國 立 政 治 大 學 應 用 數 學 系

數 學 教 學 碩 士 在 職 專 班

碩士學位論文

A Phase-type Queueing Model with Multiple Servers by Matrix Decomposition Approaches

以矩陣分解法計算特別階段形機率分配並有多人服務之排隊模型

碩專班學生：顏源亨 撰

指導教授： 陸 行 博士

中華民國 一百 年 七 月 十二 日

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 thesis, 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. After determining the stationary probability, we study the departure process and the moments of inter-departure times. We compute the moments of inter-departure times and the variance by applying two numerical methods (Matlab and Promodel). The lag_k correlation of inter-departure times is also introduced in the thesis. The proposed approach is proved theoretically and verified with illustrative examples.

中文摘要

穩定狀態機率是讓我們了解各種排隊網絡性能的基礎。在擬生死過程（Quasi-Birth-and-Death）Phase-type 分配中求得穩定狀態機率，通常是依賴排隊網絡的結構。在這篇論文中，我們提出了一種計算方法−LU 分解，可以求得在排隊網絡中有多台服務器的穩定狀態機率。此計算方法提供了一種通用的方法，使得複雜的大矩陣變成小矩陣，並減低計算的複雜性。當需要計算一個複雜的大矩陣，這個成果變得更加重要。文末，我們提到了離開時間間隔，並用兩種方法 (Matlab 和 Promodel) 去計算期望值和變異數，我們發現兩種方法算出的數據相近，接著計算離開顧客的時間間隔相關係數。最後，我們提供數值實驗以計算不同服務器個數產生的離去過程和相關係數，用來說明我們的方法。

誌謝

在學校教書多年，來到政大應數系接觸等候理論，承蒙恩師 陸行教授認真的指導，在研究所三年的學習階段，多次的討論過程中獲益良多，在此致上最衷心的謝意和敬意。口試期間，承蒙 張國華教授、曾正男教授不吝惠賜諸多寶貴意見，使得本論文更趨完整，特此銘謝。

研究期間，首先感謝王嘉宏學長對論文結構、寫法的指導，更要感謝林雋民同學在數學運算程式的寫法、運算過程的指導，在經過不斷的嘗試與挫折中，最後得到正確的數據，猶如荒漠之中遇見桃花源。也感謝學弟們，在每次討論之中，讓我得到許多啟發。

這段期間所面臨的壓力，感謝我的同事，不時的關心與鼓勵。以及我親愛的老婆珮璇，每天細心呵護我的健康，給我滿滿的愛。最後衷心感謝我的父母對我的栽培，因為有你們背後的支持，得以順利完成碩士學位，願將此項殊榮獻給我最摯愛的家人共享。

Contents

List of Figures

List of Tables

Chapter 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 inter-arrival 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 thesis, 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 solution procedure in [8], which will be combined with Ramaswami's formula [7] and block LU factorization [6] in this thesis. The main contribution in the thesis 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 sub-matrices by using Matrix-Geometric Solution 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 Solution 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 Solution Method, the infinitesimal generator matrix is decomposed into sub-matrices, 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 inter-departure times of customers leaving the system are correlated [4]. In the standard network node approximation approach, the departure process from a workstation system is normally approximated by assuming that these inter-departure times are independent and identically distributed (i.i.d.). However, this i.i.d. assumption allows for a simple approximation [12] of the squared coefficient of variation (SCV) of departure process (C_d^2) as a function of the systems utilization (u) and the arrival and service processes SCV's (C_a^2, C_s^2) for a $G/G/n$ queue as

$$C_d^2 = (1-u^2)(C_a^2-1) + u^2(C_s^2-1)/\sqrt{n}.$$

Bitran and Dasu [1] developed a phase-type distribution representation of the departure process from a single server system and provided moments of the inter-departure times of a $\sum Ph_i/Ph/1$ queue. We extend those results to get the moment of inter-departure times and correlations between successive departures.

The remainder of the thesis is organized as follows. Chapter 2 introduces a queueing model with phase-type Markovian arrival process. In Chapter 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 Solution Method combined with Ramaswami's formula and LU factorization. We introduce the inter-departure times in Chapter 4. Numerical results of $MAP/M/n$ queueing systems with multiple servers are given in Chapter 5, and numerical results of the stationary distribution are compared with approximation methods and simulations. Concluding remarks are to be given in Chapter 6.

Chapter 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. 2.1, 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.,

$$\mathbf{S}_1 = \begin{bmatrix} -\mu \end{bmatrix}, \ \mathbf{S}_{1o} = \begin{bmatrix} \mu \end{bmatrix}.$$

The queueing network has two independent and identical arrival streams, where there are two phases for each arrival stream [4]. 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.

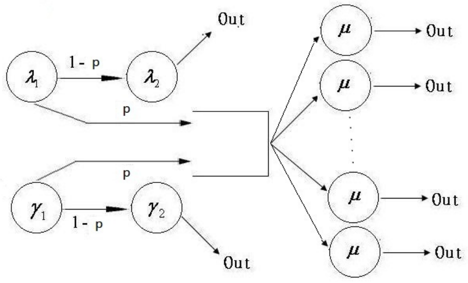


Figure 2.1: A semi $MAP/M/n$ queueing model

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

$$\mathbf{T}_1 = \begin{bmatrix} -\lambda_1 & (1-p)\lambda_1 \\ \lambda_2 & -\lambda_2 \end{bmatrix}, \ \mathbf{T}_{1o} = \begin{bmatrix} p\lambda_1 \\ 0 \end{bmatrix},$$

$$\mathbf{T}_2 = \begin{bmatrix} -\gamma_1 & (1-p)\gamma_1 \\ \gamma_2 & -\gamma_2 \end{bmatrix}, \ \mathbf{T}_{2o} = \begin{bmatrix} p\gamma_1 \\ 0 \end{bmatrix}.$$

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 a 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 Solution 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

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{2.1}$$

where there exists 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 Solution 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 $\rho = \overline{\lambda}/(n\mu) < 1$.

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

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

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

$$b \;=\; -(\lambda_1\lambda_2\gamma_2 + \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + \gamma_1\gamma_2\lambda_2 + n\mu\lambda_2\gamma_1 + 2n\mu\lambda_1\gamma_1 + n\mu\lambda_1\gamma_2),$$

and

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

Proof :

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

$$\overline{\lambda} = p[(\frac{1}{\lambda_1}p + (\frac{1}{\lambda_1} + \frac{1}{\lambda_2})(1-p)]^{-1} + p[(\frac{1}{\gamma_1}p + (\frac{1}{\gamma_1} + \frac{1}{\gamma_2})(1-p)]^{-1} < n\mu. \tag{2.2}$$

It implies that

$$\frac{\lambda_1 p}{\lambda_1 + \lambda_2 - \lambda_1 p} + \frac{\gamma_1 p}{\gamma_1 + \gamma_2 - \gamma_1 p} < n\mu. \tag{2.3}$$

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

$$b = -(\lambda_1\lambda_2\gamma_2 + \gamma_1\gamma_2\lambda_1 + \lambda_1\lambda_2\gamma_1 + \gamma_1\gamma_2\lambda_2 + n\mu\lambda_2\gamma_1 + 2n\mu\lambda_1\gamma_1 + n\mu\lambda_1\gamma_2),$$

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

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

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 needs not for $(0, 1, \cdots, n-1)$. We define the vector of probabilities that there are k customers in the system as

$$\begin{aligned} \boldsymbol{\pi}_k &= \lim_{t\to\infty} Pr\{A(t) = k, J(t) = (i,j)\} \\ &= (\pi_{k,1,1}\ \pi_{k,1,2}\ \pi_{k,2,1}\ \pi_{k,2,2}), \end{aligned} \tag{2.4}$$

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

$$\mathbf{L} \otimes \mathbf{M} = [\mathbf{l}_{ij}\mathbf{M}] \ \ for\ all\ i,\ j,$$

where $\mathbf{l}_{ij}$ is the ith row and jth column element of the matrix $\mathbf{L}$.

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, we obtain

$$\mathbf{B}_{00} = \mathbf{T}_1 \oplus \mathbf{T}_2 = \begin{bmatrix} -\lambda_1 - \gamma_1 & (1-p)\gamma_1 & (1-p)\lambda_1 & 0 \\ \gamma_2 & -\lambda_1 - \gamma_2 & 0 & (1-p)\lambda_1 \\ \lambda_2 & 0 & -\lambda_2 - \gamma_1 & (1-p)\gamma_1 \\ 0 & \lambda_2 & \gamma_2 & -\lambda_2 - \gamma_2 \end{bmatrix},$$

and

$$\mathbf{C} = (\mathbf{T}_{1o} \otimes \mathbf{e}_1^T) \oplus (\mathbf{T}_{2o} \otimes \mathbf{e}_1^T) = \begin{bmatrix} p(\lambda_1 + \gamma_1) & 0 & 0 & 0 \\ 0 & p\lambda_1 & 0 & 0 \\ 0 & 0 & p\gamma_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Using the arrival and service process parameters in terms of the Kronecker product and sum, we obtain sub-matrices $\mathbf{A}_{10}$, $\mathbf{A}_{21}$, $\mathbf{A}_{(i)(i-1)}$, $\mathbf{A}$, which represent a customer is in service, finishes the service, and departs the system, respectively.

$$\mathbf{A}_{10} = \mathbf{I}_T \otimes \mathbf{S}_{1o} = \begin{bmatrix} \mu & 0 & 0 & 0 \\ 0 & \mu & 0 & 0 \\ 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & \mu \end{bmatrix},$$

$$\mathbf{A}_{21} = \mathbf{I}_T \otimes (\mathbf{S}_{1o} \oplus \mathbf{S}_{1o}) = \begin{bmatrix} 2\mu & 0 & 0 & 0 \\ 0 & 2\mu & 0 & 0 \\ 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 2\mu \end{bmatrix},$$

$$\mathbf{A}_{(i)(i-1)} = \mathbf{I}_T \otimes (\underbrace{\mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}}_{i}) = \begin{bmatrix} i\mu & 0 & 0 & 0 \\ 0 & i\mu & 0 & 0 \\ 0 & 0 & i\mu & 0 \\ 0 & 0 & 0 & i)\mu \end{bmatrix},$$

for $3 \leqslant i \leqslant n-1$ and

$$\mathbf{A} = \mathbf{I}_T \otimes (\underbrace{\mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}}_{n}) = \begin{bmatrix} n\mu & 0 & 0 & 0 \\ 0 & n\mu & 0 & 0 \\ 0 & 0 & n\mu & 0 \\ 0 & 0 & 0 & n\mu \end{bmatrix},$$

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

Next, we define sub-matrices $\mathbf{B}_{00}$, $\mathbf{B}_{11}$, $\mathbf{B}_{ii}$, and $\mathbf{B}$ as follows, where the internal phase changes for the composite arrival process, which are

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

$$= \begin{bmatrix} -\lambda_1-\gamma_1 & (1-p)\gamma_1 & (1-p)\lambda_1 & 0 \\ \gamma_2 & -\lambda_1-\gamma_2 & 0 & (1-p)\lambda_1 \\ \lambda_2 & 0 & -\lambda_2-\gamma_1 & (1-p)\gamma_1 \\ 0 & \lambda_2 & \gamma_2 & -\lambda_2-\gamma_2 \end{bmatrix},$$

$$\mathbf{B}_{11} = \mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \mathbf{S}_1$$

$$= \begin{bmatrix} -\lambda_1-\gamma_1-\mu & (1-p)\gamma_1 & (1-p)\lambda_1 & 0 \\ \gamma_2 & -\lambda_1-\gamma_2-\mu & 0 & (1-p)\lambda_1 \\ \lambda_2 & 0 & -\lambda_2-\gamma_1-\mu & (1-p)\gamma_1 \\ 0 & \lambda_2 & \gamma_2 & -\lambda_2-\gamma_2-\mu \end{bmatrix},$$

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

$$= \begin{bmatrix} -\lambda_1-\gamma_1-i\mu & (1-p)\gamma_1 & (1-p)\lambda_1 & 0 \\ \gamma_2 & -\lambda_1-\gamma_2-i\mu & 0 & (1-p)\lambda_1 \\ \lambda_2 & 0 & -\lambda_2-\gamma_1-i\mu & (1-p)\gamma_1 \\ 0 & \lambda_2 & \gamma_2 & -\lambda_2-\gamma_2-i\mu \end{bmatrix},$$

for $2 \leqslant i \leqslant n-1$,

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

$$= \begin{bmatrix} -\lambda_1-\gamma_1-n\mu & (1-p)\gamma_1 & (1-p)\lambda_1 & 0 \\ \gamma_2 & -\lambda_1-\gamma_2-n\mu & 0 & (1-p)\lambda_1 \\ \lambda_2 & 0 & -\lambda_2-\gamma_1-n\mu & (1-p)\gamma_1 \\ 0 & \lambda_2 & \gamma_2 & -\lambda_2-\gamma_2-n\mu \end{bmatrix},$$

and

$$\mathbf{C} = (\mathbf{T}_{1o} \otimes \mathbf{e}_1^T) \oplus (\mathbf{T}_{2o} \otimes \mathbf{e}_1^T) = \begin{bmatrix} p(\lambda_1 + \gamma_1) & 0 & 0 & 0 \\ 0 & p\lambda_1 & 0 & 0 \\ 0 & 0 & p\gamma_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

representing that a customer 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,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \ddots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{A} & \mathbf{B} & \mathbf{C} & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{A} & \mathbf{B} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \tag{2.5}$$

where n is the number of servers in the system. The matrix $\mathbf{Q}$ is composed of sub-matrices along with the block tridiagonal matrix.

