A Matrix Decomposition Approach for Solving State Balance Equations of A Phase-type Queueing Model with Multiple Servers

Yuan-Heng Yen
Department of Mathematical Sciences
National Chengchi University
No. 64, Sec. 2, ZhiNan Rd., Wenshan Dist.,
Taipei 11605, Taiwan (R.O.C.)
ufo63@ms18.hinet.net

Hsing Paul Luh
Department of Mathematical Sciences
National Chengchi University
No. 64, Sec. 2, ZhiNan Rd., Wenshan Dist.,
Taipei 11605, Taiwan (R.O.C.)
slu@nccu.edu.tw

Chia-Hung Wang
Department of Mathematical Sciences
National Chengchi University
No. 64, Sec. 2, ZhiNan Rd., Wenshan Dist.,
Taipei 11605, Taiwan (R.O.C.)
jhwang728@hotmail.com

ABSTRACT

Stationary probabilities are fundamental in response to various measures of performance in queueing networks. Solving stationary probabilities in Quasi-Birth-and-Death(QBD) with phase-type distribution normally are dependent on the structure of the queueing network. In this paper, a new computing scheme is developed for attaining stationary probabilities in queueing networks with multiple servers. This scheme provides a general approach of considering the complexity of computing algorithm. The result becomes more significant when a large matrix is involved in computation. The background theorem of this approach is proved and provided with an illustrative example in this paper.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*queueing theory, Markov processes*

General Terms

Performance

Keywords

Phase-type distribution, multiple servers, stationary probability.

1. INTRODUCTION

The Markovian arrival process (MAP) is a generalization of the Poisson process, where the arrivals are governed by a Markov chain [10]. We consider a semi $MAP/M/n$ queueing system, where customers arrive at the system according to a phase-type process but may leave the system without services. The family of phase-type distributions is widely used in algorithmic probability [5]. A continuous time phase-type distribution is the distribution of the time until absorption in an absorbing Markovian process. We assume the interarrival time follows a typical MAP but the arrival rate is smaller since the renege occurs. All n servers of the system are identical, and their service times are independent and identically distributed (i.i.d.) random variables following exponential distributions. Each incoming customer receives service immediately if he/she finds an idle server upon arrival.

Although $MAP/M/n$ queues have been studied extensively by many researchers, analytical solutions for the stationary probability have not yet been studied comprehensively in the literature [5]. In this paper, we study the stationary distribution of such a semi $MAP/M/n$ queueing system with multiple servers. We compute the stationary probability by applying the matrix geometric procedure in [8], which will be combined with Ramaswami's formula [7] and block LU factorization [6] in this paper. The main contribution in the paper is to present a matrix decomposition approach for the stationary probability in a phase-type $MAP/M/n$ queueing model. Through solving the system of submatrices by using Matrix-Geometric Method, we obtain the stationary probability.

Matrix analytic methods are popular as modeling tools because they give one the ability to construct and analyze a wide class of queueing models in a unified and algorithmically tractable way [7]. The Matrix-Geometric Method [5, 8] relies on identifying two parts within the structure of the underlying continuous time Markov chain, including the initial/boundary part and the repetitive part. The initial part has a non-regular structure and each component in it must be represented in detail [8]. The repetitive part has a regular structure and can be represented in stochastic process algebras as a composition of several components. In Matrix-Geometric Method, the infinitesimal generator matrix is decomposed into submatrices, with each one of them representing the transition rates in a particular area within a given part, or between them [5, 8]. The size of the state space would be reasonably small compared with the size of the infinitesimal generator matrix of the Markovian process even if the system is infinite [3].

The remainder of the paper is organized as follows. Sec-

QTNA 2011, Aug. 23-26, 2011, Seoul, Korea

tion 2 introduces a queueing model with phase-type Markovian arrival process. In Section 3, we present a matrix decomposition approach for the stationary probability in a phase-type $MAP/M/n$ queueing model by applying the Matrix-Geometric Method combined with Ramaswami's formula and LU factorization. Numerical results of $MAP/M/n$ queueing systems with multiple servers are given in Section 4, and numerical results of the stationary distribution are compared with approximation methods and simulations. Concluding remarks are to be given in Section 5.

2. PROBLEM DEFINITIONS

2.1 Markovian arrival process with phase-type distributions

We consider a single queueing station and model the queueing network as a semi $MAP/M/n$ queue shown in Fig. 1, where n servers are all identical. The mean service times of each server is exponentially distributed with rate μ. Let $\mathbf{S}_1$ and $\mathbf{S}_{1o}$ represent a transition of service that customer stays with the server and finishes the service, individually, i.e.,

$$\mathbf{S}_1 = \left[\begin{array}{c} -\mu \end{array}\right], \ \mathbf{S}_{1o} = \left[\begin{array}{c} \mu \end{array}\right].$$

The queueing network has two independent and identical arrival streams, where there are two phases for each arrival stream [?]. For the first arrival stream, the time spent in the first phase is exponentially distributed with rate λ_1, and the time spent in the second phase is also exponentially distributed with rate λ_2. Similarly, for the other arrival stream, the time spent in the first phase is exponentially distributed with rate γ_1, and the time spent in the second phase is also exponentially distributed with rate γ_2. After the first phase of arrival stream, the incoming arrival goes to the queueing system (and is to be served) with probability $0 \leq p \leq 1$; otherwise, it jumps to the second phase and then departs directly with probability $(1-p)$. All arrival streams operate in a similar manner.

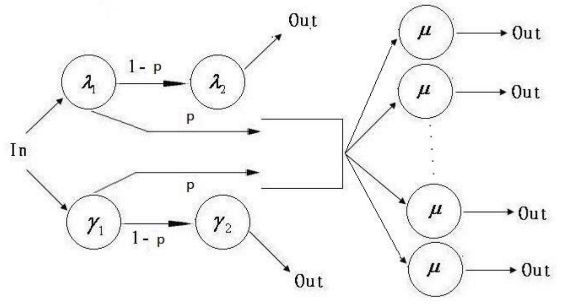

Figure 1: A semi $MAP/M/n$ queueing model

Hence, customers arrive at the system according to a phase-type process with mean arrival rate $\overline{\lambda} > 0$, where the mean arrival rate is defined as

$$\overline{\lambda} = p[(\frac{1}{\lambda_1}p+(\frac{1}{\lambda_1}+\frac{1}{\lambda_2})(1-p)]^{-1}+p[(\frac{1}{\gamma_1}p+(\frac{1}{\gamma_1}+\frac{1}{\gamma_2})(1-p)]^{-1}.$$

These two arrival processes are independent to each other, and parameters are given by $(\lambda_1, p, \lambda_2)$ and (γ_1, p, γ_2), individually. Namely, arrival processes of this queueing model are characterized by

$$\mathbf{T}_1 = \left[\begin{array}{cc} -\lambda_1 & (1-p)\lambda_1 \\ \lambda_2 & -\lambda_2 \end{array}\right], \ \mathbf{T}_{1o} = \left[\begin{array}{c} p\lambda_1 \\ 0 \end{array}\right],$$

$$\mathbf{T}_2 = \left[\begin{array}{cc} -\gamma_1 & (1-p)\gamma_1 \\ \gamma_2 & -\gamma_2 \end{array}\right], \ \mathbf{T}_{2o} = \left[\begin{array}{c} p\gamma_1 \\ 0 \end{array}\right],$$

Note that matrices $\mathbf{T}_m$, for $m = 1, 2$ correspond to phase transitions, and $\mathbf{T}_{mo}$ corresponds to the rate as arrivals enter the system. Both arrival processes are MAP distributed inter-arrival times denoted by $(\mathbf{e}_1, \mathbf{T}_m, \mathbf{T}_{mo})$, for $m = 1, 2$, where $\mathbf{e}_1$ is an 2×1 vector with the first element equals to 1 and another element equals to 0.

The advantage of phase-type distributions is their generality and versatility, which permits the calculation of performance measures of stochastic models with a high degree of accuracy [3]. The Matrix-Geometric Methods allows us to deal with the models whose activities are not necessarily exponentially distributed, while at the same time overcoming the problem of the rapid growth of the state space introduced by the need to explicitly construct the infinitesimal generator matrix of the underlying Markovian process.

The one-step transition matrix embedded in the Markov chain of the arrival process is given by

$$\mathbf{\Phi} = \left[\begin{array}{ccccc} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}\right], \tag{1}$$

where there exist nonnegative off-diagonal and negative diagonal elements in the matrix $\mathbf{B}_{00} = [b_{ij}]$, and the elements of matrix $\mathbf{C} = [c_{ij}]$ are nonnegative. Since $\mathbf{\Phi}$ is the infinitesimal generator of the MAP, we have

$$(\mathbf{B}_{00} + \mathbf{C})\mathbf{1} = 0,$$

where $\mathbf{1}$ is an 4×1 vector with all its elements equal to 1. Since $(\mathbf{B}_{00} + \mathbf{C})$ is the infinitesimal generator, there exists a stationary probability vector

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_{1,1}, \boldsymbol{\theta}_{1,2}, \boldsymbol{\theta}_{2,1}, \boldsymbol{\theta}_{2,2}),$$

where $\boldsymbol{\theta}_{i,j}$ is the stationary probability that an arrival is in the i-th phase of the first stream and the other arrival is in the j-th phase of the second stream. The repetition of the state transitions for vector processes implies a geometric form where scalars are replaced by matrices. Such Markovian processes are called Matrix-Geometric processes. To determine the stationary probability, we need to solve the following balance equations

$$\boldsymbol{\theta}(\mathbf{B}_{00} + \mathbf{C}) = \mathbf{0}, \ \boldsymbol{\theta}\mathbf{1} = 1.$$

In the following section, we recall a special phase-type distributions.

