INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Efficient Computations for the Discrete GI/G/1 System

Attahiru Sule Alfa, Jungong Xue,

To cite this article:
Attahiru Sule Alfa, Jungong Xue, (2007) Efficient Computations for the Discrete GI/G/1 System. INFORMS Journal on Computing 19(3):480-484. https://doi.org/10.1287/ijoc.1060.0190

INFORMS Journal on Computing
Vol. 19, No. 3, Summer 2007, pp. 480–484
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1903 | 0480

DOI 10.1287/ijoc.1060.0190

Efficient Computations for the Discrete GI/G/1 System

Attahiru Sule Alfa
Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, Manitoba R3T 5V6, Canada, alfa@ee.umanitoba.ca

Jungong Xue
Department of Mathematics, Fudan University, Shanghai 200433, China, xuej@fudan.edu.cn

We consider the discrete-time GI/G/1 system with discrete interarrival times and service-times distributions that have finite supports, and formulate it as a PH/PH/1 system. We then take advantage of the resulting special structure to develop efficient methods for computing its rate matrices and the decay rates of its queue length and waiting time.

Key words: matrix-analytic methods; discrete GI/G/1 systems; discrete PH/PH/1 systems; queue length; waiting time; tail probabilities
History: Accepted by Edward P. C. Kao, Area Editor for Computational Probability and Analysis; received December 2004; revised August 2005, November 2005, February 2006; accepted April 2006. Published online in *Articles in Advance* July 20, 2007.

1. Introduction

The GI/G/1 system is the most general of all single-server queues with renewal processes. Using discrete-time approaches and matrix-analytic methods, Alfa and Li (2001) and Alfa (2003, 2004) present algorithmic approaches for studying this system as a special case of the PH/PH/1. They obtained the distributions of the queue length, waiting time, busy period, etc. One of the weaknesses of their approach is that the order of the matrices resulting from their analysis could be large because of the representations used for both the interarrival and service times, leading to sizeable computational effort for performance measures of the GI/G/1. Taking advantage of some special structures that the blocks in the transition matrices assume, we develop efficient methods for computing the rate matrices and stationary distributions for the queue length and waiting time. To some extent, these methods circumvent the difficulties associated with the order of the associated matrices.

In this discrete-time GI/G/1 system, if the support for the interarrival and service times is finite, it is well known that both the queue length and waiting time have geometric tails. In many situations, the tail probabilities can be estimated through computation of their decay rates. We show that the decay rates can be computed very quickly.

Model parameters are described in Section 2. In Sections 3 and 4, we study the Markov chains for the queue length and waiting time and develop efficient methods to compute the rate matrices, respectively. Section 5 discusses how to compute the decay rates. We summarize the numerical experiments in Section 6.

2. Model Parameters

First we define discrete time. Let us observe this system at equally spaced time epochs sequentially numbered $0, 1, 2, \ldots$. We assume that observations are carried out only at the beginning of an epoch. Hence all events that occur between epochs n and $n+1$ are assumed to occur at epoch $n+1$, $n = 0, 1, 2, \ldots$ (as in Dafermos and Neuts 1971). The basic assumptions and notation are as follows:

- Interarrival times $\mathbf{A}$ of the customers are i.i.d. with distribution $a_j = \Pr\{\mathbf{A} = j\}$, $j = 1, 2, \ldots, N < \infty$.
- Service times $\mathbf{S}$ of the customers are i.i.d. with distribution $s_j = \Pr\{\mathbf{S} = j\}$, $j = 1, 2, \ldots, M < \infty$.
- Let $\boldsymbol{\alpha} = [a_1, a_2, \ldots, a_N]$ and $\boldsymbol{\beta} = [s_1, s_2, \ldots, s_M]$.

Let I_j be the identity matrix of order j, and $\mathbf{0}$ and $\mathbf{1}$ the column vectors of zeros and ones, respectively, of appropriate order. Let $\mathbf{0}'$ denote the transpose of $\mathbf{0}$.

The following proposition is pointed out in Alfa (2004).

Proposition 1. *Let $\{p_j\}$ be a discrete distribution on the positive integers with finite support. Let K be the smallest K for which $\sum_{j>K} p_j = 0$. Then this discrete distribution has phase type representation $(\boldsymbol{\varrho}, H)$, where*

$$\boldsymbol{\varrho} = [p_1, p_2, \ldots, p_K], \qquad H = \begin{bmatrix} \mathbf{0}' & 0 \\ I_{K-1} & \mathbf{0} \end{bmatrix},$$

in terms of a Markov chain with states $0, 1, 2, \ldots, K$, where 0 is the absorbing state.