Chapter 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 (3.1)-(3.5):

$$\boldsymbol{\pi}_0\mathbf{B}_{00} + \boldsymbol{\pi}_1\mathbf{A}_{10} = \mathbf{0}, \tag{3.1}$$

$$\boldsymbol{\pi}_0\mathbf{C} + \boldsymbol{\pi}_1\mathbf{B}_{11} + \boldsymbol{\pi}_2\mathbf{A}_{21} = \mathbf{0}, \tag{3.2}$$

$$\boldsymbol{\pi}_1 C + \boldsymbol{\pi}_2\mathbf{B}_{22} + \boldsymbol{\pi}_3\mathbf{A}_{32} = \mathbf{0}, \tag{3.3}$$

$$\vdots$$

$$\boldsymbol{\pi}_{n-2}\mathbf{C} + \boldsymbol{\pi}_{n-1}\mathbf{B}_{(n-1)(n-1)} + \boldsymbol{\pi}_n\mathbf{A}_{(n)(n-1)} = \mathbf{0}. \tag{3.4}$$

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

$$\boldsymbol{\pi}_{i-1}\mathbf{C} + \boldsymbol{\pi}_i\mathbf{B} + \boldsymbol{\pi}_{i+1}\mathbf{A} = \mathbf{0}, \quad i = n, n+1, n+2, \cdots . \tag{3.5}$$

Using (3.5), 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 (3.1)-(3.5) in matrix form as follows

$$\begin{bmatrix} \boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{n-1} & \boldsymbol{\pi}_n \end{bmatrix} \cdot \mathbf{Q}_1 = \mathbf{0}, \tag{3.6}$$

where

$$\mathbf{Q}_1 = \begin{bmatrix} \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \mathbf{C} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{(n-1)(n-2)} & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{A} & \mathbf{B} + \mathbf{RA} \end{bmatrix}.$$

In addition, by using the normalization condition, we obtain

$$\boldsymbol{\pi}_0 \cdot \mathbf{1} + \boldsymbol{\pi}_1 \cdot \mathbf{1} + \cdots + \boldsymbol{\pi}_n(\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1} = 1. \tag{3.7}$$

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

$$\begin{bmatrix} \boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{n-1} & \boldsymbol{\pi}_n \end{bmatrix} \cdot \mathbf{Q}_2 = [1, \mathbf{0}], \tag{3.8}$$

where

$$\mathbf{Q}_2 = \begin{bmatrix} \mathbf{1} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{A}_{10} & \mathbf{B}_{11} & \mathbf{C} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{A}_{21} & \mathbf{B}_{22} & \ddots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{(n-1)(n-2)} & \mathbf{B}_{(n-1)(n-1)} & \mathbf{C} \\ (\mathbf{I} - \mathbf{R})^{-1} \cdot \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{A} & \mathbf{B} + \mathbf{RA} \end{bmatrix}.$$

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

Ramaswami's formula [7]

Consider computing $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$. That is,

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

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

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

$$\mathbf{B}_{-1} = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A} \end{bmatrix}_{4\times 4(n+1)},$$

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{C} \end{bmatrix}_{4(n+1)\times 4}.$$

It also gives

$$\begin{bmatrix} \mathbf{B} & \mathbf{C} & \mathbf{0} & \cdots \\ \mathbf{A} & \mathbf{B} & \mathbf{C} & \cdots \\ \mathbf{0} & \mathbf{A} & \mathbf{B} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix} = \mathbf{VW},$$

where

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_0 & \mathbf{V}_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{V}_0 & \mathbf{V}_1 & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_0 & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \mathbf{W} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ -\mathbf{H} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & -\mathbf{H} & \mathbf{I} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Then we have

$$\begin{bmatrix} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} & \boldsymbol{\pi}_{n+2} & \cdots \end{bmatrix} \left[\begin{array}{c|ccc} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{0} & \cdots \\ \hline \mathbf{B}_{-1} & & & \\ \mathbf{0} & & \mathbf{VW} & \\ \vdots & & & \end{array}\right] = \mathbf{0},$$

which is equivalent to

$$\begin{bmatrix} \boldsymbol{\pi}^* & \boldsymbol{\pi}_{n+1} & \boldsymbol{\pi}_{n+2} & \cdots \end{bmatrix} \left[\begin{array}{c|ccc} \mathbf{B}_0 & \mathbf{B}_1^* & \mathbf{0} & \cdots \\ \hline \mathbf{B}_{-1} & & & \\ \mathbf{0} & & \mathbf{V} & \\ \vdots & & & \end{array}\right] = \mathbf{0}.$$

As we know

$$\mathbf{W}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{H} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{H} & \mathbf{I} & \cdots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix}, \text{and} \begin{bmatrix} \mathbf{B}_1^* & \mathbf{0} & \mathbf{0} & \cdots \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{0} & \cdots \end{bmatrix} \mathbf{W}^{-1}.$$

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:

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

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

and

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

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

$$\boldsymbol{\pi}^*(\mathbf{B}_0 - \mathbf{B}_1\mathbf{V}_0^{-1}\mathbf{B}_{-1}) = 0 \tag{3.9}$$

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

$$\begin{bmatrix} \mathbf{B}_{00}^{T\,*} & \mathbf{A}_{10}{}^* & \boldsymbol{\Omega} & \boldsymbol{\Omega} & \cdots & \boldsymbol{\Omega} & \overline{\boldsymbol{\Omega}} \\ \mathbf{C} & \mathbf{B}_{11}{}^T & \mathbf{A}_{21} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{B}_{22}^T & \mathbf{A}_{32} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C} & \mathbf{B}_{(n-1)(n-1)}^T & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{C} & (\mathbf{B}+\mathbf{R}\mathbf{A})^T \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\pi}_0 \\ \boldsymbol{\pi}_1 \\ \boldsymbol{\pi}_2 \\ \vdots \\ \boldsymbol{\pi}_{n-1} \\ \boldsymbol{\pi}_n \end{bmatrix} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

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

$$\mathbf{Z}_n = \begin{bmatrix} \mathbf{B}_{00}^{T\,**} & \mathbf{A}_{10}{}^{**} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{C} & \mathbf{B}_{11}^T & \mathbf{A}_{21} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \mathbf{B}_{22}^T & \mathbf{A}_{32} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C} & \mathbf{B}_{(n-1)(n-1)}^T & \mathbf{A} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{C} & (\mathbf{B}+\mathbf{RA})^T \end{bmatrix},$$

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 nonsigular 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 nonsigular 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 $\mathbf{Z}_n = \mathbf{LU}$ [9].

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:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{L}_1 & \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_2 & \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{L}_{n-1} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{L}_n & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}_0 & \mathbf{F}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 & \mathbf{F}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_2 & \mathbf{F}_3 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{U}_{n-1} & \mathbf{F}_n \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{U}_n \end{bmatrix}.$$

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

Algorithm 1: **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 2: **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 singular 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

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

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

Algorithm 1: **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 composing a 4×4 matrix.

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

Chapter 4

Inter-Departure times

4.1 Departure process

Characterizing the departure process involves developing an infinitesimal generator for the inter-departure times. The elements needed in this development are the departure-point stationary probabilities $\mathbf{d} = (\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \cdots)$. They are related to the continuous time stationary probabilities $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots)$ by the following relationships. Here, we denote the total arrival average rate by $\overline{\lambda}$ due to superposition of the two arrival streams, and

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

$$\mathbf{d}_0 = \boldsymbol{\pi}_1 \mathbf{A}_{10}/\overline{\lambda}, \tag{4.1}$$

$$\mathbf{d}_1 = \boldsymbol{\pi}_2 \mathbf{A}_{21}/\overline{\lambda}, \tag{4.2}$$

$$\mathbf{d}_2 = \boldsymbol{\pi}_3 \mathbf{A}_{32}/\overline{\lambda}, \tag{4.3}$$

$$\mathbf{d}_{i-1} = \boldsymbol{\pi}_i \mathbf{A}_{i(i-1)}/\overline{\lambda},\ for\ i = 3, 4, \cdots, n. \tag{4.4}$$

The departure process has three different partitions:

First, when there is no customer in the system, a departure has to wait until at least one customer has occurred followed by a service.

Second, when there is at least 1 and at most $(n-1)$ customers in the system, the inter-departure time can be a function of single processing customer's remaining service or it could evolve from a customer arrived and its completion of service.

Third, when a departing customer leaves the system with at least n remaining customers, the minimum of the remaining service time of one of these customers or a complete service time of another customer becomes the inter-departure time.

We find that when there remain at least n customers in the system, the inter-departure time characteristics are the same for all these cases. The infinitesimal generator matrix for the departure process $\mathbf{G}_{n1}$ is given by

$$\mathbf{G}_{n1} = \begin{array}{c|cccccc} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \cdots & \mathbf{n-1} & n^+ \\ \hline \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{B}_{11} & \mathbf{C} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{22} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{n-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{B}_{(n-1)(n-1)} & \widehat{\mathbf{C}} \\ n^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{SS}_{in} \end{array},$$

where only two of these sub-matrices are given by

$$\mathbf{SS}_{in} = \mathbf{S}_1 \oplus \cdots \oplus \mathbf{S}_1,$$

$$\widehat{\mathbf{C}} = \mathbf{C} \cdot \mathbf{1}.$$

The probabilities of the departure-process system starting in the various states $\mathbf{d} = (\mathbf{d}_0, \mathbf{d}_1, \cdots, d_{n^+})$ are made up of the departure point probabilities, with

$$d_{n^+} = (\sum_{a=n+1}^{\infty} \boldsymbol{\pi}_a \mathbf{A}/\overline{\lambda})\mathbf{1}.$$

This series can be written in closed form as

$$\begin{aligned} d_{n^+} &= (\sum_{a=n+1}^{\infty} \boldsymbol{\pi}_n \mathbf{R}^{a-n} \mathbf{A}/\overline{\lambda})\mathbf{1} \\ &= \boldsymbol{\pi}_n[(\mathbf{I}-\mathbf{R})^{-1} - \mathbf{I}]\mathbf{A}/\overline{\lambda})\mathbf{1}. \end{aligned}$$

4.2 Moments of inter-departure times

The stationary inter-departure time is of the phase type distribution characterized by $[\mathbf{d}, \mathbf{G}_{n1}]$. Thus, from Neuts [8], the moments of inter-departure times random variable X are given below

$$E[X^k] = k!(-1)^k \mathbf{d}(\mathbf{G}_{n1})^{-k}\mathbf{1}, k = 1, 2, \ldots,$$

$$Var[X] = E[X^2] - (E[X])^2.$$

4.3 Lag_k correlations between successive departures

The stationary probabilities of the states of the arrival process, $\boldsymbol{\theta}$, are obtained from its Markovian arrival process representation. We define

$$\widehat{\mathbf{A}}_{n(n-1)} = \boldsymbol{\theta} \otimes (\mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}).$$

To get the lag_k correlations of the output inter-departure times, it is necessary to develop the generator matrix $\widehat{\mathbf{G}_n}$ segmented into two matrices: the internal transitions(without departures) $\mathbf{G}_{n1}$ and the matrix containing the departure transition $\mathbf{G}_{n2}$ such that $\widehat{\mathbf{G}_n} = \mathbf{G}_{n1} + \mathbf{G}_{n2}$.