2.2 A Phase-type queueing model

In general, the embedded Markov chain is ergodic if the stability condition of the system is $\overline{\lambda}/(n\mu) < 1$.

LEMMA 1. *Given the mean arrival rate $\overline{\lambda} > 0$ and $\overline{\lambda}/(n\mu) < 1$, the effective range of p is $0 \leqslant p < w$, where*

$$w = \min\{1, \frac{-b - \sqrt{b^2 - 4ac}}{2a}\},$$

$$a = \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + n\mu\lambda_1\gamma_1,$$

$$\begin{aligned} b &= -(\lambda_1\lambda_2\gamma_2 + \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + \gamma_1\gamma_2\lambda_2 \\ &\quad + n\mu\lambda_2\gamma_1 + 2n\mu\lambda_1\gamma_1 + n\mu\lambda_1\gamma_2), \end{aligned}$$

and

$$c = n\mu(\lambda_1 + \lambda_2)(\gamma_1 + \gamma_2).$$

PROOF. Because $\overline{\lambda}/(n\mu) < 1$, we have

$$\begin{aligned} \overline{\lambda} &= p[(\frac{1}{\lambda_1}p + (\frac{1}{\lambda_1} + \frac{1}{\lambda_2})(1-p)]^{-1} \\ &\quad + p[(\frac{1}{\gamma_1}p + (\frac{1}{\gamma_1} + \frac{1}{\gamma_2})(1-p)]^{-1} \\ &< n\mu. \end{aligned} \tag{2}$$

It implies that

$$\frac{\lambda_1 p}{\lambda_1 + \lambda_2 - \lambda_1 p} + \frac{\gamma_1 p}{\gamma_1 + \gamma_2 - \gamma_1 p} < n\mu. \tag{3}$$

By using the form $ap^2 + bp + c > 0$, we can combine the above inequality, and then solve the inequality. It gives

$$p > \frac{-b + \sqrt{b^2 - 4ac}}{2a},$$

or

$$p < \frac{-b - \sqrt{b^2 - 4ac}}{2a},$$

where $a = \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + n\mu\lambda_1\gamma_1$,

$$\begin{aligned} b &= -(\lambda_1\lambda_2\gamma_2 + \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + \gamma_1\gamma_2\lambda_2 \\ &\quad + n\mu\lambda_2\gamma_1 + 2n\mu\lambda_1\gamma_1 + n\mu\lambda_1\gamma_2), \end{aligned}$$

and $c = n\mu(\lambda_1 + \lambda_2)(\gamma_1 + \gamma_2)$. Because the probability p satisfies $0 \leqslant p \leqslant 1$, we have $0 \leqslant p < w$ if $w = \min\{1, \frac{-b-\sqrt{b^2-4ac}}{2a}\}$. $\square$

Let $A(t)$ denote the number of customers arriving in $(0, t]$ and $J(t)$ be the state of the Markov chain at time t with state space $\{(1,1),(1,2),(2,1),(2,2)\}$. Then $\{A(t), J(t)\}$ is a three-dimensional Markovian process with state space $\{(k,i,j) : k \geq 0,\ i,j = 1,2\}$, where k is the number of customers in the system, i is the phase of the first arrival stream, and j is the phase of the second arrival stream.

The state $\{(k,1,1),(k,1,2),(k,2,1),(k,2,2)\}$ is called the level k of the system, for $k \geqslant 0$. Then, there exists an integer n such that the levels 0 up to $n-1$ from the boundary, and those for $k \geqslant n$ are repeating. Transitions between the repeating states have the property that the rates from (k,i,j) to the state $(k+v, i', j')$ for $0 \leqslant v \leqslant \infty$ and $i', j' = 1, 2$ are independent of the value k for $k \geqslant n$. From that n onwards, the behavior of the system for all $k \geqslant n$ is the same as the behavior of the system for n, where k is the number of queued customers. Such similarity need not for $(0, 1, \cdots, n-1)$. We define the vector of probabilities that there are k customers in the system as

$$\begin{aligned} \boldsymbol{\pi}_k &= \lim_{t\to\infty} Pr\{A(t) = k, J(t) = (i,j)\} \\ &= (\pi_{k,1,1}\ \pi_{k,1,2}\ \pi_{k,2,1}\ \pi_{k,2,2}), \end{aligned} \tag{4}$$

where $\boldsymbol{\pi}$ can be partitioned into blocks which correspond to state 0, state 1, state 2, etc., e.g., $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots)$.

Recall that the Kronecker product of any two matrices $\mathbf{L}$ and $\mathbf{M}$ is defined as

$$\mathbf{L} \otimes \mathbf{M} = [\mathbf{L}_{ij}\mathbf{M}]$$

for all i, j. In addition, the Kronecker sum of any two matrices $\mathbf{L}$ and $\mathbf{M}$ is given by

$$\mathbf{L} \oplus \mathbf{M} = \mathbf{L} \otimes \mathbf{I}_M + \mathbf{I}_L \otimes \mathbf{M}.$$

By applying Kronecker matrix operations, Then we obtain

$$\mathbf{B}_{00} = \mathbf{T}_1 \oplus \mathbf{T}_2$$

and

$$\mathbf{C} = (\mathbf{T}_{1o} \otimes \mathbf{e}_1^T) \oplus (\mathbf{T}_{2o} \otimes \mathbf{e}_1^T).$$

Using the arrival and service process parameters in terms of the Kronecker product and sum. We obtain submatrices $\mathbf{A}_{(i)(i-1)}$, $\mathbf{A}$. An arrival is in the server, finishes the service, and departs the system.

$$\mathbf{A}_{(i)(i-1)} = \mathbf{I}_T \otimes (\underbrace{\mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}}_{i}),$$

for $1 \leqslant i \leqslant n-1$ and

$$\mathbf{A} = \mathbf{I}_T \otimes (\underbrace{\mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}}_{n}),$$

where $\mathbf{I}_T$ is an identity matrix of dimensions equal to the sum of the dimensions of the two arrival processes, i.e., $\mathbf{I}_T = \mathbf{I}_{4\times4}$.

Next, we define submatrices $\mathbf{B}_{ii}$, and $\mathbf{B}$ as follows, where the internal phase changes for the composite arrival process. That is,

$$\mathbf{B}_{ii} = \mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \underbrace{\mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_1}_{i},$$

for $0 \leqslant i \leqslant n-1$,

$$\mathbf{B} = \mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \underbrace{\mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_1}_{n},$$

and

$$\mathbf{C} = (\mathbf{T}_{1o} \otimes \mathbf{e}_1^T) \oplus (\mathbf{T}_{2o} \otimes \mathbf{e}_1^T),$$

where $\mathbf{C}$ represents an arrival goes into the queueing system.

Hence, in our queueing model, there exists the infinitesimal generator matrix of a continuous time Markovian process with the structure $\mathbf{Q} =$

$$\begin{bmatrix} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \ddots & \cdots & \mathbf{0} & \mathbf{0} & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{A} & \mathbf{B} & \mathbf{C} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{5}$$

where n is the number of servers in the system. The matrix $\mathbf{Q}$ is composed of submatrices.

3. MATRIX-GEOMETRIC SOLUTIONS

3.1 State balance equations

The stationary probabilities for the queue satisfy $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, $\boldsymbol{\pi}\mathbf{1} = 1$, and $\boldsymbol{\pi} \geqslant \mathbf{0}$. We can find the $\boldsymbol{\pi}_i$'s by solving the following state balance equations (6)-(10):

$$\boldsymbol{\pi}_0\mathbf{B}_{00} + \boldsymbol{\pi}_1\mathbf{A}_{10} = \mathbf{0}, \tag{6}$$

$$\boldsymbol{\pi}_0\mathbf{C} + \boldsymbol{\pi}_1\mathbf{B}_{11} + \boldsymbol{\pi}_2\mathbf{A}_{21} = \mathbf{0}, \tag{7}$$

$$\boldsymbol{\pi}_1 C + \boldsymbol{\pi}_2\mathbf{B}_{22} + \boldsymbol{\pi}_3\mathbf{A}_{32} = \mathbf{0}, \tag{8}$$

$$\vdots$$

$$\boldsymbol{\pi}_{n-2}\mathbf{C} + \boldsymbol{\pi}_{n-1}\mathbf{B}_{(n-1)(n-1)} + \boldsymbol{\pi}_n\mathbf{A}_{(n)(n-1)} = \mathbf{0}. \tag{9}$$

The equation for the repeating states of the process is given by:

$$\boldsymbol{\pi}_{i-1}\mathbf{C} + \boldsymbol{\pi}_i\mathbf{B} + \boldsymbol{\pi}_{i+1}\mathbf{A} = \mathbf{0}, \quad i = n, n+1, n+2, \cdots . \tag{10}$$

Using (10), the matrix geometric procedure gives the vector solution $\boldsymbol{\pi}_{n+k-1} = \boldsymbol{\pi}_{n-1}\mathbf{R}^k$, for $k = 0, 1, 2, \cdots$, where $\mathbf{R}$ is the matrix solution of the equation $\mathbf{C} + \mathbf{RB} + \mathbf{R}^2\mathbf{A} = \mathbf{0}$. Neuts [8] showed that the iteration

$$\mathbf{R}_{k+1} = -(\mathbf{C} + \mathbf{R}_k^2\mathbf{A})\mathbf{B}^{-1}$$

converges to the solution $\mathbf{R}$ starting with $\mathbf{R}_0 = \mathbf{0}$.

