserver buffers with two-state Markovian arrivals and constant service times of multiple slots," *Mathematical Methods of Operations Research*, vol. 67, no. 2, pp. 269–284, 2008.
[14] P. Gao, S. Wittevrongel, J. Walraevens, M. Moeneclaey, and H. Bruneel, "Calculation of delay characteristics for multiserver queues with constant service times," *European Journal of Operational Research*, vol. 199, no. 1, pp. 170–175, 2009.
[15] V. B. Iversen, "Decomposition of an M/D/rk queue with FIFO into K E_k/D/r queues with FIFO," *Operations Research Letters*, vol. 2, no. 1, pp. 20–21, 1983.
[16] M. H. van Hoorn, "Numerical analysis of multiserver queues with deterministic service and special phase-type arrivals," *Zeitschrift für Operations Research*, vol. 30, no. 1, pp. A15–A28, 1986.
[17] S. Nishimura, "A spectral analysis for MAP/D/N queue," in *Advances in Algorithmic Methods for Stochastic Models*, G. Latouche and P. Taylor, Eds., pp. 279–294, Notable, Neshanic Station, NJ, USA, 2000.
[18] T. Takine, "A simple approach to the MAP/D/s queue," *Stochastic Models*, vol. 16, no. 5, pp. 543–556, 2000.
[19] S. Wittevrongel and H. Bruneel, "Discrete-time queues with correlated arrivals and constant service times," *Computers & Operations Research*, vol. 26, no. 2, pp. 93–108, 1999.
[20] D. A. Xerocostas and C. Demertzes, "Steady state solution of the E_k/D/r queueing model," *OR Spektrum*, vol. 4, no. 1, pp. 47–51, 1982.
[21] N. K. Kim and M. L. Chaudhry, "The use of the distributional little's law in the computational analysis of discrete-time GI/G/1 and GI/D/c queues," *Performance Evaluation*, vol. 65, no. 1, pp. 3–9, 2008.
[22] M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models*, vol. 2, Johns Hopkins University Press, Baltimore, Md, USA, 1981.
[23] M. L. Chaudhry, "Computing stationary queueing-time distributions of GI/D/1 and GI/D/c queues," *Naval Research Logistics*, vol. 39, no. 7, pp. 975–996, 1992.
[24] A. S. Alfa, "Combined elapsed time and matrix-analytic method for the discrete time GI/G/1 and GI^X/G/1 systems," *Queueing Systems*, vol. 35, no. 1, pp. 5–25, 2003.
[25] A. S. Alfa and J. Xue, "Efficient computations for the discrete GI/G/1 system," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 480–484, 2007.
[26] G. Latouche and V. Ramaswami, "A logarithmic reduction algorithm for quasi-birth-death processes," *Journal of Applied Probability*, vol. 30, no. 3, pp. 650–674, 1993.
[27] D. A. Bini and B. Meini, "On cyclic reduction applied to a class of toeplitz-like matrices arising in queueing problems," in *Computations with Markov Chains*, W. J. Stewart, Ed., pp. 21–38, Kluwer Academic Publisher, Dordrecht, The Netherlands, 1995.
[28] N. Akar and K. Sohraby, "An invariant subspace approach in M/G/1 and G/M/1 type Markov chains," *Stochastic Models*, vol. 13, no. 3, pp. 381–416, 1997.
[29] G. Hadley, *Linear Algebra*, Addison-Wesley, Reading, Mass, USA, 1969.
[30] M. M. Rahman and A. S. Alfa, "Computational procedures for a class of GI/D/k systems in discrete time—an extended analysis," http://www.ee.umanitoba.ca/~mmrahman/gidkreport.pdf.
[31] P. A. Watters, *Solaris 10: The Complete Reference*, McGraw-Hill, New York, NY, USA, 2005.