The matrix $\mathbf{G}_{n2}$ is characterized as

$$\mathbf{G}_{n2} = \begin{array}{c|cccccc} & \mathbf{0} & \mathbf{1} & \mathbf{2} & \cdots & \mathbf{n-1} & n^+ \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{A}_{10} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{A}_{21} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{n-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} \\ n^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ddots & (1-t)\widehat{\mathbf{A}}_{n(n-1)} & t\mathbf{S}\mathbf{S}_{out} \end{array},$$

where t is the probability that departure leaves the system with at least n customers remaining. Define t and $\mathbf{SS}_{out}$ as

$$t = \frac{\sum_{a=n+1}^{\infty} \boldsymbol{\pi}_a \mathbf{1}}{\boldsymbol{\pi}_n \mathbf{1} + \sum_{a=n+1}^{\infty} \boldsymbol{\pi}_a \mathbf{1}} = \frac{d_{n^+}}{\mathbf{d}_{n-1}\mathbf{1} + d_{n^+}},$$

$$\mathbf{SS}_{out} = \mathbf{S}_{1o} \oplus \cdots \oplus \mathbf{S}_{1o}.$$

Given the $\mathbf{G}_{n1}$ and $\mathbf{G}_{n2}$ matrices, from Bodrog, Horvath, Telek [2], and Telek, Horvath [11], the lag_k correlation is computed by

$$lag_k = \frac{(\bar{\lambda})^2 \mathbf{d}(-\mathbf{G}_{n1})^{-1}((-\mathbf{G}_{n1})^{-1}\mathbf{G}_{n2})^k(-\mathbf{G}_{n1})^{-1}\mathbf{1} - 1}{2(\bar{\lambda})^2 \mathbf{d}(-\mathbf{G}_{n1})^{-2}\mathbf{1} - 1)}.$$

Chapter 5

Numerical Examples

5.1 Queueing models with two servers

In this Chapter we present four sets of numerical examples to demonstrate the matrix decomposition approach for stationary probabilities of phase-type queueing models with multiple servers.

$M/M/2$ queueing models

Consider a classic model of multiple servers for a further comparison and validate our model. Without loss of the Poisson assumption, we consider an arrival stream combined by two independent Poisson processes. Here, we present a numerical example of $M/M/2$ queueing model. 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 MA, RA, and LU methods, we take $\bar{\pi}_0\ \bar{\pi}_1\ \cdots\ \bar{\pi}_{12}\ \bar{\pi}_{13}$ to compare the values. We find that they are the same.

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
MA	0.1429	0.2143	0.1607	0.1205	0.0904	0.0678	0.0508
RA	0.1429	0.2143	0.1607	0.1205	0.0904	0.0678	0.0508
LU	0.1429	0.2143	0.1607	0.1205	0.0904	0.0678	0.0508

$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
MA	0.0381	0.0286	0.0214	0.0160	0.0121	0.0090	0.0068
RA	0.0381	0.0286	0.0214	0.0160	0.0121	0.0090	0.0068
LU	0.0381	0.0286	0.0214	0.0160	0.0121	0.0090	0.0068

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

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 $1 - \bar{\pi}_0 - \frac{1}{2}\bar{\pi}_1$. 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.

The stationary probabilities of the states of the arrival process

Then, 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 Markovian 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 5.1.

We have the values $\boldsymbol{\theta}$ = (0.1860, 0.4340, 0.1110, 0.2590). We can find that the value is close to

$$\boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \boldsymbol{\pi}_2 + \boldsymbol{\pi}_3 + \ldots = \boldsymbol{\pi}_0 + \boldsymbol{\pi}_1 + \boldsymbol{\pi}_2(\mathbf{I} - \mathbf{R})^{-1} = (0.1838, 0.4412, 0.1103, 0.2647).$$

Table 5.1: 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

The matrix geometric solution procedure

By the matrix geometric solution 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.

Ramaswami's formula

With RA method [7], we have

$$\boldsymbol{\pi}_0 = (0.0874, 0.2798, 0.0613, 0.1962),$$

$$\boldsymbol{\pi}_1 = (0.0608, 0.1186, 0.0333, 0.0528),$$

and

$$\boldsymbol{\pi}_2 = (0.0229, 0.0293, 0.0105, 0.0104),$$

which will be compared with simulation, LU approach in the next section.

LU factorization

Next, by applying the algorithm of LU factorization, 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

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

Numerical experiments by changing p

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, **MA** represents the matrix geometric solution procedure, **RA** 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 5.2 and Table 5.3 shows the comparison of numerical results.

Promodel

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 5.4. Here, the

Table 5.2: Probabilities obtained from three methods with $p = 0.1,\ 0.2,\ 0.3$.

p	0.1			0.2			0.3		
	MA	**RA**	**LU**	**MA**	**RA**	**LU**	**MA**	**RA**	**LU**
$\boldsymbol{\pi}_0$	0.0976	0.0976	0.0976	0.0949	0.0949	0.0949	0.0916	0.0916	0.0916
	0.3732	0.3732	0.3732	0.3448	0.3448	0.3448	0.3139	0.3140	0.3139
	0.0910	0.0910	0.0910	0.0816	0.0816	0.0816	0.0717	0.0717	0.0717
	0.3477	0.3477	0.3477	0.2963	0.2963	0.2963	0.2458	0.2458	0.2458
$\bar{\pi}_0$	0.9095	0.9095	0.9095	0.8176	0.8176	0.8176	0.7230	0.7231	0.7230
$\boldsymbol{\pi}_1$	0.0153	0.0153	0.0153	0.0308	0.0308	0.0308	0.0461	0.0461	0.0461
	0.0364	0.0364	0.0364	0.0690	0.0690	0.0690	0.0969	0.0969	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	0.0485	0.0485
$\bar{\pi}_1$	0.0847	0.0847	0.0847	0.1585	0.1585	0.1585	0.2196	0.2196	0.2196
$\boldsymbol{\pi}_2$	0.0013	0.0013	0.0013	0.0055	0.0055	0.0055	0.0127	0.0127	0.0127
	0.0021	0.0021	0.0021	0.0081	0.0081	0.0081	0.0176	0.0175	0.0176
	0.0008	0.0008	0.0008	0.0031	0.0031	0.0031	0.0065	0.0065	0.0065
	0.0010	0.0010	0.0010	0.0036	0.0036	0.0036	0.0070	0.0070	0.0070
$\bar{\pi}_2$	0.0052	0.0052	0.0052	0.0203	0.0203	0.0203	0.0438	0.0437	0.0437

queue empty rate is referred to the probability of no customer in the queue. We find that the sum $\sum_{k=0}^{2} \bar{\pi}_k$ 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 5.4 .

According to Table 5.2 and Table 5.3, we find that if the value of p is smaller, the $\boldsymbol{\pi}_i$ value of three methods are very close. But when the value of p is closed to its upper bound, the $\boldsymbol{\pi}_i$'s are a little different.

Table 5.3: Probabilities obtained from three methods with $p = 0.4,\ 0.5,\ 0.6$.

p	0.4			0.5			0.6		
	MA	**RA**	**LU**	**MA**	**RA**	**LU**	**MA**	**RA**	**LU**
	0.0873	0.0874	0.0873	0.0815	0.0817	0.0815	0.0730	0.0727	0.0730
$\boldsymbol{\pi}_0$	0.2798	0.2798	0.2798	0.2410	0.2410	0.2410	0.1956	0.1948	0.1956
	0.0612	0.0613	0.0612	0.0500	0.0502	0.0500	0.0379	0.0377	0.0379
	0.1962	0.1962	0.1962	0.1478	0.1480	0.1478	0.1014	0.1010	0.1014
$\bar{\pi}_0$	0.6245	0.6247	0.6245	0.5203	0.5209	0.5203	0.4079	0.4062	0.4079
	0.0608	0.0608	0.0608	0.0739	0.0740	0.0739	0.0832	0.0834	0.0832
$\boldsymbol{\pi}_1$	0.1187	0.1186	0.1187	0.1321	0.1317	0.1321	0.1336	0.1323	0.1336
	0.0332	0.0333	0.0332	0.0354	0.0355	0.0354	0.0338	0.0340	0.0338
	0.0529	0.0528	0.0529	0.0512	0.0510	0.0512	0.0436	0.0431	0.0436
$\bar{\pi}_1$	0.2656	0.2655	0.2656	0.2926	0.2922	0.2926	0.2942	0.2938	0.2942
	0.0230	0.0229	0.0230	0.0360	0.0360	0.0360	0.0505	0.0504	0.0505
$\boldsymbol{\pi}_2$	0.0295	0.0293	0.0295	0.0423	0.0417	0.0423	0.0533	0.0515	0.0533
	0.0105	0.0105	0.0105	0.0145	0.0145	0.0145	0.0173	0.0162	0.0173
	0.0106	0.0104	0.0106	0.0133	0.0129	0.0133	0.0142	0.0135	0.0142
$\bar{\pi}_2$	0.0736	0.0731	0.0736	0.1061	0.1051	0.1061	0.1353	0.1326	0.1353

Table 5.4: Comparison of queue empty rates of two servers.

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

Inter-departure times

The departure process has three different partitions:

First, when there is no customer in the system, a departure has to wait until at least one customer has occurred followed by a service.

Second, when there is one customer in the system, the inter-departure time can be a function of single processing customer 's remaining service or it could evolve from a customer arrived and its completion of service.

Third, when a departing customer leaves the system with at least two remaining customers, the minimum of the remaining service time of one of these customers or complete service time of another customer becomes the inter-departure time.

Now, we begin by describing the two arrival processes $(\lambda_1, p, \lambda_2) = (10, 0.4, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.4, 5)$, and let $\mu = 10$.