According to Proposition 1, the interarrival-time distribution $\{a_j\}$ and service-time distribution $\{s_j\}$ can be represented by phase-type distributions $(\boldsymbol{\alpha}, T)$ and $(\boldsymbol{\beta}, S)$, respectively, where

$$T = \begin{bmatrix} \mathbf{0}' & 0 \\ I_{N-1} & \mathbf{0} \end{bmatrix}, \qquad S = \begin{bmatrix} \mathbf{0}' & 0 \\ I_{M-1} & \mathbf{0} \end{bmatrix}.$$

This representation will give us an algorithmic structure that will help us analyze the system of interest.

We assume that $E[\mathbf{A}] > E[\mathbf{S}]$, so the system is stable. We also assume that both the interarrival-time and service-time distributions are aperiodic.

3. The Queue Length

It is immediately clear from the discussions of Section 2 that the discrete-time GI/G/1 system can be studied as a special case of the discrete time PH/PH/1 system. In this section we develop the Markov chain associated with the queue length.

At time j, let Q_j be the number of customers in the system, J_j be the remaining service time of the customer who is in service, and K_j be the remaining time before the next customer arrives. Consider the state space $\Delta_q = \{(0, K_j) \cup (Q_j, J_j, K_j),\ Q_j \geq 1,\ J_j = 1, 2, \ldots, M,\ K_j = 1, 2, \ldots, N\}$. It is immediately clear that this is a Markov chain with transition matrix

$$P_q = \begin{bmatrix} B_1 & B_0 & & & & \\ B_2 & A_1 & A_0 & & & \\ & A_2 & A_1 & A_0 & & \\ & & A_2 & A_1 & A_0 & \\ & & & \ddots & \ddots & \ddots \end{bmatrix},$$

where $B_0 = \boldsymbol{\beta} \otimes (\mathbf{t}\boldsymbol{\alpha})$, $B_1 = T$, $B_2 = \mathbf{s} \otimes T$, $A_0 = S \otimes (\mathbf{t}\boldsymbol{\alpha})$, $A_1 = (\mathbf{s}\boldsymbol{\beta}) \otimes (\mathbf{t}\boldsymbol{\alpha}) + S \otimes T$, $A_2 = (\mathbf{s}\boldsymbol{\beta}) \otimes T$. Here, $\mathbf{t} = \mathbf{1} - T\mathbf{1}$ and $\mathbf{s} = \mathbf{1} - S\mathbf{1}$. Let $\boldsymbol{\pi}$ be the stationary distribution of this Markov chain and, according to the partition of P_q, be partitioned as $\boldsymbol{\pi} = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \ldots]$. From the matrix-geometric theorem (Neuts 1981), $\boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_1 R_q^k$, $k \geq 1$, where R_q is the minimal nonnegative solution to the matrix quadratic equation $R_q = A_0 + R_q A_1 + R_q^2 A_2$. Note that $\boldsymbol{\pi}_0$ satisfies

$$\boldsymbol{\pi}_0 (B_1 + HB_2) = \boldsymbol{\pi}_0, \tag{1}$$

$$\boldsymbol{\pi}_0 \mathbf{1} + \boldsymbol{\pi}_0 H (I_{MN} - R_q)^{-1} \mathbf{1} = 1, \tag{2}$$

and $\boldsymbol{\pi}_1 = \boldsymbol{\pi}_0 H$. Here, $H = B_0 (I_{MN} - A_1 - R_q A_2)^{-1}$. The following iteration is an important computational scheme to calculate the rate matrix R_q:

$$\begin{aligned} R_q(0) &= 0, \\ R_q(n+1) &= A_0 [I_{MN} - A_1 - R_q(n) A_2]^{-1}. \end{aligned} \tag{3}$$

From Neuts (1981), the matrix sequence $\{R_q(n)\}$ converges increasingly to R_q. Generally speaking, this method converges faster than do some linear fixed-point algorithms, but it needs to perform a matrix inverse, which is of order MN, at each iteration. We now discuss how to compute the matrix inversion efficiently.