We rewrite the above equations (6)-(10) in matrix form as follows

$$\left[\begin{array}{ccccc} \boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{n-1} & \boldsymbol{\pi}_n \end{array}\right] \cdot \mathbf{Q}_1 = \mathbf{0}, \tag{11}$$

where $\mathbf{Q}_1 =$

$$\left[\begin{array}{ccccccc} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \mathbf{C} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{(n-1)(n-2)} & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{A} & \mathbf{B}+\mathbf{RA} \end{array}\right].$$

In addition, by using the normalization condition, we obtain

$$\boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \cdots + \boldsymbol{\pi}_n(\mathbf{I} - \mathbf{R})^{-1} = 1. \tag{12}$$

Then the solution for the probabilities $\boldsymbol{\pi}_0, \boldsymbol{\pi}_1 ..., \boldsymbol{\pi}_n$ can be determined by

$$\left[\begin{array}{ccccc} \boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{n-1} & \boldsymbol{\pi}_n \end{array}\right] \cdot \mathbf{Q}_2 = [1, \mathbf{0}], \tag{13}$$

where $\mathbf{Q}_2 =$

$$\left[\begin{array}{ccccccc} \mathbf{1} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ (\mathbf{I}-\mathbf{R})^{-1}\cdot\mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{A} & \mathbf{B}+\mathbf{RA} \end{array}\right]$$

by adding the first column into $\mathbf{Q}_1$.

By the stability assumption, the infinitesimal generator matrix is irreducible. The necessary condition for this is that matrices $\mathbf{B}$ and $\mathbf{B}_{ii}$, for $i = 0, 1, 2, \cdots, n-1$, are nonsingular, which implies that inverses of those matrices can be determined. The computation of the matrix $\mathbf{R}$ is by means of the iterative procedure [5].

The sequence $\{R_k\}_k$ is entry-wise nondecreasing and converges monotonically to a nonnegative matrix $\mathbf{R}$. This follows the fact that $\mathbf{B^{-1}}$ is a nonnegative matrix. The number of iterations needed for convergence increases as the spectral radius of $\mathbf{R}$ increases. We terminate the iteration and return with the solution of $\mathbf{R}$ when

$$\|\mathbf{R}_{k+1} - \mathbf{R}_k\|_\infty \leqslant \varepsilon,$$

where ε is a given small constants.

3.2 An algorithm for matrix decomposition

Consider computing $\boldsymbol{\pi} = (\boldsymbol{\pi}_i)_{i\in N}$, such that $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$. That is,

$$\left[\begin{array}{cccc} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} & \boldsymbol{\pi}_{n+2} & \cdots \end{array}\right] \left[\begin{array}{c|ccccc} \mathbf{B_0} & \mathbf{B_1} & \mathbf{0} & \mathbf{0} & \cdots \\ \hline \mathbf{B}_{-1} & \mathbf{B} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A} & \mathbf{B} & \mathbf{C} & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{array}\right] = \mathbf{0},$$

where $\boldsymbol{\pi}^* = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_n]$,

$$\mathbf{B}_0 = \left[\begin{array}{cccccc} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \ddots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{(n-1)(n-2)} & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{A} & \mathbf{B} \end{array}\right],$$

$$\mathbf{B}_{-1} = \left[\begin{array}{cccc} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A} \end{array}\right]_{4\times 4(n+1)},$$

$$\mathbf{B}_1 = \left[\begin{array}{c} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{C} \end{array}\right]_{4(n+1)\times 4}.$$

It also gives

$$\left[\begin{array}{cccc} \mathbf{B} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{A} & \mathbf{B} & \mathbf{C} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{array}\right] = \mathbf{VW},$$

where

$$\mathbf{V} = \left[\begin{array}{cccc} \mathbf{V}_0 & \mathbf{V}_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_0 & \mathbf{V}_1 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_0 & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{array}\right], \mathbf{W} = \left[\begin{array}{cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ -\mathbf{H} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & -\mathbf{H} & \mathbf{I} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{array}\right].$$

Then we have

$$\left[\begin{array}{cccc} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} & \boldsymbol{\pi}_{n+2} & \cdots \end{array}\right] \left[\begin{array}{c|ccc} \mathbf{B_0} & \mathbf{B_1} & \mathbf{0} & \cdots \\ \hline \mathbf{B}_{-1} & & & \\ \mathbf{0} & & \mathbf{VW} & \\ \vdots & & & \end{array}\right] = \mathbf{0},$$

which is equivalent to

$$\left[\begin{array}{cccc} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} & \boldsymbol{\pi}_{n+2} & \cdots \end{array}\right] \left[\begin{array}{c|ccc} \mathbf{B_0} & \mathbf{B}_1^* & \mathbf{0} & \cdots \\ \hline \mathbf{B}_{-1} & & & \\ \mathbf{0} & & \mathbf{V} & \\ \vdots & & & \end{array}\right] = \mathbf{0}.$$

As we know

$$\mathbf{W}^{-1} = \left[\begin{array}{cccc} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{H} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{H} & \mathbf{I} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{array}\right], \text{and} \left[\begin{array}{cccc} \mathbf{B}_1^* & \mathbf{0} & \mathbf{0} & \cdots \end{array}\right] =$$

$$\left[\begin{array}{cccc} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \cdots \end{array}\right]\mathbf{W}^{-1}.$$

Then, we have $\mathbf{B}_1^* = \mathbf{B}_1 \cdot \mathbf{I}$.

Next, to determine $\mathbf{H}$ and $\mathbf{V}_0$, we solve the following equations:

$$\mathbf{V}_0 - \mathbf{V}_1\mathbf{H} = \mathbf{B}, \tag{14}$$

$$\mathbf{V}_1 = \mathbf{C}, \tag{15}$$

and

$$-\mathbf{V}_0\mathbf{H} = \mathbf{A}. \tag{16}$$

From the first two equations, it yields

$$\begin{bmatrix} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 \\ \mathbf{B}_{-1} & \mathbf{V}_0 \end{bmatrix} = \mathbf{0}.$$

Then, by solving the following equations

$$\boldsymbol{\pi}^*(\mathbf{B}_0 - \mathbf{B}_1\mathbf{V}_0^{-1}\mathbf{B}_{-1}) = 0$$

and

$$\boldsymbol{\pi}_0 \cdot \mathbf{1} + \boldsymbol{\pi}_1 \cdot \mathbf{1} + \cdots + \boldsymbol{\pi}_n(\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1} = 1,$$

we get $\boldsymbol{\pi}^* = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_n]$.

LU factorization

Considering the matrix $\mathbf{Q}_1$. Here, we assume that $\boldsymbol{\pi}^*\mathbf{Q}_1 = \mathbf{0}$. The equations are of the homogeneous system. We use LU factorization to obtain $\boldsymbol{\pi}^*$ in the following steps.

Step 1: Let the first column of $\mathbf{Q}_1$ be replaced by the column vector

$$(\mathbf{1}, \cdots, \mathbf{1}, (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1})^T.$$

Then, the modified $\mathbf{Q}_1$ is rewritten as a new matrix $\mathbf{Q}_3$, and we have

$$\boldsymbol{\pi}^*\mathbf{Q}_3 = \begin{bmatrix} \mathbf{y} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix},$$

where

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

Step 2: If we transpose $\boldsymbol{\pi}^*\mathbf{Q}_3$, it gives

$$(\boldsymbol{\pi}^*\mathbf{Q}_3)^T = \mathbf{Q}_3^T \boldsymbol{\pi}^{*T} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}.$$

Then, we have

$$\begin{bmatrix} \mathbf{B}_{00}^{T\,*} & \mathbf{A}_{10}{}^* & \boldsymbol{\Omega} & \cdots & \boldsymbol{\Omega} & \overline{\boldsymbol{\Omega}} \\ \mathbf{C} & \mathbf{B}_{11}{}^T & \mathbf{A}_{21} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{B}_{22}^T & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_{(n-1)(n-1)}^T & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C} & (\mathbf{B} + \mathbf{R}\mathbf{A})^T \end{bmatrix}.$$

$$\begin{bmatrix} \boldsymbol{\pi}_0 \\ \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \vdots \\ \boldsymbol{\pi}_{n-1} \\ \boldsymbol{\pi}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

where

$$\mathbf{B}_{00}^{T\,*} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ (1-p)\gamma_1 & -\lambda_1 - \gamma_2 & 0 & \lambda_2 \\ (1-p)\lambda_1 & 0 & -\lambda_2 - \gamma_1 & \gamma_2 \\ 0 & (1-p)\lambda_1 & (1-p)\gamma_1 & -\lambda_2 - \gamma_2 \end{bmatrix},$$

$$\mathbf{A}_{10}{}^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix}, \boldsymbol{\Omega} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \overline{\boldsymbol{\Omega}} = \begin{bmatrix} (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Step 3: Applying Gaussian elimination, we transform $\boldsymbol{\Omega}$ and $\overline{\boldsymbol{\Omega}}$ into a zero matrix.