Then, we find that when there remain at least two customers in the system, the inter-departure time characteristics are the same for all these cases ($k \geq 2$). The infinitesimal generator matrix for the departure process $\mathbf{G}_{21}$ has three different segmentations given by

$$\mathbf{G}_{21} = \begin{array}{c|ccc} & \mathbf{0} & \mathbf{1} & 2^+ \\ \hline \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{B}_{11} & \widehat{\mathbf{C}} \\ 2^+ & \mathbf{0} & \mathbf{0} & \mathbf{SS}_{in} \end{array},$$

where

$$\widehat{\mathbf{C}} = \mathbf{C} \cdot \mathbf{1} = \begin{bmatrix} 12 \\ 4 \\ 8 \\ 0 \end{bmatrix},$$

$$\mathbf{SS}_{in} = \mathbf{S}_1 \oplus \mathbf{S}_1 = \begin{bmatrix} -20 \end{bmatrix}.$$

Thus, we have

$$\mathbf{G}_{21} = \begin{bmatrix} -30 & 12 & 6 & 0 & 12 & 0 & 0 & 0 & 0 \\ 5 & -15 & 0 & 6 & 0 & 4 & 0 & 0 & 0 \\ 10 & 0 & -30 & 12 & 0 & 0 & 8 & 0 & 0 \\ 0 & 10 & 5 & -15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -40 & 12 & 6 & 0 & 12 \\ 0 & 0 & 0 & 0 & 5 & -25 & 0 & 6 & 4 \\ 0 & 0 & 0 & 0 & 10 & 0 & -40 & 12 & 8 \\ 0 & 0 & 0 & 0 & 0 & 10 & 5 & -25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -20 \end{bmatrix}.$$

The probabilities of the departure-process system starting in the various states $\mathbf{d} = (\mathbf{d}_0, \mathbf{d}_1, d_{2^+})$ are made up of the departure point probabilites, with

$$\mathbf{d}_0 = \boldsymbol{\pi}_1 \mathbf{A}_{10}/\overline{\lambda},$$

$$\mathbf{d}_1 = \boldsymbol{\pi}_2 \mathbf{A}_{21}/\overline{\lambda},$$

$$d_{2^+} = (\sum_{n=3}^{\infty} \boldsymbol{\pi}_n \mathbf{A}/\overline{\lambda})\mathbf{1}.$$

This series can be written in closed form as

$$d_{2^+} = (\sum_{n=3}^{\infty} \boldsymbol{\pi}_2 \mathbf{R}^{n-2} \mathbf{A}/\overline{\lambda})\mathbf{1},$$

$$= \boldsymbol{\pi}_2[(\mathbf{I}-\mathbf{R})^{-1} - \mathbf{I}]\mathbf{A}/\overline{\lambda})\mathbf{1}.$$

By above equations. We can compute $\mathbf{d}$:

$$\mathbf{d}_0 = (0.1253, 0.2446, 0.0685, 0.1090),$$

$$\mathbf{d}_1 = (0.0946, 0.1215, 0.0434, 0.0436),$$

$$d_{2^+} = 0.1495.$$

Moments of inter-departure times

The stationary inter-departure time is of the phase type distribution characterized by $[\mathbf{d}, \mathbf{G}_{21}]$. Thus, from Neuts [8] we know that the moments of inter-departure times random variable X are given by

$$E[X] = (-1)\mathbf{d}(\mathbf{G}_{21})^{-1}\mathbf{1}.$$

Then we use different p to compute $E[X]$ by Matlab and Promodel :

p	0.1	0.2	0.3	0.4	0.5	0.6
$E[X]$ Matlab	1.0405	0.4846	0.2991	0.2061	0.1500	0.1123
$E[X]$ Promodel	1.00	0.49	0.30	0.21	0.15	0.11

and we use the same parameters to compute $Var[X]$ by Matlab and Promodel,

$$Var[X] = E[X^2] - E[X]^2.$$

p	0.1	0.2	0.3	0.4	0.5	0.6
$Var[X]$ Matlab	1.1703	0.2711	0.1091	0.0541	0.0295	0.0167
$Var[X]$ Promodel	1.1484	0.2796	0.1339	0.0559	0.0344	0.0156

Lag_k correlations between successive departures

To get the lag_k correlations of the output inter-departure times, it is necessary to develop the infinitesimal generator matrix $\widehat{\mathbf{G_2}}$ segmented into two matrices: the internal transitions (without departures) $\mathbf{G_{21}}$ and the matrix containing the departure transition $\mathbf{G}_{22}$ such that $\widehat{\mathbf{G_2}} = \mathbf{G}_{21} + \mathbf{G}_{22}$.

The matrix $\mathbf{G}_{22}$ is characterized as

$$\mathbf{G}_{22} = \begin{array}{c|ccc} & \mathbf{0} & \mathbf{1} & 2^+ \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{A}_{10} & \mathbf{0} & \mathbf{0} \\ 2^+ & \mathbf{0} & (1-t)\widehat{\mathbf{A}}_{21} & t\mathbf{SS}_{out} \end{array},$$

where t is the probability that departure leaves the system with at least two customers remaining and $(1-t)$ is the probability that only one customer remains.

We define t as

$$t = \frac{\sum_{a=3}^{\infty} \boldsymbol{\pi}_a \mathbf{1}}{\boldsymbol{\pi}_2 \mathbf{1} + \sum_{a=3}^{\infty} \boldsymbol{\pi}_a \mathbf{1}} = \frac{d_{2^+}}{\mathbf{d}_1 + d_{2^+}} = \frac{d_{2^+}}{1 - \mathbf{d}_0 \mathbf{1}} = 0.3303.$$

Two of sub-matrices in $\mathbf{G}_{22}$ are given by

$$\widehat{\mathbf{A}}_{21} = \boldsymbol{\theta} \otimes (\mathbf{S_{1o}} \oplus \mathbf{S}_{1o}) = [0.1838, 0.4412, 0.1103, 0.2647] \otimes [20],$$

$$\mathbf{SS}_{out} = \mathbf{S}_{1o} \oplus \mathbf{S}_{1o} = [20].$$

Thus, we have

$$\mathbf{G}_{22} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.4620 & 5.9088 & 1.4772 & 3.5453 & 6.6066 \end{bmatrix}.$$

Given $\mathbf{G}_{21}$ and $\mathbf{G}_{22}$ matrices, from Bodrog, Horvath, Telek [2], and Telek, Horvath [11], we have that the lag_k correlation is computed by

$$lag_k = \frac{(\overline{\lambda})^2 \mathbf{d}(-\mathbf{G}_{21})^{-1}((-\mathbf{G}_{21})^{-1}\mathbf{G}_{22})^k(-\mathbf{G}_{21})^{-1}\mathbf{1} - 1}{2(\overline{\lambda})^2 \mathbf{d}(-\mathbf{G}_{21})^{-2}\mathbf{1} - 1)},$$

where $\overline{\lambda} = (p[(\frac{1}{\lambda_1}p + (\frac{1}{\lambda_1} + \frac{1}{\lambda_2})(1-p)]^{-1} + p[(\frac{1}{\gamma_1}p + (\frac{1}{\gamma_1} + \frac{1}{\gamma_2})(1-p)]^{-1} = 4.8529)$ is the average effective arrival rate. We can compute lag_k $(k = 1, 2, 3, 4)$,

$$lag_1 = 0.0342, lag_2 = 0.0224, lag_3 = 0.0206, lag_4 = 0.0203.$$

5.2 Queueing models with three servers

$M/M/3$ queueing models

Once more, we present a numerical example of $M/M/3$ queues to verify our model. 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 MA, RA, and LU methods, we take $\bar{\pi}_0\ \bar{\pi}_1\ \cdots\ \bar{\pi}_{12}\ \bar{\pi}_{13}$ to compare the values. We find that they are the same.

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
MA	0.2105	0.3158	0.2368	0.1184	0.0592	0.0296	0.0148
RA	0.2105	0.3158	0.2368	0.1184	0.0592	0.0296	0.0148
LU	0.2105	0.3158	0.2368	0.1184	0.0592	0.0296	0.0148

$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
MA	0.0074	0.0037	0.0018	0.0009	0.0005	0.0002	0.0001
RA	0.0074	0.0037	0.0018	0.0009	0.0005	0.0002	0.0001
LU	0.0074	0.0037	0.0018	0.0009	0.0005	0.0002	0.0001

In this example, we find the probability of idle system is 0.210526, which is the same as $\bar{\pi}_0$. 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 $1 - \bar{\pi}_0 - \frac{2}{3}\bar{\pi}_1 - \frac{1}{3}\bar{\pi}_2$, it gives the busy rate of queues as follows

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

The matrix geometric solution procedure

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

From the matrix geometric solution 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

$$\boldsymbol{\pi}_0 = (0.0887, 0.2838, 0.0622, 0.1988),$$

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

$$\boldsymbol{\pi}_2 = (0.0240, 0.0297, 0.0109, 0.0103),$$

and

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

Ramaswami's formula

By applying RA method, we have

$$\boldsymbol{\pi}_0 = (0.0887, 0.2838, 0.0622, 0.1988),$$

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

$$\boldsymbol{\pi}_2 = (0.0240, 0.0297, 0.0109, 0.0103),$$

and

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

LU factorization

By using LU factorization given in previous section, we obtain

$$\boldsymbol{\pi}_0 = (0.0887, 0.2838, 0.0622, 0.1988),$$

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

$$\boldsymbol{\pi}_2 = (0.0240, 0.0297, 0.0109, 0.0103),$$

and

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

Numerical experiments by changing p

We observe the effect of changing p on the numerical results obtained from three methods. Here, we have $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, $\mu = 10$, and $0 \leqslant p < w = 1$, for $p = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9$. Tables 5.5-5.7 show the comparison of numerical results.

Table 5.5: Probabilities obtained from three methods with $p = 0.2,\ 0.3,\ 0.4$.

p	0.2			0.3			0.4		
	MA	**RA**	**LU**	**MA**	**RA**	**LU**	**MA**	**RA**	**LU**
$\boldsymbol{\pi}_0$	0.0950	0.0950	0.0950	0.0921	0.0921	0.0921	0.0887	0.0887	0.0887
	0.3452	0.3452	0.3452	0.3155	0.3155	0.3155	0.2838	0.2838	0.2838
	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	0.2469	0.1988	0.1988	0.1988
$\bar{\pi}_0$	0.8186	0.8186	0.8186	0.7266	0.7267	0.7266	0.6335	0.6335	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	0.1204	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	0.0534	0.0534
$\bar{\pi}_1$	0.1587	0.1587	0.1587	0.2208	0.2208	0.2208	0.2698	0.2698	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	0.0297
	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	0.0103	0.0103
$\bar{\pi}_2$	0.0203	0.0203	0.0203	0.0441	0.0440	0.0441	0.0749	0.0749	0.0749
$\boldsymbol{\pi}_3$	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	0.0056	0.0056
	0.0003	0.0003	0.0003	0.0011	0.0011	0.0011	0.0026	0.0026	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	0.0164	0.0165

Table 5.6: Probabilities obtained from three methods with $p = 0.5,\ 0.6,\ 0.7$.

p	0.5			0.6			0.7		
	MA	**RA**	**LU**	**MA**	**RA**	**LU**	**MA**	**RA**	**LU**
	0.0847	0.0847	0.0847	0.0796	0.0797	0.0796	0.0728	0.0730	0.0728
$\boldsymbol{\pi}_0$	0.2493	0.2493	0.2493	0.2112	0.2113	0.2112	0.1679	0.1681	0.1679
	0.0519	0.0519	0.0519	0.0411	0.0412	0.0411	0.0299	0.0302	0.0299
	0.1527	0.1527	0.1527	0.1091	0.1092	0.1091	0.0690	0.0693	0.0690
$\bar{\pi}_0$	0.5386	0.5386	0.5386	0.4410	0.4414	0.4410	0.3396	0.3406	0.3396
	0.0775	0.0776	0.0775	0.0921	0.0922	0.0921	0.1044	0.1046	0.1044
$\boldsymbol{\pi}_1$	0.1366	0.1366	0.1366	0.1440	0.1438	0.1440	0.1392	0.1387	0.1392
	0.0369	0.0369	0.0369	0.0370	0.0371	0.0370	0.0335	0.0337	0.0335
	0.0525	0.0525	0.0525	0.0462	0.0462	0.0462	0.0352	0.0351	0.0352
$\bar{\pi}_1$	0.3035	0.3036	0.3035	0.3193	0.3193	0.3193	0.3123	0.3121	0.3123
	0.0390	0.0390	0.0390	0.0581	0.0581	0.0581	0.0809	0.0809	0.0809
$\boldsymbol{\pi}_2$	0.0434	0.0433	0.0434	0.0567	0.0564	0.0567	0.0663	0.0655	0.0663
	0.0154	0.0154	0.0154	0.0194	0.0194	0.0194	0.0215	0.0216	0.0215
	0.0130	0.0130	0.0130	0.0142	0.0141	0.0142	0.0131	0.0129	0.0131
$\bar{\pi}_2$	0.1108	0.1107	0.1108	0.1484	0.1480	0.1484	0.1818	0.1809	0.1818
	0.0138	0.0138	0.0138	0.0256	0.0256	0.0256	0.0435	0.0434	0.0435
$\boldsymbol{\pi}_3$	0.0106	0.0105	0.0106	0.0171	0.0168	0.0171	0.0243	0.0235	0.0243
	0.0047	0.0047	0.0047	0.0074	0.0074	0.0074	0.0100	0.0100	0.0100
	0.0028	0.0027	0.0028	0.0038	0.0037	0.0038	0.0043	0.0041	0.0043
$\bar{\pi}_3$	0.0319	0.0317	0.0319	0.0539	0.0535	0.0539	0.0821	0.0810	0.0821