Consider the matrix sequence $\mathscr{X} = \{X_i,\ 0 \leq i \leq N-1\}$, where $X_i \in \mathbf{R}^{M \times M}$. We define $f(\mathscr{X}) = \sum_{i=0}^{N-1} X_i \otimes (\mathbf{t}\boldsymbol{\alpha} T^i)$. We adopt the convention that $T^0 = I_N$ and $S^0 = I_M$. We now present a lemma to describe the inverse of $I_{MN} - f(\mathscr{X})$.

Lemma 2. *Let $\mathscr{X} = \{X_i,\ 0 \leq i \leq N-1\}$ and $X^* = \sum_{i=0}^{N-1} a_{i+1} X_i$, where $X_i \in \mathbf{R}^{M \times M}$. If $I_M - X^*$ is invertible, then $I_{MN} - f(\mathscr{X})$ is invertible and*

$$(I_{MN} - f(\mathscr{X}))^{-1} = I_{MN} + f(\widehat{\mathscr{X}}), \tag{4}$$

where $\widehat{\mathscr{X}} = \{\widehat{X}_i,\ 0 \leq i \leq N-1\}$ and $\widehat{X}_i = (I_M - X^*)^{-1} X_i$.

Proof. It is clear that

$$\begin{aligned} f^2(\mathscr{X}) &= \sum_{j=0}^{N-1} \left(\sum_{i=0}^{N-1} [X_i \otimes (\mathbf{t}\boldsymbol{\alpha} T^i)][X_j \otimes (\mathbf{t}\boldsymbol{\alpha} T^j)] \right) \\ &= \sum_{j=0}^{N-1} (X^* X_j) \otimes (\mathbf{t}\boldsymbol{\alpha} T^j) = (X^* \otimes I_N) f(\mathscr{X}). \end{aligned}$$

We thus have $[(I_M - X^*)^{-1} \otimes I_N](f(\mathscr{X}) - f^2(\mathscr{X})) = f(\mathscr{X})$. It follows from

$$\begin{aligned} I_{MN} &= I_{MN} - f(\mathscr{X}) + [(I_M - X^*)^{-1} \otimes I_N](f(\mathscr{X}) - f^2(\mathscr{X})) \\ &= (I_{MN} + [(I_M - X^*)^{-1} \otimes I_N] f(\mathscr{X}))(I_{MN} - f(\mathscr{X})) \end{aligned}$$

that (4) holds. □

Applying Lemma 2 to iteration (3), we derive the following result, which states that for each n, there exists a matrix sequence $\mathscr{X}(n) = \{X_i(n),\ 0 \leq i \leq N-1\}$ such that $R_q(n) = f(\mathscr{X}(n))$.

Theorem 3. *Let $S^* = \sum_{i=1}^{K} a_i S^i$, where $K = \min(M-1, N-1)$. Then the matrix sequence $\{R_q(n), n \geq 0\}$ generated by iteration* (3) *assumes the form*

$$R_q(n) = f(\mathscr{X}(n)), \tag{5}$$

where $\mathscr{X}(n) = \{X_i(n),\ 0 \leq i \leq N-1\}$. From $\mathscr{X}(n)$ construct $\mathscr{Y}(n) = \{Y_i(n),\ 0 \leq i \leq N-1\}$, where $Y_i(n)$ is successively defined as

$$\begin{aligned} Y_0(n) &= \mathbf{s}\boldsymbol{\beta}, \\ Y_i(n) &= Y_{i-1}(n) S + X_{i-1}(n) \mathbf{s}\boldsymbol{\beta}, \quad 1 \leq i \leq N-1. \end{aligned} \tag{6}$$

Then $\mathscr{X}(n+1)$ is determined from $\mathscr{Y}(n)$ by

$$X_i(n+1) = S^{i+1} + S^*(I_M - Y^*(n))^{-1} Y_i(n), \quad 0 \leq i \leq N-1, \tag{7}$$

where $Y^*(n) = \sum_{i=0}^{N-1} a_{i+1} Y_i(n)$. Moreover, R_q assumes the form $R_q = f(\mathscr{X})$, where $\mathscr{X} = \{X_i,\ 0 \leq i \leq N-1\}$ and $X_i = \lim_{n \to \infty} X_i(n)$.