Then it gives

$$\mathbf{Z}_n = \begin{bmatrix} \mathbf{B}_{00}^{T\,**} & \mathbf{A}_{10}{}^{**} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{B}_{11}^T & \mathbf{A}_{21} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{B}_{22}^T & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{B}_{(n-1)(n-1)}^T & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C} & (\mathbf{B} + \mathbf{R}\mathbf{A})^T \end{bmatrix},$$

where $\mathbf{B}_{00}^{T\,**}$, $\mathbf{A}_{10}{}^{**}$, are obtained by Gaussian elimination.

THEOREM 1. *$\mathbf{Z}_n$ is a nonsingular matrix.*

PROOF. By **Step 1**, we know

$$\boldsymbol{\pi}^*\mathbf{Q}_3 = \begin{bmatrix} \mathbf{y} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix},$$

where

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}.$$

The solution of $\boldsymbol{\pi}^*$ is unique by Matrix Geometric Solution, and $\mathbf{Q}_3$ is a nonsingular matrix. We transpose the matrix $\mathbf{Q}_3$ to $\mathbf{Q}_3^T$. By **Step 3**, we determine $\mathbf{Z}_n$. $\mathbf{Q}_3$ is a nonsingular matrix, so is $\mathbf{Z}_n$. □

THEOREM 2. *(Roger and Charles [9]) Let $\mathbf{Z} \in \mathbf{M}_{m\times m}$, a set of $m \times m$ matrices. There exists permutation matrices $\mathbf{D}$, $\mathbf{E} \in \mathbf{M}_{m\times m}$, a lower triangular matrix $\mathbf{L} \in \mathbf{M}_{m\times m}$, and an upper triangular matrix $\mathbf{U} \in \mathbf{M}_{m\times m}$ such that*

$$\mathbf{Z} = \mathbf{DLUE}.$$

If $\mathbf{Z}$ is nonsingular, one may take $\mathbf{E} = \mathbf{I}$ and $\mathbf{Z}$ may be written as

$$\mathbf{Z} = \mathbf{DLU}.$$

PROOF. If rank $\mathbf{Z}$=k, $\mathbf{Z}$ has a k-by-k nonsingular submatrix, which may, by permutation of rows and columns, be permuted into the upper left corner. Now apply Theorem D in Appendix B to the upper left corner and apply Theorem LU in Appendix A to achieve a factorization. If $\mathbf{Z}$ is nonsingular, Theorem D in Appendix B indicates that permutation on the right is unnecessary in order to apply Theorem D in Appendix B, which verifies the second factorization and completes the proof. □

Step 4:

By

$$\mathbf{Z}_n \cdot \boldsymbol{\pi}^{*T} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

and according to above Theorems 1 and 2, we can infer that as follows **Remark** 1 and **Remark** 2. Let $\mathbf{Z}_n(\{i\})$, $i = 1, \cdots, 4(n+1)$ be formed with the first i rows squared matrix of $\mathbf{Z}_n$. $\mathbf{Z}_n(\{1, 2, \cdots, i\})$ denote a series of matrices $\mathbf{Z}_n(\{1\})$, $\mathbf{Z}_n(\{2\})$, $\cdots$, $\mathbf{Z}_n(\{i\})$.

Remark 1. $\mathbf{Z}_n$ is a $4(n+1) \times 4(n+1)$ matrix and nonsingular, and

$$det(\mathbf{Z}_n)(\{1, \cdots, j\}) \neq 0, \ \forall j = 1, \cdots, 4(n+1),$$

which implies $\mathbf{Z}_n = \mathbf{LU}$.

Because $\mathbf{Z}_n = \mathbf{LU}$, we can slove $[\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_n]$ by

$$\mathbf{LU} \cdot \boldsymbol{\pi}^{*T} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

It gives the LU factorization of $\mathbf{Z}_n$ as follows:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_1 & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{L}_{n-1} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{L}_n & \mathbf{I} \end{bmatrix}$$

$$\cdot \begin{bmatrix} \mathbf{U}_0 & \mathbf{F}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 & \mathbf{F}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_2 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{U}_{n-1} & \mathbf{F}_n \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{U}_n \end{bmatrix}.$$

The following algorithm is given for $\mathbf{L}_i$ and $\mathbf{U}_i$:

***Algorithm* LU factorization:**

Input $\mathbf{U}_0 = \mathbf{B}_{00}^{T\ **}$

for $i = 1 : n$

do $\mathbf{L}_i = \mathbf{C}\mathbf{U}_{i-1}^{-1}$

do $\mathbf{U}_i = \mathbf{B}_{ii}^{T*} - \mathbf{L}_i\mathbf{F}_i$

end

After completing the LU factorization, the vector $\boldsymbol{\pi}$ can be obtained via block forward and backward substitution:

***Algorithm* Forward and backward substitution:**

Input $\mathbf{y_0} = [1, 0, 0, 0]^T$

for $i = 1 : n$

do $\mathbf{y}_i = -\mathbf{L}_i y_{i-1}$

end

do $\boldsymbol{\pi}_n = \mathbf{U}_n^{-1}\mathbf{y}_n$

for $i = n-1 : -1 : 0$

do $\boldsymbol{\pi}_i = \mathbf{U}_i^{-1}(\mathbf{y}_i - \mathbf{F}_{i+1}\boldsymbol{\pi}_{i+1})$

end

According the above algorithm, we obtain the stationary probability $\boldsymbol{\pi}^* = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_n]$.

Remark 2. If $\mathbf{Z}_n$ is a $4(n+1) \times 4(n+1)$ matrix and nonsingular with some $1 \leq j \leq 4(n+1)$ such that

$$det(\mathbf{Z}_n)(\{j\}) = 0,$$

then by Theorem 2 there exists a permutation matrix $\mathbf{D} \in \mathbf{M}_{4(n+1)\times 4(n+1)}$ matrix such that

$$det(\mathbf{D}^T\mathbf{Z}_n)(\{1, \cdots, j\}) \neq 0, \quad j = 1, \cdots, 4(n+1)$$

which implies $\mathbf{D}^T\mathbf{Z}_n = \mathbf{LU}$ and $\mathbf{Z}_n = \mathbf{DLU}$.

Because $\mathbf{Z}_n = \mathbf{DLU}$, we can slove $\boldsymbol{\pi}^*$ by

$$\mathbf{DLU} \cdot \boldsymbol{\pi}^{*T} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \Rightarrow \quad \mathbf{LU} \cdot \boldsymbol{\pi}^{*T} = \mathbf{D^T} \cdot \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

After completing the LU factorization, the vector $\boldsymbol{\pi}$ can be obtained via block forward and backward substitution:

***Algorithm* Forward and backward substitution:**

Input $\mathbf{y_0} = [\mathbf{D}^T(\{4\})][1, 0, 0, 0]^T$

for $i = 1 : n$

do $\mathbf{y}_i = -\mathbf{L}_i y_{i-1}$

end

do $\boldsymbol{\pi}_n = \mathbf{U}_n^{-1}\mathbf{y}_n$

for $i = n-1 : -1 : 0$

do $\boldsymbol{\pi}_i = \mathbf{U}_i^{-1}(\mathbf{y}_i - \mathbf{F}_{i+1}\boldsymbol{\pi}_{i+1})$

end

In the above Algorithm, $\mathbf{D}^T(\{4\})$ is the first four rows and columns composed a 4×4 matrix.

According to the above algorithm, we obtain the stationary probability $\boldsymbol{\pi}^* = [\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_n]$.

4. NUMERICAL EXAMPLES

4.1 A queueing model with two servers

In this section we present three sets of numerical examples to demonstrate the matrix decomposition approach for stationary probabilities of phase-type queueing models with multiple servers. First, we consider the system with two servers, where parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, 0.4, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.4, 5)$, and let $\mu = 10$.

The stationary probabilities of the states of the arrival process, $\boldsymbol{\theta}$, are obtained from its Markov arrival process representation. By solving the following equations $\boldsymbol{\theta}(\mathbf{B}_{00} + \mathbf{C}) = \mathbf{0}$ and $\boldsymbol{\theta}\mathbf{1} = 1$ with Matlab [13], we have

$$\boldsymbol{\theta} = (0.1838, 0.4412, 0.1103, 0.2647).$$

For comparison of estimated values $\boldsymbol{\theta}$, we also use a simulation programming of queueing models, Promodel [14]. From the stationary probabilities obtained by simulation with Promodel, the results are shown in Table 1. we have the values (0.1860, 0.4340, 0.1110, 0.2590). We can find that the value is close to $\boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \boldsymbol{\pi}_2 + \boldsymbol{\pi}_3 + \ldots = \boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \boldsymbol{\pi}_2(\mathbf{I} - \mathbf{R})^{-1} = (0.1838, 0.4412, 0.1103, 0.2647)$.

By the matrix geometric procedure of the vector solution, we use $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, $\boldsymbol{\pi}\mathbf{1} = 1$, and $\boldsymbol{\pi} \geqslant \mathbf{0}$ to determine $\boldsymbol{\pi}$. It gives $\boldsymbol{\pi}_0 = (0.0873, 0.2798, 0.0612, 0.1962)$,

$$\boldsymbol{\pi}_1 = (0.0608, 0.1187, 0.0332, 0.0529),$$

and

$$\boldsymbol{\pi}_2 = (0.0230, 0.0295, 0.0105, 0.0106),$$

which are consistent with the numerical results of simulation in Promodel.