Table 5.7: Probabilities obtained from three methods with $p = 0.8,\ 0.9$.

p	0.8			0.9		
	MA	**RA**	**LU**	**MA**	**RA**	**LU**
	0.0625	0.0631	0.0625	0.0446	0.0453	0.0446
$\boldsymbol{\pi}_0$	0.1173	0.1174	0.1173	0.0568	0.0565	0.0568
	0.0184	0.0189	0.0184	0.0072	0.0079	0.0072
	0.0344	0.0379	0.0344	0.0090	0.0099	0.0090
$\bar{\pi}_0$	0.2326	0.2373	0.2326	0.1176	0.1196	0.1176
	0.1106	0.1111	0.1106	0.0984	0.0991	0.0984
$\boldsymbol{\pi}_1$	0.1165	0.1154	0.1165	0.0672	0.0650	0.0672
	0.0255	0.0259	0.0255	0.0125	0.0133	0.0125
	0.0208	0.0208	0.0208	0.0065	0.0067	0.0065
$\bar{\pi}_1$	0.2734	0.2732	0.2734	0.1846	0.1841	0.1846
	0.1042	0.1044	0.1042	0.1134	0.1136	0.1134
$\boldsymbol{\pi}_2$	0.0662	0.0643	0.0662	0.0453	0.0422	0.0453
	0.0200	0.0202	0.0200	0.0120	0.0126	0.0120
	0.0093	0.0090	0.0093	0.0035	0.0034	0.0035
$\bar{\pi}_2$	0.1997	0.1979	0.1997	0.1742	0.1718	0.1742
	0.0676	0.0675	0.0676	0.0890	0.0887	0.0890
$\boldsymbol{\pi}_3$	0.0292	0.0272	0.0292	0.0240	0.0207	0.0240
	0.0112	0.0112	0.0112	0.0082	0.0083	0.0082
	0.0038	0.0034	0.0038	0.0017	0.0015	0.0017
$\bar{\pi}_3$	0.1118	0.1093	0.1118	0.1229	0.1192	0.1229

Promodel

We compare the values of $\sum_{k=0}^{3} \bar{\pi}_k$ 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 5.8 shows the comparison of numerical results.

Table 5.8: Comparison of the queue empty rates of three servers.

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

It shows the same situation like the queueing with two servers. According to Tables 5.5-5.7, we find that if the value of p is smaller, the $\boldsymbol{\pi}_i$ value of three methods are very close. But when the value of p is closed to its upper bound, the $\boldsymbol{\pi}_i$'s are a little different.

Inter-departure times

Now, we begin by describing the two arrival processes as $(\lambda_1, p, \lambda_2) = (10, 0.4, 10)$, $(\gamma_1, p, \gamma_2) = (20, 0.4, 5)$, and $\mu = 10$.

Then, we find that when there remain at least three customers in the system, the inter-departure time characteristics are the same for all these cases ($k \geq 3$). The infinitesimal generator matrix for the departure process $\mathbf{G}_{31}$ has three different segmentations are given by

$$\mathbf{G}_{31} = \begin{array}{c|cccc} & \mathbf{0} & \mathbf{1} & \mathbf{2} & 3^+ \\ \hline \mathbf{0} & \mathbf{B}_{00} & \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{B}_{11} & \mathbf{C} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{22} & \widehat{\mathbf{C}} \\ 3^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{SS}_{in} \end{array},$$

where sub-matrices given by

$$\widehat{\mathbf{C}} = \mathbf{C} \cdot \mathbf{1} = \begin{bmatrix} 12 \\ 4 \\ 8 \\ 0 \end{bmatrix},$$

$$\mathbf{SS}_{in} = \mathbf{S}_1 \oplus \mathbf{S}_1 \oplus \mathbf{S}_1 = \begin{bmatrix} -30 \end{bmatrix}.$$

Thus, we have

$$\mathbf{G}_{31} = \begin{bmatrix}
-30 & 12 & 6 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
5 & -15 & 0 & 6 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & 0 & -30 & 12 & 0 & 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 5 & -15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -40 & 12 & 6 & 0 & 12 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 5 & -25 & 0 & 6 & 0 & 4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 10 & 0 & -40 & 12 & 0 & 0 & 8 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 10 & 5 & -25 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -50 & 12 & 6 & 0 & 12 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & -35 & 0 & 6 & 4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & -50 & 12 & 8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10 & 5 & -35 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -30
\end{bmatrix}.$$

The probabilities of the departure-process system starting in the various states $\mathbf{d} = (\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, d_{3^+})$ are made up of the departure point probabilites, with

$$\mathbf{d}_0 = \boldsymbol{\pi}_1 \mathbf{A}_{10} / \overline{\lambda},$$

$$\mathbf{d}_1 = \boldsymbol{\pi}_2 \mathbf{A}_{21} / \overline{\lambda},$$

$$\mathbf{d}_2 = \boldsymbol{\pi}_3 \mathbf{A}_{32} / \overline{\lambda},$$

$$d_{3^+} = (\sum_{n=4}^{\infty} \boldsymbol{\pi}_n \mathbf{A} / \overline{\lambda}) \mathbf{1}.$$

This series can be written in closed form as

$$d_{3^+} = (\sum_{n=4}^{\infty} \boldsymbol{\pi}_3 \mathbf{R}^{n-3} \mathbf{A}/\overline{\lambda})\mathbf{1}$$

$$= \boldsymbol{\pi}_3[(\mathbf{I}-\mathbf{R})^{-1} - \mathbf{I}]\mathbf{A}/\overline{\lambda})\mathbf{1}.$$

By above equations. We can compute $\mathbf{d}$

$$\mathbf{d}_0 = (0.1280, 0.2480, 0.0698, 0.1100)$$

$$\mathbf{d}_1 = (0.0989, 0.1226, 0.0448, 0.0424)$$

$$\mathbf{d}_2 = (0.0407, 0.0347, 0.0158, 0.0105)$$

$$d_{3^+} = 0.0338$$

Moments of inter-departure times

The stationary inter-departure time is of the phase type distribution characterized by $[\mathbf{d}, \mathbf{G}_{31}]$. Thus, from Neuts [8] we know that the moments of inter-departure times random variable X are given by

$$E[X] = (-1)\mathbf{d}(\mathbf{G}_{31})^{-1}\mathbf{1}.$$

Then we use different p to compute $E[X]$ by Matlab and Promodel :

p	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$E[X]$ Matlab	0.4846	0.2991	0.2061	0.1500	0.1123	0.0851	0.0643	0.0475
$E[X]$ Promodel	0.50	0.31	0.21	0.15	0.11	0.08	0.06	0.04

and we use the same parameters to compute $Var[X]$ by Matlab and Promodel,

$$Var[X] = E[X^2] - E[X]^2.$$

p	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$Var[X]$ Matlab	0.2712	0.1093	0.0543	0.0299	0.0172	0.0099	0.0055	0.0027
$Var[X]$ Promodel	0.32	0.1176	0.0516	0.0344	0.0156	0.0119	0.0051	0.0025

Lag_k correlations between successive departures

To get the lag_k correlations of the output inter-departure times, it is necessary to develop the infinitesimal generator matrix $\widehat{\mathbf{G_3}}$ segmented into two matrices: the internal transitions (without departures) $\mathbf{G}_{31}$ and the matrix containing the departure transition $\mathbf{G}_{32}$ such that $\widehat{\mathbf{G_3}} = \mathbf{G}_{31} + \mathbf{G}_{32}$.

The matrix $\mathbf{G}_{32}$ is characterized as

$$\mathbf{G}_{32} = \begin{array}{c|cccc} & \mathbf{0} & \mathbf{1} & \mathbf{2} & 3^+ \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{A}_{10} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{2} & \mathbf{0} & \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} \\ 3^+ & \mathbf{0} & \mathbf{0} & (1-t)\,\widehat{\mathbf{A}}_{32} & t\mathbf{SS}_{out} \end{array},$$

where t is the probability that departure leaves the system with at least three customers remaining. We compute t as

$$t = \frac{\sum_{a=4}^{\infty} \boldsymbol{\pi}_a \mathbf{1}}{\boldsymbol{\pi}_3 \mathbf{1} + \sum_{a=4}^{\infty} \boldsymbol{\pi}_a \mathbf{1}} = \frac{d_{3^+}}{\mathbf{d}_2 \mathbf{1} + d_{3^+}} = \frac{d_{3^+}}{1 - \mathbf{d}_0 \mathbf{1} - \mathbf{d}_1 \mathbf{1}} = 0.2494.$$

Two of these sub-matrices are given by

$$\widehat{\mathbf{A}}_{32} = \boldsymbol{\theta} \otimes (\mathbf{S}_{1o} \oplus \mathbf{S}_{1o} \oplus \mathbf{S}_{1o}) = [0.1838, 0.4412, 0.1103, 0.2647] \otimes [30],$$

$$\mathbf{SS}_{out} = \mathbf{S}_{1o} \oplus \mathbf{S}_{1o} \oplus \mathbf{S}_{1o} = [30].$$

Thus, we have

$$
\mathbf{G}_{32} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4.1391 & 9.9339 & 2.4835 & 5.9604 & 7.4831
\end{bmatrix}.
$$

Given $\mathbf{G}_{31}$ and $\mathbf{G}_{32}$ matrices, from Bodrog, Horvath, Telek [2], and Telek, Horvath [11], we have that the lag_k correlation is computed by

$$
lag_k = \frac{(\overline{\lambda})^2 \mathbf{d}(-\mathbf{G}_{31})^{-1}((-\mathbf{G}_{31})^{-1}\mathbf{G}_{32})^k(-\mathbf{G}_{31})^{-1}\mathbf{1} - 1}{2(\overline{\lambda})^2 \mathbf{d}(-\mathbf{G}_{31})^{-2}\mathbf{1} - 1)},
$$

where $\overline{\lambda} = (p[(\frac{1}{\lambda_1}p + (\frac{1}{\lambda_1} + \frac{1}{\lambda_2})(1-p)]^{-1} + p[(\frac{1}{\gamma_1}p + (\frac{1}{\gamma_1} + \frac{1}{\gamma_2})(1-p)]^{-1} = 4.8529)$ is the average effective arrival rate. We can compute lag_k $(k = 1, 2, 3, 4)$,

$$
lag_1 = 0.0344, lag_2 = 0.0133, lag_3 = 0.0111, lag_4 = 0.0106.
$$

5.3 Queueing models with more than twenty servers

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 $\mu = 0.8$.

The numerical results are compared in three different approaches, **MA** represents matrix geometric solution procedure, **RA** represents Ramaswami's formula, and **LU** represents LU factorization.

Then, it gives the solutions.

$\bar{\pi}_i$	$\bar{\pi}_0$	$\bar{\pi}_1$	$\bar{\pi}_2$	$\bar{\pi}_3$	$\bar{\pi}_4$	$\bar{\pi}_5$	$\bar{\pi}_6$
MA	0.0040	0.0202	0.0525	0.0939	0.1303	0.1491	0.1464
RA	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

$\bar{\pi}_i$	$\bar{\pi}_7$	$\bar{\pi}_8$	$\bar{\pi}_9$	$\bar{\pi}_{10}$	$\bar{\pi}_{11}$	$\bar{\pi}_{12}$	$\bar{\pi}_{13}$
MA	0.1269	0.0989	0.0703	0.0461	0.0281	0.0161	0.0087
RA	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

$\bar{\pi}_i$	$\bar{\pi}_{14}$	$\bar{\pi}_{15}$	$\bar{\pi}_{16}$	$\bar{\pi}_{17}$	$\bar{\pi}_{18}$	$\bar{\pi}_{19}$	$\bar{\pi}_{20}$
MA	0.0045	0.0022	0.0010	0.0005	0.0002	0.0000	0.0000
RA	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

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

Numerical experiments by changing of p with $n = 20$

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 $0 \leqslant p < w = 0.8098$.