Proof. It is clear that $R_q(0) = f(\mathscr{X}(0))$, where $\mathscr{X}(0) = \{X_i(0),\ 0 \le i \le N-1\}$ and $X_i(0) = 0$. Suppose that (5) holds for $R_q(n)$. We now prove that it also holds for $R_q(n+1)$. Since $R_q(n)A_2 = \sum_{i=1}^{N-1}(X_{i-1}(n)\mathbf{s}\boldsymbol{\beta}) \otimes (\mathbf{t}\boldsymbol{\alpha}T^i)$, iteration (3) can be written as

$$R_q(n+1) = (S \otimes \mathbf{t}\boldsymbol{\alpha})(I_{MN} - S \otimes T)^{-1} \cdot (I_{MN} - \tilde{R}_q(n)(I_{MN} - S \otimes T)^{-1})^{-1}, \quad (8)$$

where

$$\tilde{R}_q(n) = (\mathbf{s}\boldsymbol{\beta}) \otimes (\mathbf{t}\boldsymbol{\alpha}) + \sum_{i=1}^{N-1}(X_{i-1}(n)\mathbf{s}\boldsymbol{\beta}) \otimes (\mathbf{t}\boldsymbol{\alpha}T^i). \quad (9)$$

Because $(I_{MN} - S \otimes T)^{-1} = I_{MN} + \sum_{i=1}^{K} S^i \otimes T^i$, we have $(S \otimes \mathbf{t}\boldsymbol{\alpha})(I_{MN} - S \otimes T)^{-1} = \sum_{i=1}^{K} S^i \otimes (\mathbf{t}\boldsymbol{\alpha}T^{i-1})$ and that there exists the matrix sequence $\mathscr{Y}(n) = \{Y_i(n),\ 0 \le i \le N-1\}$ such that $\tilde{R}_q(n)(I_{MN} - S \otimes T)^{-1} = f(\mathscr{Y}(n))$. Comparing (9) with $\tilde{R}_q(n) = f(\mathscr{Y}(n))(I_{MN} - S \otimes T) = Y_0(n) + \sum_{i=1}^{N-1}(Y_i(n) - Y_{i-1}(n)S) \otimes (\mathbf{t}\boldsymbol{\alpha}T^i)$, we show that the matrices $Y_i(n)$, $0 \le i \le N-1$, satisfy (6). Applying Lemma 2 to (8), we have

$$\begin{aligned} R_q(n+1) &= \left(\sum_{i=1}^{K} S^i \otimes (\mathbf{t}\boldsymbol{\alpha}T^{i-1})\right)(I_{MN} - f(\mathscr{Y}(n)))^{-1} \\ &= \left(\sum_{i=1}^{K} S^i \otimes (\mathbf{t}\boldsymbol{\alpha}T^{i-1})\right) \\ &\quad \cdot (I_{MN} + [(I_{MN} - Y^*(n))^{-1} \otimes I_N]f(\mathscr{Y}(n))) \\ &= \sum_{i=0}^{N-1}(S^{i+1} + S^*(I_M - Y^*(n))^{-1}Y_i(n)) \otimes (\mathbf{t}\boldsymbol{\alpha}T^i). \end{aligned}$$

Therefore $R_q(n+1) = f(\mathscr{X}(n+1))$ and $\mathscr{X}(n+1)$ satisfy (7).

Since the matrix sequence $\{R_q(n)\}$ converges increasingly to R_q, for fixed i the matrix sequence $\{X_i(n)\}$ also increasingly converges. Consequently, $R_q = \lim_{n\to\infty} R_q(n) = \sum_{i=1}^{N-1}(\lim_{n\to\infty} X_i(n)) \otimes (\mathbf{t}\boldsymbol{\alpha}T^i)$. □

We now apply Theorem 3 to iteration (3) and develop an algorithm to calculate R_q. Instead of forming $R_q(n)$ explicitly, we calculate $\mathscr{X}(n)$ at each iteration. According to (6), it needs to compute $\hat{Y}_i(n) = S^*(I_M - Y^*(n))^{-1}Y_i(n)$, $1 \le i \le N-1$, at the n-th iteration. Multiplying by $S^*(I_M - Y^*(n))^{-1}$ on both sides of (6) we obtain the successive recurrence $\hat{Y}_i(n) = \hat{Y}_{i-1}(n)S + S^*(I_M - Y^*(n))^{-1}(X_{i-1}(n)\mathbf{s})\boldsymbol{\beta}$. Since $\hat{Y}_{i-1}(n)S$ can be explicitly formed, it requires $O(M^2(M+N))$ flops to compute these $\hat{Y}_i(n)$, $1 \le i \le N-1$.