With Ramaswami's formula [7], we have

$$\boldsymbol{\pi}_0 = (0.0875, 0.2800, 0.0615, 0.1965),$$

$$\boldsymbol{\pi}_1 = (0.0608, 0.1181, 0.0333, 0.0526),$$

and

$$\boldsymbol{\pi}_2 = (0.0227, 0.0287, 0.0104, 0.0102),$$

which will be compared with simulation, LU approach in the next section. Next, by applying the algorithm of LU factorization, it gives

$$\boldsymbol{\pi}_0 = (0.0872, 0.2794, 0.0611, 0.1959),$$

$$\boldsymbol{\pi}_1 = (0.0607, 0.1185, 0.0332, 0.0528),$$

and

$$\boldsymbol{\pi}_2 = (0.0229, 0.0294, 0.0105, 0.0106).$$

Here, we observe the numerical results of changing values of p, and other variables are fixed. That is, it gives $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, $\mu = 10$, and $0 \leqslant p < 0.8840$, where $p = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6$.

The numerical results are compared in three different approaches, **G** represents matrix geometric procedure, **R** represents Ramaswami's formula, and **LU** represents LU factorization. Let $\bar{\pi}_i$ be the probability of i customers in system, i.e., $\bar{\pi}_i = \boldsymbol{\pi}_i \mathbf{1}$. Table 2 and Table 3 shows the comparison of numerical results.

In order to estimate the $\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ accurately. By simulation in Promodel [14], it gives the queue empty rates which are shown in Table 4. We find that the sum $\sum_{k=0}^{2} \boldsymbol{\pi}_k$ is equal to the queue empty rate obtained with simulations, where $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, $0 \leqslant p < 0.8840$, $p = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6$ and $\mu = 10$. The comparison results are shown in Table 4 .

Consider a classic model of multiple servers for a further comparison and validate our model. Here, we present a numerical example of $M/M/2$ queueing models. The parameters of two arrival processes $(\lambda_1, p, \lambda_2)$ and (γ_1, p, γ_2) are given as $(\lambda_1, p, \lambda_2) = (10, 1, 10)$, $(\gamma_1, p, \gamma_2) = (5, 1, 20)$, $\mu = 10$.

By applying the matrix geometric procedure, it gives the vector solution $\boldsymbol{\pi}_0 = (0.1429, 0.0000, 0.0000, 0.0000)$,

$$\boldsymbol{\pi}_1 = (0.2143, 0.0000, 0.0000, 0.0000),$$

and

$$\boldsymbol{\pi}_2 = (0.1607, 0.0000, 0.0000, 0.0000).$$

Then we take the example as an M/M/2 queueing. We find the probability of idle system is 0.142857. The value is the same as the sum of the $\boldsymbol{\pi}_0$.

In $M/M/2$ queues, we find that the busy rate of the queueing is $(\lambda_1 + \gamma_1)/2\mu = 15/20 = 0.75$. We estimate the values by using $1 - \boldsymbol{\pi}_0 - \frac{1}{2}\boldsymbol{\pi}_1$. The busy rate of the queueing is $1 - 0.1429 - \frac{1}{2} \times 0.2143 = 0.74995 \doteqdot 0.75$. All results are consistent with the standards of the classic model.

4.2 A queueing model with three servers

Here, we present numerical results of queueing systems with three servers, where two arrival processes are given with $(\lambda_1, p, \lambda_2) = (10, 0.4, 10)$ and $(\gamma_1, p, \gamma_2) = (20, 0.4, 5)$, and let $\mu = 10$.

From the matrix geometric procedure, we solve $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$, $\boldsymbol{\pi}\mathbf{1} = 1$, and $\boldsymbol{\pi} \geqslant \mathbf{0}$ to estimate $\boldsymbol{\pi}$. Then, it gives the vector solution $\boldsymbol{\pi}_0 = (0.0888, 0.2839, 0.0622, 0.1989)$,

$$\boldsymbol{\pi}_1 = (0.0621, 0.1203, 0.0339, 0.0534),$$

$$\boldsymbol{\pi}_2 = (0.0240, 0.0297, 0.0109, 0.0102),$$

and

$$\boldsymbol{\pi}_3 = (0.0065, 0.0055, 0.0025, 0.0017).$$

By applying Ramaswami's formula, we have

$$\boldsymbol{\pi}_0 = (0.0888, 0.2838, 0.0622, 0.1989),$$

$$\boldsymbol{\pi}_1 = (0.0621, 0.1203, 0.0339, 0.0533),$$

$$\boldsymbol{\pi}_2 = (0.0240, 0.0297, 0.0109, 0.0102),$$

and

$$\boldsymbol{\pi}_3 = (0.0065, 0.0055, 0.0025, 0.0017).$$

LU factorization

By using LU factorization given in previous section, we obtain

$$\boldsymbol{\pi}_0 = (0.0888, 0.2839, 0.0622, 0.1989),$$

$$\boldsymbol{\pi}_1 = (0.0621, 0.1204, 0.0339, 0.0534),$$

$$\boldsymbol{\pi}_2 = (0.0240, 0.0298, 0.0109, 0.0103),$$

and

$$\boldsymbol{\pi}_3 = (0.0066, 0.0056, 0.0026, 0.0017).$$

We observe the effect of changing p on the numerical results obtained from three methods. Here, we have $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, $\mu = 10$,

$$0 \leqslant p < w = 1, \tag{17}$$

for $p = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. Tables 5-7 show the comparison of numerical results.

We compare the values of $\sum_{k=0}^{3} \boldsymbol{\pi}_k$ with the queue empty rate obtained by using simulation in Promodel. The variable values are given as $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and p varies from 0.2 to 0.9, with $\mu = 10$. Table 8 shows the comparison of numerical results.

We present a numerical example of $M/M/3$ queues to verify the results. Here, two arrival processes $(\lambda_1, p, \lambda_2)$ and (γ_1, p, γ_2) are given with $(\lambda_1, p, \lambda_2) = (10, 1, 10)$, $(\gamma_1, p, \gamma_2) = (5, 1, 20)$, and $\mu = 10$. By applying the matrix geometric procedure, it gives the solution

$$\boldsymbol{\pi}_0 = (0.2105, 0.0000, 0.0000, 0.0000),$$

$$\boldsymbol{\pi}_1 = (0.3158, 0.0000, 0.0000, 0.0000),$$

$$\boldsymbol{\pi}_2 = (0.2368, 0.0000, 0.0000, 0.0000),$$

and

$$\boldsymbol{\pi}_3 = (0.1184, 0.0000, 0.0000, 0.0000).$$

In this example, we find the probability of idle system is 0.210526, which is the same as the sum of the $\boldsymbol{\pi}_0$. In $M/M/3$ queues, we know that the busy rate of the queueing is $(\lambda_1 + \gamma_1)/3\mu = 15/30 = 0.5$. By using $1 - \boldsymbol{\pi}_0 - \frac{2}{3}\boldsymbol{\pi}_1 - \frac{1}{3}\boldsymbol{\pi}_2$, it gives the busy rate of queues as follows

$$1 - 0.2105 - \frac{1}{3} \times 0.3158 - \frac{2}{3} \times 0.2368 \doteqdot 0.5263 \doteqdot 0.5.$$

4.3 A queueing model with twenty servers

Here, we present numerical results of queueing systems with twenty servers. The parameters of two arrival processes are given with $(\lambda_1, p, \lambda_2) = (10, 0.4, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.4, 5)$, and let $\mu = 0.8$. Then, it gives the solutions in Table 9.

According to the above four forms, we can find that the values calculated by three methods are almost equal.

Next, we consider the system with twenty servers, where p varies from 0 to 0.75. Here, parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and let $\mu = 0.8$.

By Lemma 1, we have

$$0 \leqslant p < w = 0.8098, \tag{18}$$

Table 10 shows the comparison of numerical results with three different methods.

We compare the values of $\sum_{k=0}^{20} \boldsymbol{\pi}_k$ with the queue empty rate obtained by using simulation in ProModel. The variable values are given as $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and p varies from 0.4 to 0.75, $\mu = 0.8$. Table 11 shows the comparison of numerical results.

Accorrding to two Figures 5.1-2, we can find that when p is close to its upper bound (0.9206), the relative error becomes large.

4.4 A queueing model with twenty-five servers

Here, we present numerical results of queueing systems with twenty-five servers. The parameters of two arrival processes are given with $(\lambda_1, p, \lambda_2) = (10, 0.5, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.5, 5)$, and let $\mu = 0.9$. Then, it gives the solutions in Table 12.

According to the above three forms, we can find that the values calculated by three methods are almost equal.

Next, we consider the system with twenty-five servers, where p varies from 0 to 0.85. Here, parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and let $\mu = 0.9$.

By Lemma 1, we have

$$0 \leqslant p < w = 0.9206. \tag{19}$$

Table 13 shows the comparison of numerical results with three different methods.

We compare the values of $\sum_{k=0}^{25} \boldsymbol{\pi}_k$ with the queue empty rate obtained by using simulation in ProModel. The variable values are given as $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and p varies from 0.5 to 0.85, $\mu = 0.9$. Table 14 shows the comparison of numerical results.

Accorrding to two Figures 5.3-4, we can find that when p is close to its upper bound (0.9206), the relative error becomes large.

4.5 A queueing model with thirty servers

Here, we present numerical results of queueing systems with thirty servers. The parameters of two arrival processes are given with $(\lambda_1, p, \lambda_2) = (10, 0.6, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.6, 5)$, and let $\mu = 1$. Then, it gives the solutions in Table 15.

According to the above three forms, we can find that the values calculated by three methods are almost equal.