Table 5.9 shows the comparison of numerical results with three different methods.

Table 5.9: 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	0.9953	0.9828	0.9468	0.8582	0.6570
LU factorization	1.0000	0.9998	0.9990	0.9953	0.9822	0.9430	0.8421	0.6122

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 $\mu = 0.9$.

The numerical results are compared in three different approaches. **MA** represents matrix geometric solution procedure, **RA** represents Ramaswami's formula, and **LU** represents LU factorization.

Then, it gives the solutions.

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

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

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

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

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

Numerical experiments by changing of p with $n = 25$

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 $0 \leqslant p < w = 0.9206$.

Table 5.10 shows the comparison of numerical results with three different methods.

Table 5.10: 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	0.9943	0.9779	0.9266	0.7835
LU factorization	1.0000	1.0000	0.9998	0.9986	0.9942	0.9769	0.9201	0.7562

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 $\mu = 1$.

The numerical results are compared in three different approaches, **MA** represents matrix geometric solution procedure, **RA** represents Ramaswami's formula, and **LU** represents LU factorization.

Then, it gives the solutions.

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

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

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

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

$\bar{\pi}_i$	$\bar{\pi}_{28}$	$\bar{\pi}_{29}$	$\bar{\pi}_{30}$
MA	0.0000	0.0000	0.0000
RA	0.0000	0.0000	0.0000
LU	0.0000	0.0000	0.0000

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

Numerical experiments by changing of p with $n = 30$

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 $0 \leqslant p < w = 1$.

Table 5.11 shows the comparison of numerical results with three different methods.

Table 5.11: 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	0.9973	0.9864	0.9441	0.7655
LU factorization	1.0000	1.0000	0.9999	0.9995	0.9973	0.9860	0.9377	0.7493

Numerical comparsion with simulation of twenty servers

We compare the values of $\sum_{k=0}^{20} \bar{\pi}_k$ 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 to 0.75, $\mu = 0.8$. Table 5.12 shows the comparison of numerical results.

Table 5.12: Comparison of queue empty rates 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

Figure 5.1: The queue empty rate determined by four different methods

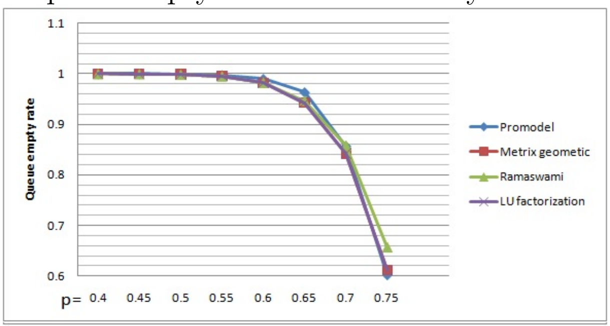
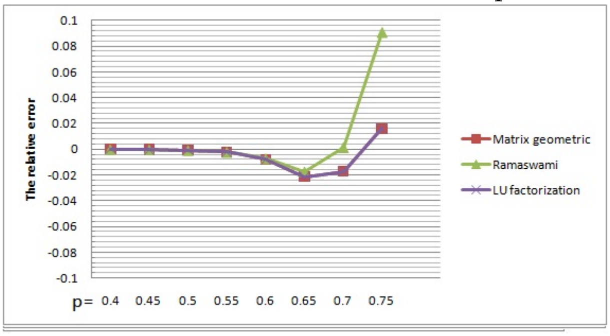


Figure 5.2: Relative errors of three methods compared with Promodel



Accorrding to two Figures 5.1-2, we can find that when p is close to its upper bound (0.8098), the relative error becomes large.

Numerical comparsion with simulation of twenty-five servers

We compare the values of $\sum_{k=0}^{25} \bar{\pi}_k$ 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 to 0.85, $\mu = 0.9$. Table 5.13 shows the comparison of numerical results.

Table 5.13: 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

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.

Figure 5.3: The queue empty rate determined by four different methods

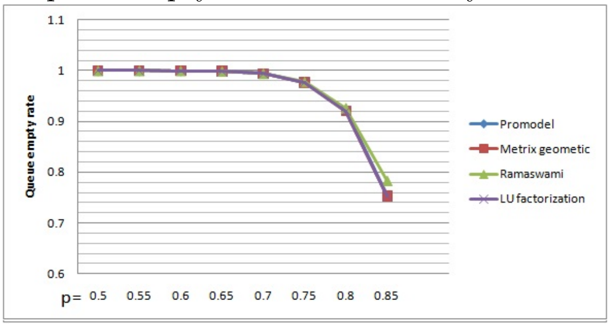
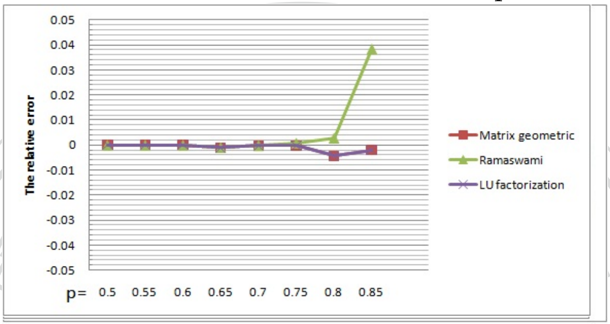


Figure 5.4: Relative errors of three methods compared with Promodel



Numerical comparsion with simulation of thirty servers

We compare the values of $\sum_{k=0}^{30} \bar{\pi}_k$ 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 to 0.95, $\mu = 1$. Table 5.14 shows the comparison of numerical results.

Table 5.14: 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

Accorrding to two Figures 5.5-6, we can find that when p is close to its upper

Figure 5.5: The queue empty rate determined by four different methods

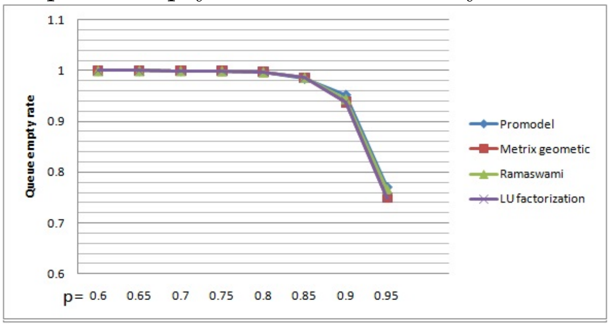
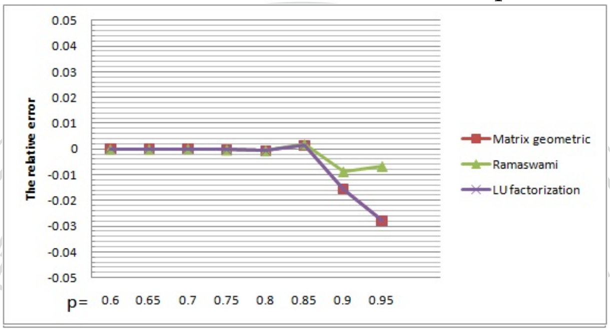


Figure 5.6: Relative errors of three methods compared with Promodel



bound (1), the relative error becomes large.

Inter-departure times with simulation of twenty servers

Now, we begin by describing the two arrival processes as $(\lambda_1, p, \lambda_2) = (10, p, 10), (\gamma_1, p, \gamma_2) = (20, p, 5)$, and let $\mu = 0.8$.

Moments of inter-departure times with $n = 20$

We obtain the moments of inter-departure times.

p	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
$E[X]$ Matlab	0.2061	0.1750	0.1500	0.1295	0.1123	0.0997	0.0851	0.0741
$E[X]$ Promodel	0.21	0.18	0.15	0.13	0.12	0.10	0.08	0.07

Then we use the same parameters to compute $Var[X]$ by Matlab and Promodel.

$$Var[X] = E[X^2] - E[X]^2$$

p	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
$Var[X]$ Matlab	0.0451	0.0325	0.0239	0.0178	0.0143	0.0101	0.0076	0.0057
$Var[X]$ Promodel	0.0459	0.0376	0.0275	0.0171	0.0128	0.0100	0.0070	0.0049

The lag_k correlations of the output inter-departure times.

p	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75
lag_1	0.0245	0.0253	0.0258	0.0260	0.0258	0.0248	0.0223	0.0163
lag_2	0.0208	0.0220	0.0228	0.0234	0.0236	0.0230	0.0209	0.0154
lag_3	0.0177	0.0191	0.0202	0.0211	0.0215	0.0213	0.0195	0.0146
lag_4	0.0151	0.0166	0.0179	0.0190	0.0197	0.0197	0.0182	0.0137

5.4 Numerical experiments with more than forty servers

Condition numbers

In this section, we compare the condition numbers given by **RA** and **LU** methods with forty, fifty, sixty, seventy and eighty servers.

First, we recall the condition number. In our example, the condition number associated with the linear system $\boldsymbol{\pi}^*\mathbf{Q}_3 = [1\ \mathbf{0}]$ in (3.6) and (3.7) gives a bound on how inaccurate the solution $\boldsymbol{\pi}^*$ will be after an approximate solution. In particular, if the condition number is large, even a small error in $[1\ \mathbf{0}]$ may cause a large error in $\boldsymbol{\pi}^*$. On the other hand, if the condition number is small then the error in $\boldsymbol{\pi}^*$ will not be much bigger than the error in $[1\ \mathbf{0}]$.

Let $\boldsymbol{\xi}$ be the error in $[1\ \mathbf{0}]$. Since that $\mathbf{Q}_3$ is a square matrix, the error in the solution $[1\ \mathbf{0}]\mathbf{Q}_3^{-1}$ is $\boldsymbol{\xi}\mathbf{Q}_3^{-1}$. The ratio of the relative error in the solution to the relative error in $[1\ \mathbf{0}]$ is

$$\frac{\|\boldsymbol{\xi}\mathbf{Q}_3^{-1}\|/\|[1\ \mathbf{0}]\mathbf{Q}_3^{-1}\|}{\|\boldsymbol{\xi}\|/\|[1\ \mathbf{0}]\|}.$$

This is easily transformed to

$$(\|\boldsymbol{\xi}\mathbf{Q}_3^{-1}\|)/\|\boldsymbol{\xi}\|)\cdot(1/\|[1\ \mathbf{0}]\mathbf{Q}_3^{-1}\|).$$

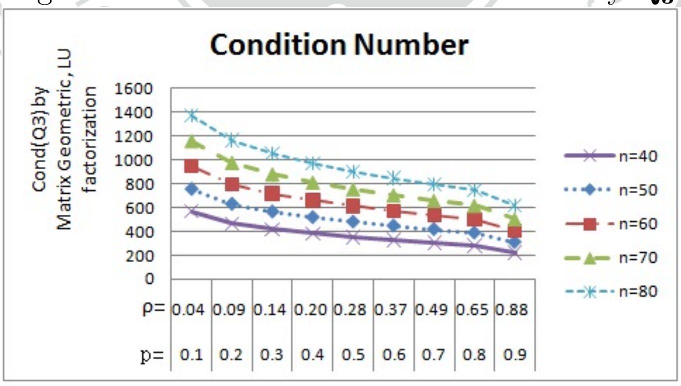
The maximum value is easily seen to be the product of the two operator norms:

$$\kappa(\mathbf{Q}_3) = \|\mathbf{Q}_3\|\cdot\|\mathbf{Q}_3^{-1}\|,$$

$\kappa(\mathbf{Q}_3)$ is the condition number of $\mathbf{Q_3}$.