The following is the algorithm to compute R_q:

Algorithm 1.

(0) Set stopping tolerance ϵ
(1) $S^* \leftarrow a_1 S + a_2 S^2 + \cdots + a_K S^K$
(2) $X_i^{\text{new}} \leftarrow 0$, $i = 0, 1, \ldots, N-1$
(3) Do
(4) $\quad X_i^{\text{old}} \leftarrow X_i^{\text{new}}$, $i = 0, 1, \ldots, N-1$
(5) $\quad Y_0 \leftarrow \mathbf{s}\boldsymbol{\beta}$
(6) $\quad$ For $i = 1{:}\ N-1$
(7) $\quad\quad Y_i \leftarrow Y_{i-1}S + X_{i-1}^{\text{old}}\mathbf{s}\boldsymbol{\beta}$
(8) $\quad$ End
(9) $\quad Y^* \leftarrow a_1 Y_0 + a_2 Y_1 + \cdots + a_N Y_{N-1}$
(10) $\quad Y^* \leftarrow S^*(I_{MN} - Y^*)^{-1}$
(11) $\quad Y_0 \leftarrow Y^* Y_0$
(12) $\quad$ For $i = 1{:}\ N-1$
(13) $\quad\quad Y_i \leftarrow Y_{i-1}S + Y^*(X_{i-1}^{\text{old}}\mathbf{s})\boldsymbol{\beta}$
(14) $\quad$ End
(15) $\quad X_i^{\text{new}} \leftarrow S^{i+1} + Y_i$, $i = 0, 1, \ldots, N-1$
(16) Until $\max_{i,j,k} |(X_i^{\text{new}})_{kl} - (X_i^{\text{old}})_{kl}| \le \epsilon$

The storage for Algorithm 1 is $O(M^2N)$ bytes and the computational load for each iteration (steps 4–15) is $O(M^2(N+M))$ flops. The logarithmic reduction iteration (Latouche and Ramaswami 1993) is a quadratic-convergent algorithm and can be used to calculate R_q in our problem. It requires computation of the inverse of an $MN \times MN$ matrix at each iteration and this takes $O(M^3N^3)$ flops. In most instances Algorithm 1 seems to go faster although for nearly saturated queues, this may not hold.

Algorithm 1 produces matrix sequence $\mathscr{X} = \{X_i,\ 0 \le i \le N-1\}$ such that $R_q = f(\mathscr{X})$. We now discuss how to solve boundary equations (1) and (2). Noting that $B_0 = \boldsymbol{\beta} \otimes (\mathbf{t}\boldsymbol{\alpha}) = \mathbf{t}(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$, we have

$$\begin{aligned} \boldsymbol{\pi}_0 &= \boldsymbol{\pi}_0 H B_2 (I_N - B_1)^{-1} \\ &= \pi_{01}(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})(I_{MN} - S \otimes T)^{-1} \\ &\quad \cdot [I_{MN} - \tilde{R}_q(I_{MN} - S \otimes T)^{-1}]^{-1}(I_N - T)^{-1}, \end{aligned}$$

where π_{01} is the first entry of $\boldsymbol{\pi}_0$ and $\tilde{R}_q = \mathbf{s}\boldsymbol{\beta} \otimes \mathbf{t}\boldsymbol{\alpha} + R_q(\mathbf{s}\boldsymbol{\beta} \otimes T)$. We know that $(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})(I_{MN} - S \otimes T)^{-1}$ and $(I_N - T)^{-1}$ can be explicitly formed. By Lemma 2 and the argument in the proof of Theorem 3, it is easy to calculate $[I_{MN} - \tilde{R}_q(I_{MN} - S \otimes T)^{-1}]^{-1}$. Therefore, $\boldsymbol{\pi}_0$ can be efficiently computed. We can obtain π_{01} by solving (2). As it is easy to compute H and $(I_{MN} - R_q)^{-1}$, this equation can be solved efficiently.

4. The Waiting Time

In this section, we develop the Markov chain for the waiting time and present an efficient method to compute its stationary distribution.