Next, we consider the system with thirty servers, where p varies from 0 to 0.95. Here, parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and let $\mu = 1$.

By Lemma 1, we have

$$0 \leqslant p < w = 1. \tag{20}$$

Table 16 shows the comparison of numerical results with three different methods.

We compare the values of $\sum_{k=0}^{30} \boldsymbol{\pi}_k$ with the queue empty rate obtained by using simulation in ProModel. The variable values are given as $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and p varies from 0.6 to 0.95, $\mu = 1$. Table 17 shows the comparison of numerical results.

Accorrding to two Figures 5.5-6, we can find that when p is close to its upper bound (1), the relative error becomes large.

5. CONCLUSION

In this paper, we present a new computing scheme for the stationary probabilities of a phase-type queueing model with multiple servers. The matrix geometric procedure has been developed by using Ramaswami's formula and blocks LU factorization. With LU factorization, an efficient algorithm for solving stationary probabilities is provided to deal with the complex computation of large matrices due to a large number of system states. Through a number of smaller matrices, the state balance equations of a phase-type $MAP/M/n$ queue are solved. Numerical examples are given to demonstrate the proposed matrix geometric procedure. Performance measures of these models are also illustrated with a number of approximation and simulation results. As the traffic is light, we find that the stationary probabilities obtained from our approaches and simulations are almost the same.

Acknowledgments

The authors thank two anonymous referees for their valuable and constructive comments. This research was supported in part by the National Science Council, Taiwan, R.O.C., under grant numbers NSC-98-2221-E-004-001-MY2.

6. REFERENCES

[1] Bitran, G.R., Dasu, S., Analysis of the $\sum Ph/Ph/1$ queue. *Operations Research*, Vol. 42, No. 1, pp.158–174, 1994.

[2] Bodrog, L., Horvath, A., Telek, M., Moment characterization of matrix exponential and Markovian arrival processes. *Annals of operations Reseach*, to appear, 2008.

[3] Chuan, Y.W., Luh, H., Solving a two-node closed queueing network by a new approach, *International Journal of Information and Management Sciences*, Vol. 16, No. 4, pp. 49–62, 2004.

[4] Curry, G.L., Gautam, N., Characterizing the departure process from a two server Markovian queue: A non-renewal approach, *Proceedings of the 2008 Winter Simulation Conference*, pp. 2075–2082, 2008.

[5] El-Rayes, A., Kwiatkowska, M., Norman, G., Solving infinite stochastic process algebra model through martix-geometric methods, *Proceedings of 7th Process Algebras and Performance Modelling Workshop (PAPM99)*, J. Hillston and M. Silva (Eds.), pp. 41–62, University of Zaragoza, 1999.

[6] Gene H. Golub, Charles F. Van Loan, *Matrix Computations*, 3rd Edition, The Johns Hopkins University Press, 1996.

[7] Latouche, G., Ramaswami, V., *Introduction to Matrix Analytic Methods in Stochastic Modeling*, ASA-SIAM Series on Statistics and Applied Probability (SIAM), Society for Industrial Mathematics, Philadelphia, PA, 2000.

[8] Neuts, M.F., *Matrix-Geometric Solutions in Stochastic Models*, The John Hopkins University Press, 1981.

[9] Roger, A.H., Charles, R.J., *Matrix analysis*, 4th Edition,The Press Syndicate of the University of Cambrige, 1990.

[10] Sikdar, K., Gupta, U.C., The queue length distributions in the finite buffer bulk-service $MAP/G/1$ queue with multple vacations, *Sociedad de Estadistica e Investigacion Operativa*, Vol. 13, No.1, pp. 75–103, 2005.

[11] Telek, M., Horvath, G., A minimal representation of Markov arrival processes and a moments matching method. *Performance Evaluation*, Vol. 64, pp. 1153–1168, 2007.

[12] Whitt, W., The queueing network analyzer, *The Bell system Technical Journal*, Vol. 62, No. 9, pp. 2779–2814, 1983.

[13] The MathWorks Company, *MATLAB The Language of Technical Computing: Using MALTAB*, Version 6, 2002.

[14] Promodel Corp., *Promodel User Guide*, Promodel Corp., 2001.

Table 1: Arrival processes of two phases simulated in Promodel.

arrival	phase 1 in stream 1	phase 2 in stream 1	phase 1 in stream 2	phase 2 in stream 2
average content	0.62	0.37	0.30	0.70
state	$(k,1,1)$	$(k,1,2)$	$(k,2,1)$	$(k,2,2)$
stationary probability	0.1860	0.4340	0.1110	0.2590

Table 2: Probabilities obtained from three methods with $p = 0.1,\ 0.2,\ 0.3$.

p	0.1			0.2			0.3		
	M	R	LU	M	R	LU	M	R	LU
	0.0976	0.0976	0.0976	0.0949	0.0949	0.0949	0.0916	0.0916	0.0915
$\boldsymbol{\pi}_0$	0.3732	0.3732	0.3732	0.3448	0.3448	0.3447	0.3139	0.3140	0.3137
	0.0910	0.0910	0.0910	0.0816	0.0816	0.0815	0.0717	0.0718	0.0717
	0.3477	0.3477	0.3477	0.2963	0.2963	0.2962	0.2458	0.2458	0.2456
$\bar{\pi}_0$	0.9095	0.9095	0.9095	0.8176	0.8176	0.8173	0.7230	0.7232	0.7226
	0.0153	0.0153	0.0153	0.0308	0.0308	0.0308	0.0461	0.0461	0.0461
$\boldsymbol{\pi}_1$	0.0364	0.0364	0.0364	0.0690	0.0690	0.0690	0.0969	0.0968	0.0969
	0.0111	0.0111	0.0111	0.0206	0.0206	0.0206	0.0281	0.0281	0.0281
	0.0219	0.0219	0.0219	0.0381	0.0381	0.0381	0.0485	0.0484	0.0485
$\bar{\pi}_1$	0.0847	0.0847	0.0847	0.1585	0.1585	0.1585	0.2196	0.2194	0.2196
	0.0013	0.0013	0.0013	0.0055	0.0055	0.0055	0.0127	0.0126	0.0127
$\boldsymbol{\pi}_2$	0.0021	0.0021	0.0021	0.0081	0.0081	0.0081	0.0176	0.0174	0.0175
	0.0008	0.0008	0.0008	0.0031	0.0031	0.0031	0.0065	0.0064	0.0065
	0.0010	0.0010	0.0010	0.0036	0.0036	0.0036	0.0070	0.0069	0.0070
$\bar{\pi}_2$	0.0052	0.0052	0.0052	0.0203	0.0203	0.0203	0.0438	0.0433	0.0437

Table 3: Probabilities obtained from three methods with $p = 0.4,\ 0.5,\ 0.6$.

p	0.4			0.5			0.6		
	M	R	LU	M	R	LU	M	R	LU
	0.0873	0.0875	0.0872	0.0815	0.0820	0.0813	0.0730	0.0738	0.0727
$\boldsymbol{\pi}_0$	0.2798	0.2800	0.2794	0.2410	0.2413	0.2404	0.1956	0.1957	0.1948
	0.0612	0.0615	0.0611	0.0500	0.0506	0.0499	0.0379	0.0391	0.0377
	0.1962	0.1965	0.1959	0.1478	0.1485	0.1475	0.1014	0.1028	0.1010
$\bar{\pi}_0$	0.6245	0.6255	0.6236	0.5203	0.5224	0.5191	0.4079	0.4114	0.4052
	0.0608	0.0608	0.0607	0.0739	0.0738	0.0737	0.0832	0.0825	0.0829
$\boldsymbol{\pi}_1$	0.1187	0.1181	0.1185	0.1321	0.1304	0.1318	0.1336	0.1295	0.1331
	0.0332	0.0333	0.0332	0.0354	0.0355	0.0353	0.0338	0.0341	0.0336
	0.0529	0.0526	0.0528	0.0512	0.0506	0.0511	0.0436	0.0425	0.0434
$\bar{\pi}_1$	0.2656	0.2648	0.2642	0.2926	0.2903	0.2912	0.2942	0.2896	0.2930
	0.0230	0.0227	0.0229	0.0360	0.0352	0.0359	0.0505	0.0480	0.0503
$\boldsymbol{\pi}_2$	0.0295	0.0287	0.0294	0.0423	0.0401	0.0422	0.0533	0.0479	0.0530
	0.0105	0.0104	0.0105	0.0145	0.0142	0.0145	0.0173	0.0167	0.0172
	0.0106	0.0102	0.0106	0.0133	0.0124	0.0133	0.0142	0.0125	0.0142
$\bar{\pi}_2$	0.0736	0.0720	0.0732	0.1061	0.1019	0.1050	0.1353	0.1251	0.1347

Table 4: Comparison of queue empty rate.

p	0.1	0.2	0.3	0.4	0.5	0.6
$\boldsymbol{\pi}_0+\boldsymbol{\pi}_1+\boldsymbol{\pi}_2$ (Matrix geometric method)	0.9996	0.9963	0.9863	0.9637	0.9192	0.8373
Queue empty rate (Promodel 20 hours)	0.9996	0.9964	0.9877	0.9675	0.9139	0.8448

Table 5: Probabilities obtained from three methods with $p = 0.2,\ 0.3,\ 0.4$.