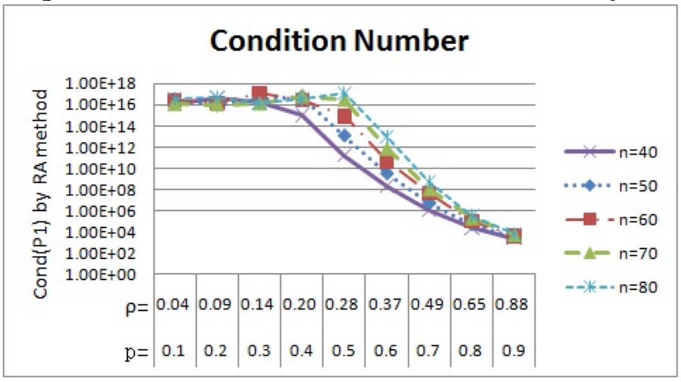
We consider the matrix $\mathbf{Q}_3$ in Matrix geometric method and LU factorization. Because of $\boldsymbol{\pi}^*\mathbf{Q}_3 = [1\ \mathbf{0}]$, we apply $\mathbf{Q}_3$ to compute the condition number by Matlab in the following. The parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and consider $\rho = \overline{\lambda}/(n\mu)$ from 0.04 to 0.88 with respect to different p.

Figure 5.7: Condition number determined by $\mathbf{Q_3}$



Next, we define the matrix $\mathbf{P} = \mathbf{B}_0 - \mathbf{B}_1\mathbf{V}_0^{-1}\mathbf{B}_{-1}$ in (3.9) and (3.7) with **RA** method. Let the first column of $\mathbf{P}$ be replaced by the column vector $(\mathbf{1}, \cdots, \mathbf{1}, (\mathbf{I} - \mathbf{R})^{-1}\cdot\mathbf{1})^T$. Then, the modified $\mathbf{P}$ is rewritten as a new matrix $\mathbf{P}_1$. Because of $\boldsymbol{\pi}^*\mathbf{P}_1 = [1\ \mathbf{0}]$, we apply $\mathbf{P}_1$ to compute the condition number by Matlab in the following. The parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and consider $\rho = \overline{\lambda}/(n\mu)$ from 0.04 to 0.88 with respect to different p's.

Figure 5.8: Condition number determined by $\mathbf{P}_1$



Queue empty rate

In this section, we consider the queue empty rate by comparing **MA**, **RA** and **LU** methods wit forty, fifty, sixty, seventy and eighty servers.

The parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and consider $\rho = \overline{\lambda}/(n\mu)$ from 0.04 to 0.88 with respect to different p's.

Figure 5.9: The queue empty rate computed by LU method

Then we use the **RA** method to compute the queue empty rate. The parameters of arrival processes are given by $(\lambda_1, p, \lambda_2) = (10, p, 10)$, $(\gamma_1, p, \gamma_2) = (20, p, 5)$, and from 0.04 to 0.88 with respect to different p's.

Figure 5.10: The queue empty rate computed by RA method

Chapter 6

Conclusion

In this thesis, we present a new computing scheme for computing the stationary probabilities of a phase-type queueing model with multiple servers. The matrix geometric solution procedure has been compared 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 sub-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 solution 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. At last, we use two different methods (Matlab and Promodel) to compute the moments of inter-departure times and the variance. We can find the values of two different methods are almost the same.

Bibliography

[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.

Appendix A

Theorem LU.(Thm 3.5.2 in [9])

Suppose that $\mathbf{A} \in \mathbf{M}_{n\times n}$ and that rank $\mathbf{A} = k$. If

$$det\mathbf{A}(\{1, \cdots, j\}) \neq 0, \quad j = 1, \cdots, k,$$

then $\mathbf{A}$ may be factored as

$$\mathbf{A} = \mathbf{LU}$$

with $\mathbf{L} \in \mathbf{M}_{n\times n}$ lower triangular and $\mathbf{U} \in \mathbf{M}_{n\times n}$ upper triangular. Furthermore, the factorization may be chosen so that either $\mathbf{L}$ or $\mathbf{U}$ is nonsingular; both $\mathbf{L}$ and $\mathbf{U}$ may be chosen nonsingular if and only if $k = n$, that is, if and only if $\mathbf{A}$ is nonsingular.

Proof :

We first show that, under the assumption on leading minors, $\mathbf{A}(\{1, \cdots, k\})$ may be factored as $\mathbf{L}(\{1, \cdots, k\})\ \mathbf{U}(\{1, \cdots, k\})$, with both nonsingular. It is possible to solve for the relevant entries of $\mathbf{L}$ and $\mathbf{U}$, one by one. Let $\mathbf{L} = [l_{ij}]$ and $\mathbf{U} = [u_{ij}]$. Set $u_{11} = 1$, and let $l_{i1} = a_{i1}$, $i = 1, \cdots, k$.

Solve for

$$u_{1j} = \frac{a_{1j}}{l_{11}}, \quad j = 2, \cdots, k.$$

Continue. Set $u_{22} = 1$ and let $l_{i2} = a_{i2} - l_{i1}u_{12}$, $i = 2, \cdots, k$. Solve for

$$u_{2j} = \frac{a_{2j} - l_{21}u_{1j}}{l_{22}}, \quad j = 3, \cdots, k.$$

Continue, letting successive diagonal entries of $\mathbf{U}$ be 1 and then solving for the next column of $\mathbf{L}(\{1, \cdots, k\})$ and then the next row of $\mathbf{U}(\{1, \cdots, k\})$.

Each time there is one equation in one unknown to be solved. This equation will be solvable since each l_{ii} is nonzero (because $det\ \mathbf{L}(\{1, \cdots, i\}) \times det\ \mathbf{U}(\{1, \cdots, i\}) = det\ \mathbf{A}(\{1, \cdots, i\})$). this completes the factorization of $\mathbf{A}(\{1, \cdots, k\})$.

Partition A. Since rank $\mathbf{A} = k =$ rank $\mathbf{A_{11}}$, we see that the rows of $[\mathbf{A}_{21}\ \mathbf{A}_{22}]$ are unique linear combinations of rows of $[\mathbf{A}_{11}\ \mathbf{A}_{12}]$, that is

$$\mathbf{A}_{21} = \mathbf{B}\mathbf{A}_{11} \ \ \textit{and} \ \ \mathbf{A}_{22} = \mathbf{B}\mathbf{A}_{12},$$

for some uniquely determined $\mathbf{B} \in \mathbf{M}_{n-k,k}$. Now partition the desired $\mathbf{L}$ and $\mathbf{U}$, noting that nonsingular $\mathbf{L}_{11}$ and $\mathbf{U}_{11}$ have been

$$\mathbf{U}_{12} = \mathbf{L}_{11}^{-1}\mathbf{A}_{12} \ \ \textit{and} \ \ \mathbf{L}_{21} = \mathbf{A}_{21}\mathbf{U}_{11}^{-1}.$$

Then

$\mathbf{A}_{22} = \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} = \mathbf{A}_{21}\mathbf{U}_{11}^{-1}\mathbf{L}_{11}^{-1}\mathbf{A}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} = \mathbf{B}\mathbf{A}_{11}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} = \mathbf{A}_{22} + \mathbf{L}_{22}\mathbf{U}_{22}$

To complete the factorization, it is necessary and sufficient that $\mathbf{L}_{22}\mathbf{U}_{22} = 0$

We may, for example, choose $\mathbf{L}_{22}$ (respectively $\mathbf{U}_{22}$) to be any nonsingular lower (respectively upper) triangular matrix in $\mathbf{M}_{n-k}$ we like and choose $\mathbf{U}_{22}$ (respectively $\mathbf{L}_{22}$) to be 0. Since $\mathbf{L}_{11}$ and $\mathbf{U}_{11}$ are nonsingular, either $\mathbf{L}$ or $\mathbf{U}$ may be chosen to be nonsingular. If $k = n$, $\mathbf{L} = \mathbf{L}_{11}$ and $\mathbf{U} = \mathbf{U}_{11}$ will be nonsingular; if $k < n$, not both $\mathbf{L}$ and $\mathbf{U}$ can be nonsingular because $\mathbf{A}$ is singular. This completes the proof.

Appendix B

Theorem D.(Thm 3.5.6 in [9])

Let $\mathbf{Z}_k \in \mathbf{M}_{k\times k}$ be nonsingular. Then there is a permutation matrix $\mathbf{D} \in \mathbf{M}_{\mathbf{k}\times\mathbf{k}}$ such that

$$det(\mathbf{D}^T\mathbf{Z}_k)(\{1, \cdots, j\}) \neq 0, \quad j = 1, \cdots, k$$

Note that $\mathbf{D}^T\mathbf{Z}_k$ is just a recordering of the rows of $\mathbf{Z}_k$.

Proof :

The demonstrations is by induction on k, If k =1 or 2, the result is clear by inspection; suppose that it is valid up to and including $k-1$. Consider a nonsingular $\mathbf{Z}_k \in \mathbf{M}_{k\times k}$ matrix and delete its last column. The remaining $k-1$ cloumns are linearly independent and hence contain $k-1$ linearly independent rows. Permute these rows to the first $k-1$ positions and apply the induction hypothesis to the nonsingular upper $(k-1)$-by-$(k-1)$ submatrix. This determines a desired overall permutation. Since $\mathbf{D}^T\mathbf{Z}_k$ is nonsingular, the proof is complete.