At time j, given that there are customers in the system, let L_j be how long a customer who is receiving service has been in the system, J_j be this customer's remaining service time, and let K_j be the phase of arrival at time $j - L_j$. Let this customer be labeled $\mathscr{C}$. Specifically, K_j is the remaining time at time $j - L_j$ before the arrival of customer $\mathscr{C}+1$. If there is no customer in the queue, denote by K_j the remaining time

at time j before the next customer arrives. Consider the state space

$$\Delta_w = \{(0, K_j) \cup (L_j, J_j, K_j), L_j \geq 0,\ J_j = 1, 2, \ldots, M,\ K_j = 1, 2, \ldots, N\}.$$

Note that the states $(0, K_j)$ refer to when there is no customer in the system.

It is immediately clear that this is a Markov chain with transition matrix

$$P_w = \begin{bmatrix} D_1 & D_0 & & & \\ D_2 & C_1 & C_0 & & \\ D_3 & C_2 & C_1 & C_0 & \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ D_N & C_{N-1} & C_{N-2} & C_{N-3} & \ddots \\ & C_N & C_{N-1} & C_{N-2} & \ddots \\ & & C_N & C_{N-1} & \ddots \\ & & & C_N & \ddots \\ & & & & \ddots \end{bmatrix},$$

where $D_0 = \boldsymbol{\beta} \otimes (\mathbf{t}\boldsymbol{\alpha})$, $D_1 = T$, $D_i = \mathbf{s} \otimes T^{i-1}$, $i \geq 2$ and $C_0 = S \otimes I_N$, $C_i = (\mathbf{s}\boldsymbol{\beta}) \otimes T^{i-1}(\mathbf{t}\boldsymbol{\alpha})$, $i \geq 1$.

Let $\boldsymbol{\nu}$ be the stationary distribution of P_w and be partitioned as $\boldsymbol{\nu} = [\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots]$, then $\boldsymbol{\nu}_n = \boldsymbol{\nu}_1 R_w^{n-1}$, $n \geq 1$, where R_w is the minimal nonnegative solution to the matrix equation $R_w = \sum_{i=0}^{N} R_w^i C_i$. The distribution $\boldsymbol{\nu}_0$ satisfies

$$\boldsymbol{\nu}_0 (D_1 + D_0 (I_{MN} - \bar{C})^{-1} \bar{D}) = \boldsymbol{\nu}_0, \tag{10}$$

$$\boldsymbol{\nu}_0 \mathbf{1} + \boldsymbol{\nu}_0 D_0 (I_{MN} - \bar{C})^{-1} (I_{MN} - R_w)^{-1} \mathbf{1} = 1, \tag{11}$$

and $\boldsymbol{\nu}_1 = \boldsymbol{\nu}_0 (I_{MN} - \bar{C})^{-1}$. Here $\bar{D} = D_2 + R_w D_3 + R_w^2 D_4 + \cdots + R_w^{N-2} D_N$ and $\bar{C} = C_1 + R_w C_2 + R_w^2 C_3 + \cdots + R_w^{N-1} C_N$.

Let W be the customer's waiting time in the system. Then

$$\Pr\{W = i\} = (1 - \boldsymbol{\nu}_0 \mathbf{1})^{-1} \sum_{k=0}^{\min\{i-1, M-1\}} \boldsymbol{\nu}_{i-k} (\mathbf{e}_{k+1} \otimes \mathbf{1}), \quad i \geq 1,$$

where $\mathbf{e}_{k+1}$ is column $k+1$ of I_M.

The following iteration can be used to calculate R_w:

$$\begin{aligned} R_w(0) &= 0, \\ R_w(n+1) &= C_0 [I_{MN} - C_1 - R_w(n) C_2 - \cdots \\ &\qquad - R_w^{N-1}(n) C_N]^{-1}. \end{aligned} \tag{12}$$

The following result characterizes the structure of $R_w(n)$.

Theorem 4. *Let the matrix sequence* $\{R_w(n), n \geq 1\}$ *be generated by iteration* (12). *Then,* $R_w(n)$ *assumes the form* $R_w(n) = S \otimes I_n + \mathbf{r}(n)(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$, *for* $n \geq 1$. *Denote* $\mathbf{z}(n) = \sum_{i=0}^{N-1} R_w^i(n)(\mathbf{s} \otimes T^i \mathbf{t})$. *Then* $\mathbf{r}(n+1)$ *is determined from* $R_w(n)$ *by*

$$\mathbf{r}(n+1) = \frac{1}{1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{z}(n)} (S \otimes T)\mathbf{z}(n). \tag{13}$$

Moreover, R_w *assumes the form* $R_w = S \otimes I_N + \mathbf{r}(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$ *with* $\mathbf{r} = \lim_{n \to \infty} \mathbf{r}(n)$.