p	0.2			0.3			0.4		
	M	R	LU	M	R	LU	M	R	LU
$\boldsymbol{\pi}_0$	0.0950	0.0950	0.0950	0.0921	0.0921	0.0921	0.0887	0.0888	0.0888
	0.3452	0.3452	0.3452	0.3155	0.3155	0.3156	0.2838	0.2838	0.2839
	0.0817	0.0817	0.0817	0.0721	0.0721	0.0721	0.0622	0.0622	0.0622
	0.2967	0.2967	0.2967	0.2469	0.2470	0.2470	0.1988	0.1989	0.1989
$\bar{\pi}_0$	0.8186	0.8186	0.8186	0.7266	0.7267	0.7268	0.6335	0.6337	0.6335
$\boldsymbol{\pi}_1$	0.0308	0.0308	0.0308	0.0465	0.0465	0.0465	0.0621	0.0621	0.0621
	0.0691	0.0691	0.0691	0.0974	0.0974	0.0974	0.1204	0.1203	0.1204
	0.0207	0.0207	0.0207	0.0283	0.0283	0.0283	0.0339	0.0339	0.0339
	0.0381	0.0381	0.0381	0.0486	0.0486	0.0486	0.0534	0.0533	0.0534
$\bar{\pi}_1$	0.1587	0.1587	0.1587	0.2208	0.2208	0.2208	0.2698	0.2697	0.2698
$\boldsymbol{\pi}_2$	0.0056	0.0056	0.0056	0.0130	0.0130	0.0130	0.0240	0.0240	0.0240
	0.0081	0.0081	0.0081	0.0176	0.0176	0.0176	0.0297	0.0297	0.0298
	0.0031	0.0031	0.0031	0.0066	0.0066	0.0066	0.0109	0.0109	0.0109
	0.0035	0.0035	0.0035	0.0069	0.0068	0.0069	0.0103	0.0102	0.0103
$\bar{\pi}_2$	0.0203	0.0203	0.0203	0.0441	0.0440	0.0441	0.0749	0.0748	0.0750
$\boldsymbol{\pi}_3$	0.0007	0.0007	0.0007	0.0026	0.0026	0.0026	0.0066	0.0065	0.0066
	0.0007	0.0007	0.0007	0.0024	0.0024	0.0024	0.0056	0.0055	0.0056
	0.0003	0.0003	0.0003	0.0011	0.0011	0.0011	0.0026	0.0025	0.0026
	0.0003	0.0003	0.0003	0.0008	0.0008	0.0008	0.0017	0.0017	0.0017
$\bar{\pi}_3$	0.0020	0.0020	0.0020	0.0069	0.0069	0.0069	0.0165	0.0162	0.0165

Table 6: Probabilities obtained from three methods with $p = 0.5,\ 0.6,\ 0.7$.

p	0.5			0.6			0.7		
	M	R	LU	M	R	LU	M	R	LU
$\boldsymbol{\pi}_0$	0.0847	0.0848	0.0848	0.0796	0.0801	0.0798	0.0728	0.0738	0.0730
	0.2493	0.2496	0.2495	0.2112	0.2119	0.2116	0.1679	0.1696	0.1685
	0.0519	0.0520	0.0519	0.0411	0.0415	0.0412	0.0299	0.0319	0.0300
	0.1527	0.1529	0.1528	0.1091	0.1098	0.1093	0.0690	0.0707	0.0693
$\bar{\pi}_0$	0.5386	0.5393	0.5390	0.4410	0.4433	0.4419	0.3396	0.3450	0.3408
$\boldsymbol{\pi}_1$	0.0775	0.0776	0.0776	0.0921	0.0922	0.0923	0.1044	0.1045	0.1048
	0.1366	0.1364	0.1367	0.1440	0.1436	0.1443	0.1392	0.1382	0.1396
	0.0369	0.0370	0.0370	0.0370	0.0372	0.0371	0.0335	0.0342	0.0336
	0.0525	0.0524	0.0525	0.0462	0.0462	0.0463	0.0352	0.0355	0.0353
$\bar{\pi}_1$	0.3035	0.3034	0.3038	0.3193	0.3192	0.3200	0.3123	0.3124	0.3133
$\boldsymbol{\pi}_2$	0.0390	0.0388	0.0390	0.0581	0.0576	0.0582	0.0809	0.0793	0.0811
	0.0434	0.0431	0.0435	0.0567	0.0566	0.0568	0.0663	0.0637	0.0665
	0.0154	0.0154	0.0154	0.0194	0.0193	0.0194	0.0215	0.0216	0.0216
	0.0130	0.0129	0.0130	0.0142	0.0139	0.0142	0.0131	0.0129	0.0131
$\bar{\pi}_2$	0.1108	0.1102	0.1107	0.1484	0.1464	0.1486	0.1818	0.1773	0.1824
$\boldsymbol{\pi}_3$	0.0138	0.0136	0.0138	0.0256	0.0248	0.0257	0.0435	0.0409	0.0436
	0.0106	0.0102	0.0106	0.0171	0.0160	0.0172	0.0243	0.0216	0.0244
	0.0047	0.0046	0.0047	0.0074	0.0074	0.0074	0.0100	0.0095	0.0100
	0.0028	0.0027	0.0028	0.0038	0.0038	0.0038	0.0043	0.0038	0.0043
$\bar{\pi}_3$	0.0319	0.0311	0.0319	0.0539	0.0515	0.0541	0.0821	0.0758	0.0823

Table 7: Probabilities obtained from three methods with $p = 0.8,\ 0.9$.

p	0.8			0.9		
	M	R	LU	M	R	LU
	0.0625	0.0645	0.0625	0.0446	0.0483	0.0428
$\boldsymbol{\pi}_0$	0.1173	0.1210	0.1173	0.0568	0.0653	0.0544
	0.0184	0.0205	0.0184	0.0072	0.0108	0.0069
	0.0344	0.0379	0.0344	0.0090	0.0148	0.0087
$\bar{\pi}_0$	0.2326	0.2439	0.2326	0.1176	0.1382	0.1129
	0.1106	0.1103	0.1105	0.0984	0.0979	0.0943
$\boldsymbol{\pi}_1$	0.1165	0.1152	0.1165	0.0672	0.0653	0.0644
	0.0255	0.0272	0.0255	0.0125	0.0164	0.0120
	0.0208	0.0220	0.0208	0.0065	0.0095	0.0062
$\bar{\pi}_1$	0.2734	0.2747	0.2734	0.1846	0.1901	0.1769
	0.1042	0.0996	0.1042	0.1134	0.1032	0.1087
$\boldsymbol{\pi}_2$	0.0662	0.0613	0.0662	0.0453	0.0410	0.0435
	0.0200	0.0204	0.0200	0.0120	0.0139	0.0115
	0.0093	0.0093	0.0093	0.0035	0.0047	0.0033
$\bar{\pi}_2$	0.1997	0.1906	0.1997	0.1742	0.1628	0.1670
	0.0676	0.0600	0.0676	0.0890	0.0713	0.0853
$\boldsymbol{\pi}_3$	0.0292	0.0236	0.0291	0.0240	0.0173	0.0230
	0.0112	0.0105	0.0112	0.0082	0.0080	0.0079
	0.0038	0.0032	0.0037	0.0017	0.0019	0.0017
$\bar{\pi}_3$	0.1118	0.0973	0.1116	0.1229	0.0984	0.1179

Table 8: Queue empty rate versus a number of probabilities p.

p	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\boldsymbol{\pi}_0+\boldsymbol{\pi}_1+\boldsymbol{\pi}_2+\boldsymbol{\pi}_3$ (Matrix geometric method)	0.9997	0.9985	0.9945	0.9846	0.9625	0.9158	0.8174	0.5993
Queue empty rate (Promodel 20 hours)	0.9997	0.9985	0.9954	0.9854	0.9633	0.9214	0.8255	0.6017

Table 9: Stationary probabilities of a queueing model with twenty servers

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
M	0.0040	0.0202	0.0525	0.0939	0.1303	0.1491	0.1464
R	0.0040	0.0202	0.0525	0.0939	0.1303	0.1491	0.1465
LU	0.0040	0.0202	0.0525	0.0939	0.1302	0.1490	0.1464
$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
M	0.1269	0.0989	0.0703	0.0461	0.0281	0.0161	0.0087
R	0.1269	0.0989	0.0703	0.0461	0.0281	0.0161	0.0087
LU	0.1269	0.0988	0.0703	0.0461	0.0281	0.0161	0.0087
$\bar{\pi}_i$	$\bar{\pi}_{14}$	$\bar{\pi}_{15}$	$\bar{\pi}_{16}$	$\bar{\pi}_{17}$	$\bar{\pi}_{18}$	$\bar{\pi}_{19}$	$\bar{\pi}_{20}$
M	0.0045	0.0022	0.0010	0.0005	0.0002	0.0000	0.0000
R	0.0045	0.0022	0.0010	0.0005	0.0002	0.0000	0.0000
LU	0.0045	0.0022	0.0010	0.0005	0.0002	0.0000	0.0000

Table 10: The queue empty rate of twenty servers versus probabilities p.

p	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
Matrix geometric method	1.0000	0.9998	0.9990	0.9953	0.9822	0.9430	0.8421	0.6122
Ramawami	1.0000	0.9998	0.9990	0.9954	0.9838	0.9526	0.8794	0.7098
LU factorization	1.0000	0.9998	0.9990	0.9952	0.9818	0.9417	0.8381	0.6023

Table 11: Comparison of queue empty rate of twenty servers.