Appendix C

Following is the code of the program.

```
function Rmatrixn(lam1,lam2,gam1,gam2,mu,p,n)
%%%%%%% input data
lam1=10;
lam2=10;
gam1=5;
gam2=20;
mu=input('\n    Please input the service rate at all phase, \mu=');
p=input('\n    Please input the probability that the customer leaves the system after the first phase of the service time, \p=');
n=input('\n    Please input the number of server,  \n=');
%%%%%%%%%%%%%%%%%% The range of p%%%%%%%%%%%%%%%%%%
prange=1/2/(gam1*gam2*lam1+lam1*lam2*gam1+n*mu*lam1*gam1)*(lam1*lam2*gam2+gam1*gam2*lam1+lam1*lam2*gam1+gam1*gam2*lam2+n*mu*lam2*gam1+
      2*n*mu*lam1*gam1+n*mu*lam1*gam2-(2*lam1*lam2^2*gam1^2*gam2+2*lam1^2*lam2*gam1^2*gam2+2*lam1^2*lam2^2*gam1*gam2+
      2*gam1*gam2^2*lam1^2*lam2+2*gam1^2*gam2^2*lam1*lam2+2*gam1^2*gam2*lam2^2*n*mu+2*lam1^2*lam2*gam2^2*n*mu+n^2*mu^2*lam1^2*gam2^2+
      lam1^2*lam2^2*gam2^2+lam1^2*lam2^2*gam1^2+gam1^2*gam2^2*lam1^2+gam1^2*gam2^2*lam2^2+2*lam1*lam2^2*gam2^2*gam1+
      n^2*mu^2*lam2^2*gam1^2-2*lam1*lam2^2*gam2*n*mu*gam1+2*lam1^2*lam2*gam2*n*mu*gam1-2*gam1*gam2^2*lam1*n*mu*lam2+
      2*gam1^2*gam2*lam1*n*mu*lam2-2*gam1*gam2^2*lam1^2*n*mu-2*lam1*lam2^2*gam1^2*n*mu-2*n^2*mu^2*lam2*gam1*lam1*gam2)^(1/2))
%%%%%%%%%% Define the basic matrices   %%%%%%%%%%%%%%
T1=[-lam1 (1-p)*lam1;lam2 -lam2];
T10=[p*lam1;0];
T10p=kron(T10,[1,0]);
T2=[-gam1 (1-p)*gam1;gam2 -gam2];
T20=[p*gam1;0];
T20p=kron(T20,[1,0]);
Tp=kron(T10p,eye(2))+kron(eye(2),T20p);% go into server transposed matrix
S1=[-mu];
S10=[mu];%departure server transposed matrix
An=kron(n*mu,eye(4));
Bn=[-lam1-gam1-n*mu,(1-p)*gam1,(1-p)*lam1,0;gam2,-lam1-gam2-n*mu,0,(1-p)*lam1;
     lam2,0,-lam2-gam1-n*mu,(1-p)*gam1;0,lam2,gam2,-lam2-gam2-n*mu];
C=Tp;
F=[kron(T1,eye(2))+kron(eye(2),T2)+C,ones(4,1)];
%%%%%%%%%%%%%% mean arrival rate%%%%%%%%%%%%%%%%%%%%%
lam=(inv((1/lam1)*p + ((1/lam1)+(1/lam2))*(1-p))+inv((1/gam1)*p + ((1/gam1)+(1/gam2))*(1-p)))*p %total arrival rate
%%%%%%%%%%%%%% Compute R matrix %%%%%%%%%%%%%%%%%%%%%
R2=zeros(4,4);
R1=-C*inv(Bn);
i=0;
delta=10;
    while delta > (10^-8)
        R2=-C*inv(Bn)-R1^2*An*inv(Bn);
        delta=norm(R2-R1,inf);
        R1=R2;
        i=i+1;
    end
    i;
    R2;
%%%%%%% Start Matrix geometric method %%%%%%%%%
t1 = cputime;
Qn=[zeros(4*(n+1),4*(n+1)),ones(4*(n+1),1)];% steady state
Qn(4*(n+1)-3:4*(n+1),4*(n+1)-3:4*(n+1)+1)=[Bn+R2*An,inv(eye(4)-R2)*ones(4,1)];
  for k=1:n
Qn(4*(k+1)-3:4*(k+1),4*k-3:4*k)= kron(k*mu,eye(4)) ;
Qn(4*k-3:4*k,4*k-3:4*k)=[-lam1-gam1-(k-1)*mu,(1-p)*gam1,(1-p)*lam1,0;
                         gam2,-lam1-gam2-(k-1)*mu,0,(1-p)*lam1;
                         lam2,0,-lam2-gam1-(k-1)*mu,(1-p)*gam1;
                         0,lam2,gam2,-lam2-gam2-(k-1)*mu] ;
Qn(4*k-3:4*k,4*(k+1)-3:4*(k+1))=C ;
```

```
  end
  Qn;
sol = [zeros(1,(n+1)*4),1] / Qn % Use the matrix geometric procedure to solve pi_0 pi_1 pi_2... pi_n
sum(sol) %The queue empty rate by the matrix geometric
t2 = cputime;
t2-t1% Cpu time of the matrix geometric method
%%%%%%%%%%%% Start Ramaswami's formula%%%%%%%%
t3 = cputime;
A01=[An] ;
A0=[Bn];
A1=[C];
B_0=[zeros(4*(n+1),4*(n+1))];
B_0(4*n+1:4*n+4,4*n+1:4*n+4)=[-lam1-gam1-n*mu,(1-p)*gam1,(1-p)*lam1,0;gam2,-lam1-gam2-n*mu,0,(1-p)*lam1;
    lam2,0,-lam2-gam1-n*mu,(1-p)*gam1;0,lam2,gam2,-lam2-gam2-n*mu];
for k=1:n
B_0(4*(k+1)-3:4*(k+1),4*k-3:4*k)= kron(k*mu,eye(4)) ;
B_0(4*k-3:4*k,4*k-3:4*k)=[-lam1-gam1-(k-1)*mu,(1-p)*gam1,(1-p)*lam1,0;
                          gam2,-lam1-gam2-(k-1)*mu,0,(1-p)*lam1;
                          lam2,0,-lam2-gam1-(k-1)*mu,(1-p)*gam1;
                          0,lam2,gam2,-lam2-gam2-(k-1)*mu] ;
B_0(4*k-3:4*k,4*(k+1)-3:4*(k+1))=C ;
end
B_0;
B1=[zeros(4*n,4);C];
B01=[zeros(4,4*n),An];

R3=zeros(4,4);
R1=-C*inv(Bn);
i=0;
delta=10;
while delta > (10^-8)
R3=-An*inv(Bn)-R1^2*C*inv(Bn);
delta=norm(R3-R1,inf);
R1=R3;
i=i+1;
end
i;
R3;

U0=Bn+C*R3;
P=B_0-B1*inv(U0)*B01;
sol = [zeros(1,4*(n+1)),1] / [P,[ones(4*(n-1),1);inv(eye(4)-R2)*ones(4,1);zeros(4,1)]]% Use Ramaswami's formula to solve pi_0,...pi_n
sum(sol) %The queue empty rate of Ramaswami
t4 = cputime;
t4-t3 % CPU time by Ramaswami
%%%%%%%%%%%%%%% Starting of verifying Matrix with a replaced column %%%%%%%%%%
Qn=[zeros(4*(n+1),4*(n+1))];% steady state
Qn(4*(n+1)-3:4*(n+1),4*(n+1)-3:4*(n+1))=[-lam1-gam1-(n)*mu,(1-p)*gam1,(1-p)*lam1,0;gam2,-lam1-gam2-(n)*mu,0,(1-p)*lam1;
   lam2,0,-lam2-gam1-(n)*mu,(1-p)*gam1;0,lam2,gam2,-lam2-gam2-(n)*mu]+R2*kron(n*mu,eye(4));
for k=1:n
Qn(4*(k+1)-3:4*(k+1),4*k-3:4*k)= kron(k*mu,eye(4)) ;
Qn(4*k-3:4*k,4*k-3:4*k)=[-lam1-gam1-(k-1)*mu,(1-p)*gam1,(1-p)*lam1,0;
                        gam2,-lam1-gam2-(k-1)*mu,0,(1-p)*lam1;
                        lam2,0,-lam2-gam1-(k-1)*mu,(1-p)*gam1;
                         0,lam2,gam2,-lam2-gam2-(k-1)*mu] ;
Qn(4*k-3:4*k,4*(k+1)-3:4*(k+1))=C;
end
Qn;
J1=[ones(4*(n),1);inv(eye(4)-R2)*ones(4,1)];
Z=[J1, Qn( 1:4*(n+1),2:4*(n+1) )];
sol = [1,zeros(1,4*(n+1)-1)] / [Z];%
sum(sol)  %The queue empty rate
%%%%%%%%%%%%%% Start LU factorization%%%%%%%%%%%%%%
```

```
t5 = cputime;
J=(J1)';% Transpose
K=[J1, Qn( 1:4*(n+1),2:4*(n+1) )]';%Use Gaussian elimination
temp=[J];
for i=4*(n+1):-1:9
for  j=i-1:-1:1
temp(j)=temp(j)-(K(i-4,j)/K(i-4,i))*temp(i);
 end
temp(i)=0;
 end
 temp;%Use Gaussian elimination to make first row to be zero
%%%%%%%%%%%% Compute Z_n %%%%%%%%%%%%%%%%%%%%%%%%%%%%
S=[temp;K(2:4*(n+1),1:4*(n+1))];%Use Gaussian elimination
%%%%%%%%%%%%% Algorithm 1: LU factorization %%%%%%%%%%
U(1:4,1:4)=S(1:4,1:4);
    for i = 2:(n+1)
L((i-1)*4+1:(i-1)*4+4,(i-2)*4+1:(i-2)*4+4) = S((i-1)*4+1:(i-1)*4+4,(i-2)*4+1:(i-2)*4+4)*
/(U((i-2)*4+1:(i-2)*4+4,(i-2)*4+1:(i-2)*4+4));
U((i-1)*4+1:(i-1)*4+4,(i-1)*4+1:(i-1)*4+4) = S((i-1)*4+1:(i-1)*4+4,(i-1)*4+1:(i-1)*4+4)-L((i-1)*
4+1:(i-1)*4+4,(i-2)*4+1:(i-2)*4+4)*S((i-2)*4+1:(i-2)*4+4,(i-1)*4+1:(i-1)*4+4);
    end
L;%get L1,L2,...,Ln
U;%get U1,U2,...,Un
%%%%%%%%%%%% Algorithm 2: Forward and backward substitution%%%%%%
y=zeros(4*(n+1),1);
for i=2:(n+1)
y(1:4,1)=[1;0;0;0];
y(4*i-3:4*i,1)=[zeros(4,1)-L((i-1)*4+1:(i-1)*4+4,(i-2)*4+1:(i-2)*4+4)*y(4*(i-1)-3:4*(i-1),1)];
    end
y;%solve y1 y2...yn
x=zeros(4*(n+1),1);
x(4*(n+1)-3:4*(n+1),1)=1/(U(4*(n+1)-3:4*(n+1),4*(n+1)-3:4*(n+1)))*y(4*(n+1)-3:4*(n+1),1);
for i=(n+1):-1:2
x(4*(i-1)-3:4*(i-1),1)=1/(U(4*(i-1)-3:4*(i-1),4*(i-1)-3:4*(i-1))) * [y(4*(i-1)-3:4*(i-1),1)-S(4*(i-1)-3:4*(i-1),4*i-3:4*i)*x(4*i-3:4*i,1)];
    end
x      % Use LU factorization to get pi_0,pi_1,...,pi_n
sum(x) %The queue empty rate by the matrix geometric
t6 = cputime;
t6-t5  %Cpu time of the matrix geometric method
```

Appendix D

List of frequantly used symbols in the thesis

symbol	its mean
n	the number of servers.
$\mathbf{S}_1$	an arrival stays in the server.
$\mathbf{S}_{1o}$	an arrival finishes the service.
μ	the service rate.
λ_1	arrival rate in the first phase of the first stream.
λ_2	arrival rate in the second phase of the first stream.
γ_1	arrival rate in the first phase of the second stream.
γ_2	arrival rate in the second phase of the second stream.
p	the probability of an arrival to the queue.
$\mathbf{B}$	the internal phase changes for the composite arrival process.
$\mathbf{C}$	an arrival goes into the system.
$\mathbf{A}$	an arrival finishes the service, and departures the system.
$\boldsymbol{\theta}$	stationary probability.
$\overline{\lambda}$	total arrival rate.
$\mathbf{T}_m$	correspond to phase transitions.
$\mathbf{T}_{mo}$	correspond to the rate as arrivals enter the systems phase transitions.
$\boldsymbol{\pi}_k$	the vector of probabilities of k customers in the system.
$\mathbf{Q}$	the generator matrix of a continuous time Markovian process.
$\mathbf{R}$	means of the iterative procedure.
$\boldsymbol{\pi}^*$	$\begin{bmatrix} \boldsymbol{\pi}_0 & \boldsymbol{\pi}_1 & \cdots & \boldsymbol{\pi}_{n-1} & \boldsymbol{\pi}_n \end{bmatrix}$
$\mathbf{V}$	an upper triangular matrix.
$\mathbf{W}$	a lower triangular matrix.
$\mathbf{H}$	a submatrix of $\mathbf{W}$.
$\mathbf{V}_0$	a submatrix of $\mathbf{V}$.
$\mathbf{P}$	$\mathbf{B}_0 - \mathbf{B}_1\mathbf{V}_0^{-1}\mathbf{B}_{-1}$.

symbol	its mean
$\mathbf{\Omega}$	a 4×4 matrix which the first row is 1, else 0.
$\mathbf{Z}_n$	a block tridiagonal matrix.
$\mathbf{D}$	permutation matrices.
$\mathbf{E}$	permutation matrices.
$\mathbf{L}$	a lower triangular matrix.
$\mathbf{U}$	an upper triangular matrix.
$\mathbf{F}$	a submatrix of $\mathbf{U}$.
$\mathbf{d}_0$	the departure-point stationary probabilities.
$\mathbf{G}_{n1}$	the generator matrix without departures for the departure process.
$\mathbf{G}_{n2}$	the generator matrix with departures for the departure process.
$\widehat{\mathbf{G}_n}$	$\mathbf{G}_{n1}+\mathbf{G}_{n2}$
t	the probability of departure leaves the system with at least n customers remaining.
lag_k	the lag_k correlation.