Proof. Noting that for $i \geq 1$, $C_i = (\mathbf{s} \otimes T^{i-1}\mathbf{t})(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$, we have $C_1 + R_w(n) C_2 + \cdots + R_w^{N-1}(n) C_N = \mathbf{z}(n)(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$. It follows from

$$\begin{aligned} R_w(n+1) &= (S \otimes I_N)(I_{MN} - \mathbf{z}(n)(\boldsymbol{\beta} \otimes \boldsymbol{\alpha}))^{-1} \\ &= (S \otimes I_N)\left(I_{MN} + \frac{1}{1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{z}(n)} \mathbf{z}(n)(\boldsymbol{\beta} \otimes \boldsymbol{\alpha}) \right) \\ &= S \otimes I_N + \frac{1}{1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{z}(n)} (S \otimes T)\mathbf{z}(n)(\boldsymbol{\beta} \otimes \boldsymbol{\alpha}) \end{aligned}$$

that (13) holds.

Since $R_w(n)$ increasingly converges to R_w, $\mathbf{r}(n)$ also increasingly converges. Then $R_w = \lim_{n \to \infty} R_w(n) = S \otimes I_N + (\lim_{n \to \infty} \mathbf{r}(n))(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$. □

Applying Horner's Algorithm to the computation of $\mathbf{z}(n) = \sum_{i=0}^{N-1} R_w^i(n)(\mathbf{s} \otimes T^i \mathbf{t})$ at each iteration, we develop the following algorithm to calculate R_w:

Algorithm 2.
(0) Set stopping tolerance ϵ
(1) $\mathbf{r}^{\text{new}} = (1/(1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})(\mathbf{s} \otimes \mathbf{t})))(S \otimes I_N)(\mathbf{s} \otimes \mathbf{t})$
(2) Do
(3) $\quad \mathbf{r}^{\text{old}} \leftarrow \mathbf{r}^{\text{new}}$
(4) $\quad \mathbf{z} \leftarrow \mathbf{s} \otimes T^{N-1}\mathbf{t}$
(5) $\quad$ For $i = N-1, N-2, \ldots, 1$
(6) $\quad\quad \mathbf{z} \leftarrow (S \otimes I_N)\mathbf{z} + [(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{z}]\mathbf{r}^{\text{old}} + \mathbf{s} \otimes T^i \mathbf{t}$
(7) $\quad$ End
(8) $\quad \mathbf{r}^{\text{new}} \leftarrow (1/(1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{z}))(S \otimes I_N)\mathbf{z}$
(9) Until $\max_i |(\mathbf{r}^{\text{new}})_i - (\mathbf{r}^{\text{old}})|_i \leq \epsilon$

Each iteration of Algorithm 2 costs $O(MN^2)$ flops. Clearly, $\bar{C} = \mathbf{w}(\boldsymbol{\beta} \otimes \boldsymbol{\alpha})$ with $\mathbf{w} = \sum_{i=0}^{N-1} R_w^i (\mathbf{s} \otimes T^i \mathbf{t})$. Thus,

$$(I_{MN} - \bar{C})^{-1} = I_{MN} + \frac{1}{(1 - (\boldsymbol{\beta} \otimes \boldsymbol{\alpha})\mathbf{w})} \mathbf{w}(\boldsymbol{\beta} \otimes \boldsymbol{\alpha}).$$

It is straightforward to check that

$$\begin{aligned} &(I_{MN} - R_w)^{-1} \\ &\quad = (I_M - S)^{-1} \otimes I_N + \frac{1}{1 - \boldsymbol{\mu}\mathbf{r}} [(I_M - S)^{-1} \otimes I_N] \mathbf{r}\boldsymbol{\mu}, \end{aligned} \tag{14}$$

where $\boldsymbol{\mu} = (\boldsymbol{\beta}(I_M - S)^{-1}) \otimes \boldsymbol{\alpha}$. Therefore, the inverses of $I_{MN} - \bar{C}$ and $I_{MN} - R_w$ are easy to compute. Consequently, the boundary equations (10) and (11) can be efficiently solved.