p	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
$\boldsymbol{\pi}_0+\boldsymbol{\pi}_1+\cdots+\boldsymbol{\pi}_{20}$ (Matrix geometric method)	1.0000	0.9998	0.9990	0.9953	0.9822	0.9430	0.8421	0.6122
Queue empty rate (Promodel 20 hours)	1.0000	1.0000	0.9998	0.9974	0.9899	0.9636	0.8569	0.6025

Table 12: Stationary probabilities of a queueing model with twenty-five servers

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
M	0.0015	0.0084	0.0251	0.0521	0.0841	0.1124	0.1294
R	0.0015	0.0084	0.0251	0.0521	0.0841	0.1124	0.1294
LU	0.0015	0.0084	0.0251	0.0521	0.0841	0.1124	0.1294

$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
M	0.1317	0.1208	0.1014	0.0786	0.0569	0.0387	0.0249
R	0.1317	0.1208	0.1014	0.0786	0.0569	0.0387	0.0249
LU	0.1317	0.1208	0.1014	0.0786	0.0569	0.0387	0.0249

$\bar{\pi}_i$	$\bar{\pi}_{14}$	$\bar{\pi}_{15}$	$\bar{\pi}_{16}$	$\bar{\pi}_{17}$	$\bar{\pi}_{18}$	$\bar{\pi}_{19}$	$\bar{\pi}_{20}$
M	0.0152	0.0088	0.0049	0.0026	0.0013	0.0007	0.0003
R	0.0152	0.0088	0.0049	0.0026	0.0013	0.0007	0.0003
LU	0.0152	0.0088	0.0049	0.0026	0.0013	0.0007	0.0003

$\bar{\pi}_i$	$\bar{\pi}_{21}$	$\bar{\pi}_{22}$	$\bar{\pi}_{23}$	$\bar{\pi}_{24}$	$\bar{\pi}_{25}$
M	0.0001	0.0000	0.0000	0.0000	0.0000
R	0.0001	0.0000	0.0000	0.0000	0.0000
LU	0.0001	0.0000	0.0000	0.0000	0.0000

Table 13: The queue empty rate of twenty-five servers versus probabilities p.

p	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85
Matrix geometric method	1.0000	1.0000	0.9998	0.9988	0.9942	0.9769	0.9201	0.7562
Ramawami	1.0000	1.0000	0.9998	0.9988	0.9946	0.9803	0.9392	0.8302
LU factorization	1.0000	1.0000	0.9998	0.9988	0.9945	0.9775	0.9204	0.7498

Table 14: Comparison of queue empty rates of twenty-five servers.

p	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85
$\boldsymbol{\pi}_0+\boldsymbol{\pi}_1+\cdots+\boldsymbol{\pi}_{25}$ (Matrix geometric method)	1.0000	1.0000	0.9998	0.9988	0.9942	0.9769	0.9201	0.7562
Queue empty rate (Promodel 20 hours)	1.0000	1.0000	0.9999	0.9998	0.9945	0.9772	0.9241	0.7547

Table 15: Stationary probabilities of a queueing model with thirty servers

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
M	0.0005	0.0032	0.0110	0.0258	0.0474	0.0723	0.0953
R	0.0005	0.0032	0.0110	0.0258	0.0474	0.0723	0.0953
LU	0.0005	0.0032	0.0110	0.0258	0.0474	0.0723	0.0953

$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
M	0.1113	0.1175	0.1136	0.1018	0.0853	0.0672	0.0501
R	0.1113	0.1175	0.1136	0.1018	0.0853	0.0672	0.0501
LU	0.1113	0.1175	0.1136	0.1018	0.0853	0.0672	0.0501

$\bar{\pi}_i$	$\bar{\pi}_{14}$	$\bar{\pi}_{15}$	$\bar{\pi}_{16}$	$\bar{\pi}_{17}$	$\bar{\pi}_{18}$	$\bar{\pi}_{19}$	$\bar{\pi}_{20}$
M	0.0355	0.0240	0.0156	0.0097	0.0058	0.0033	0.0019
R	0.0355	0.0240	0.0156	0.0097	0.0058	0.0033	0.0019
LU	0.0355	0.0240	0.0156	0.0097	0.0058	0.0033	0.0019

$\bar{\pi}_i$	$\bar{\pi}_{21}$	$\bar{\pi}_{22}$	$\bar{\pi}_{23}$	$\bar{\pi}_{24}$	$\bar{\pi}_{25}$	$\bar{\pi}_{26}$	$\bar{\pi}_{27}$
M	0.0010	0.0005	0.0003	0.0001	0.0000	0.0000	0.0000
R	0.0010	0.0005	0.0003	0.0001	0.0000	0.0000	0.0000
LU	0.0010	0.0005	0.0003	0.0001	0.0000	0.0000	0.0000

$\bar{\pi}_i$	$\bar{\pi}_{28}$	$\bar{\pi}_{29}$	$\bar{\pi}_{30}$
M	0.0000	0.0000	0.0000
R	0.0000	0.0000	0.0000
LU	0.0000	0.0000	0.0000

Table 16: The queue empty rate of thirty servers versus probabilities p.

p	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
Matrix geometric method	1.0000	1.0000	0.9999	0.9995	0.9973	0.9860	0.9377	0.7493
Ramawami	1.0000	1.0000	0.9999	0.9995	0.9974	0.9882	0.9546	0.8412
LU factorization	1.0000	1.0000	1.0000	0.9996	0.9973	0.9856	0.9316	0.7089

Table 17: Comparison of queue empty rates of thirty servers.

p	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95
$\boldsymbol{\pi}_0+\boldsymbol{\pi}_1+\cdots+\boldsymbol{\pi}_{30}$ (Matrix geometric method)	1.0000	1.0000	0.9999	0.9995	0.9973	0.9860	0.9377	0.7493
Queue empty rate (Promodel 20 hours)	1.0000	1.0000	0.9998	0.9998	0.9979	0.9846	0.9526	0.7707

Figure 2: The queue empty rate determined by four different methods

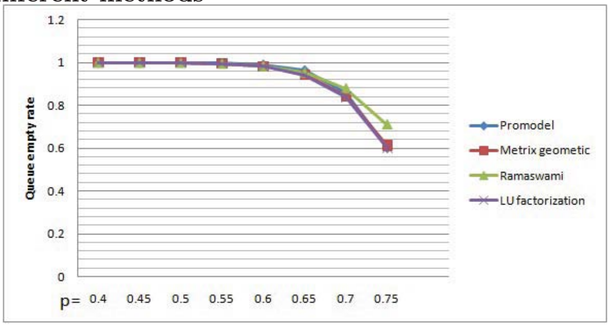

Figure 3: Relative errors of three methods compared with Promodel

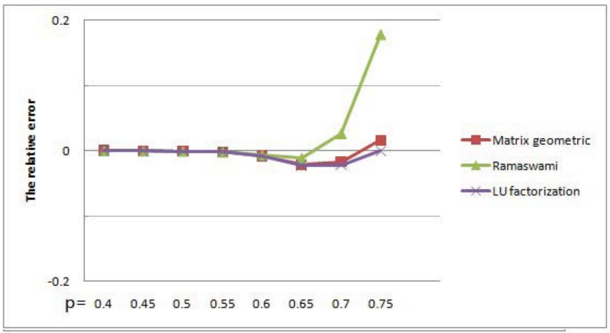

Figure 4: The queue empty rate determined by four different methods

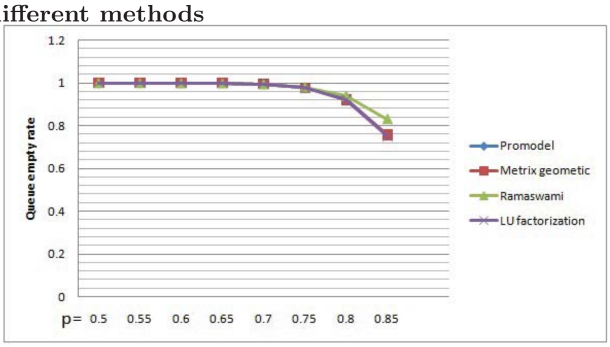

Figure 5: Relative errors of three methods compared with Promodel

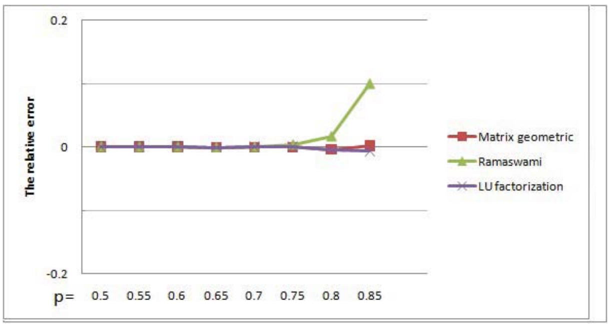

Figure 6: The queue empty rate determined by four different methods

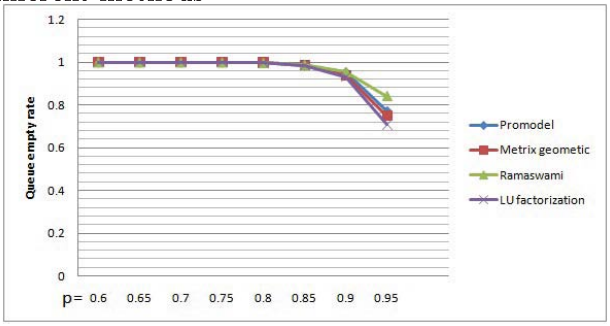

Figure 7: Relative errors of three methods compared with Promodel