5. Computing Decay Rates

In many applications, the tail distributions for queue length and waiting time are of interest. Let λ_q and λ_w

be the Perron eigenvalues of R_q and R_w, respectively. According to Neuts (1986), the tail distributions of queue length and waiting time have geometric tails, i.e., $\Pr\{L=k\}\sim\lambda_q^k$ and $\Pr\{W=k\}\sim\lambda_w^k$, where L and W represent the queue length and the waiting time, respectively, and $x_k\sim y_k$ means the ratio x_k/y_k tends to a constant as $k\to\infty$. The quantities λ_q and λ_w are referred to as *asymptotic decay rates*. Let $A_q(z)=A_0+zA_1+z^2A_2=(S+z\mathbf{s\beta})\otimes(\mathbf{t\alpha}+zT)$ and $C_w(z)=\sum_{k=0}^{N}z^kC_k=S\otimes I_N+(\mathbf{s\beta})\otimes(\mathbf{b}(z)\boldsymbol{\alpha})$, where $\mathbf{b}(z)=[z\ z^2\ \cdots\ z^N]^T$. For $z\geq 0$, denote by $\lambda_q(z)$ and $\lambda_w(z)$ the Perron eigenvalues of $A_q(z)$ and $C_w(z)$, respectively. According to Neuts (1981), λ_q and λ_w are the unique solutions to the equations $z=\lambda_q(z)$, $0<z<1$, and $z=\lambda_w(z)$, $0<z<1$, respectively. We can use bisection to solve these two equations, and at each iteration we need to calculate $\lambda_q(z)$ and $\lambda_w(z)$. We know that $\lambda_q(z)=\lambda_s(z)\lambda_t(z)$, where $\lambda_s(z)$ and $\lambda_t(z)$ are the Perron eigenvalues of $S+z\mathbf{s\beta}$ and of $\mathbf{t\alpha}+zT$, respectively. It is easy to check that $\lambda_s(z)$ and $\lambda_t(z)$ are the solutions to the equation $h(1/x)=1/z$, x>0, and $g(z/x)=z$, $x>0$, with respect to x, respectively. Here, where $h(x)=s_1x+s_2x^2+\cdots+s_Mx^M$ and $g(x)=a_1x+a_2x^2+\cdots+a_Nx^N$. Therefore, $\lambda_q(z)$ can be efficiently calculated. It can be shown that $\lambda_w(z)$ is the unique solution to the equation $h(1/x)=1/g(z)$, $x>0$, which implies that $\lambda_w(z)$ is also easy to compute.

6. Summary of Numerical Experiments

We carried out several numerical experiments to test the advantage of incorporating the special structures of R_q and R_w into computations. Exploiting the special structures of R_q and R_w significantly improved the speed of computations and this becomes more pronounced as M and N increase, and also as the traffic intensity increases. Another advantage of exploiting the special structures is that it enlarges the scale of M and N for which we can practically compute the rate matrices.

Acknowledgments

The authors thank the referee for valuable comments that greatly improved this paper. The first author's research was partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada. The second author's research was partially funded by grants from the National Science Foundation of China, the Program for New Century Excellent Talents in Universities of China, and the Shanghai Pujiang Program.

References

Alfa, A. S. 2003. The combined elapsed time and matrix-analytic approach for the GI/G/1 and the GI^X/G/1 systems. *Queueing Systems* **45** 5–25.

Alfa, A. S. 2004. Markov chain representations of discrete distributions applied to queueing models. *Comput. Oper. Res.* **33** 2365–2385.

Alfa, A. S., W. Li. 2001. Matrix-geometric method for the discrete time GI/G/1 system. *Stochastic Models* **17** 541–554.

Dafermos, S., M. F. Neuts. 1971. A single server queue in discrete time. *Cahiers du Centre Recherche Opérationnelle* **13** 23–40.

Latouche, G., V. Ramaswami. 1993. A logarithmic reduction algorithm for quasi-birth-death process. *J. Appl. Probab.* **30** 650–674.

Neuts, M. F. 1981. *Matrix-Geometric Solutions in Stochastic Models*. John Hopkins University Press, Baltimore, MD.

Neuts, M. F. 1986. The caudal characteristic curve of queues. *Adv. Appl. Probab.* **18** 221–254.