國立交通大學

工業工程與管理學系

博士論文

具有第二次可選擇服務、服務者選擇休假之多個服務者排隊分析

Analysis of Multi-server Queues with Second Optional Service and Bernoulli Vacation

研 究 生：巫佳煌

指導教授：彭文理 博士

中 華 民 國 一 零 一 年 六 月

具有第二次可選擇服務、服務者選擇休假之多個服務者排隊分析

Analysis of Multi-server Queues with Second Optional Service and Bernoulli Vacation

研 究 生：巫佳煌　　Student：Chia-Huang Wu

指導教授：彭文理　博士　　Advisor：Dr. W. L. Pearn

國 立 交 通 大 學

工業工程與管理學系

博士論文

A Dissertation Submitted to

Department of Industrial Engineering & Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Philosophy

in

Industrial Engineering & Management

June 2012

Hsinchu, Taiwan, Republic of China

中 華 民 國 一 零 一 年 六 月

具有第二次可選擇服務、服務者選擇休假之多個服務者排隊系統分析

學生：巫佳煌　　指導教授：彭文理　博士

國立交通大學管理學院

工業工程與管理學系

摘 要

本論文主要為多個服務者排隊系統含有第二次可選擇服務、服務者選擇休假以及考量顧客重試行為等條件之分析研究。多服務者排隊模式在實務上較單一服務者模式更有彈性及適用性，以往多服務者排隊系統之數學分析技巧相對較為複雜且困難，而相關文獻也較少。所有抵達系統的顧客都必須接受服務者所提供的第一必要服務，當顧客接受完第一必要服務後，部分的顧客會選擇繼續接受第二種附加服務。所謂服務者選擇休假是指每位服務者在每服務完一位顧客後都有一定的機率會進行(僅)一次的休假，並於休假結束之後回到系統之中繼續提供服務或等待新顧客的到來，即單一次休假策略。當系統中的服務者都處於忙碌時，新到達的顧客將進入循環區(orbit)等待，於一段時間後再嘗試著進入系統之中接受服務，此循環將持續進行直到該顧客接受完服務並離開系統為止，此稱為顧客之重試行為。因循環區之中大多數顧客的嘗試都是失敗的重試行為，並不會造成系統狀態的變化，於是我們假設循環區中允許重試的顧客人數有一最大上限值 N，同時可以簡化數學模式分析上的困難度。我們一共研究了 M/M/c 排隊系統含有第二次可選擇服務(及顧客重試行為)以及 M/M/c 排隊系統含有服務者選擇休假(及顧客重試行為)等四個排隊模式。

對於這四個排隊系統，我們利用矩陣幾何法 (matrix-geometric method) 以及遞迴技巧 (recursive technique) 來推論其系統達穩態之條件及穩態機率解。除此之外，要推論出這四個排隊系統確切的比率矩陣 (closed-form of rate matrix) 是相當困難的，然而在使用矩陣幾何法時，比率矩陣為最重要之元件。在本篇論文裡，我們將利用一單調收斂之數列去求得比率矩陣之近似解，然後利用推導出來的結果去求取穩態機率的近似解。之後建構成本函數來找尋在不同條件設定下的最佳的服務者個數、平均服務率、平均休假率等系統參數，經由直接搜尋法 (direct search method)

及彷牛頓法 (Quasi-Newton method)　我們可以得到近似最佳解以使得成本函數最小。由於排隊系統進行敏感度研究，可以提供系統分析者了解輸入參數對系統影響，因此，我們也將對近似解與最低成本進行敏感度分析，藉此分析來了解系統參數的變動後，對於近似解與最低成本之影響，最後，我們有提供數值結果並討論之。

關鍵字：選擇休假方策，直接搜尋法，第一必要服務，矩陣幾何法，彷牛頓法，比率矩陣，重試，第二次可選擇服務。

Analysis of Multi-server Queues with Second Optional Service and Bernoulli Vacation

Student: Chia-Huang Wu Advisor: Dr. W. L. Pearn

Department of Industrial Engineering and Management,
College of Management, National Chiao Tung University

Abstract

In this dissertation, the optimization investigated multi-server queueing systems with the second optional service (*SOS*) channel, Bernoulli vacation policy, and customer retrial behaviors are investigated. Multi-server vacation models are more flexible and applicable in practice than single server models. For the multiple server queueing models, the mathematical analyses are complicated and difficult; hence there are only a limited number of studies.

All arriving customers need the first essential service (*FES*) provided by the servers. As soon as the *FES* of a customer is completed, a customer may leave the system or opt for the *SOS*. Bernoulli vacation policy means that the server may take one and only one vacation of random length with certain probability at each service completion. As the completion of vacation, the server stays idly for the next new arriving customer or serves the customers waiting in the queue, if any. That is, the single vacation policy. If the customer finding all servers busy always joins the orbit and tries to enter the system for service later. This manner continues until the customer is eventually served then leave the system. This is so-called the customer retrial behaviors. Because most of retrial behaviors of the customers in the orbit are failed without the change of states, we assume that the number of customers who can generate retrial requests is restricted (truncated) to an upper bound value *N*. This setting makes the mathematical model easier to be analyzed.

We investigate four queueing models include the M/M/*c* (retrial) queue with *SOS* channel, the M/M/*c* (retrial) queue with modified Bernoulli single vacation policy, and the M/M/*c* retrial queue with Bernoulli single vacation policy. For those four queueing systems, we develop the stability conditions and steady-state probability solutions by the matrix-geometric method and recursive technique. Furthermore, it is rather difficult to derive the closed-form solution of the rate matrix for those four queueing systems. The rate matrix is the most important component for implementing the matrix-geometric

method to analyze the infinite capacity queueing system. Here, we employ a monotone and convergent sequence to approximate the rate matrix, and obtain the approximation solution of the steady-state probability. The expected cost functions are established to determine the optimal value of the number of servers, mean service rate, mean vacation rate and other system parameters. By implementing the direct search method and Quasi-Newton method, we can find the optimal solution heuristically so that the cost function is minimized. Because of sensitivity investigation on the queueing system with critical input parameters may provide some information for the system analyst. A sensitivity analysis is performed to discuss how the system performances and the optimal solutions are affected by the input parameters in the investigated queueing models. For illustration purpose, numerical results are also presented.

Keywords: Bernoulli vacation policy, direct search method, first essential service, matrix-geometric method, Quasi-Newton method, rate matrix, retrial, second optional service.

誌　　謝

回顧這五年的點點滴滴，心中充滿無限感激，感謝所有參與點綴我博士生求學過程的人們：感謝彭文理老師、唐麗英老師與柯沛程老師對我的指導，讓我在研究及論文撰寫方面獲得許多啟發，受益良多。感謝凱斌學長、榮弘學長與于婷學姐的幫助，讓我學習到許多論文撰寫的技巧與能力。同時，要感謝撥冗參加博士論文口試的靜宜大學 徐世輝老師與交通大學 洪暉智老師。感謝您對本論文的寶貴建言，使得這篇論文更加完善，甚為感激！

感謝俊昇、秉翰、品倫、律瑋、宛倫、佳蕙能陪著我一塊學習工工領域的課程，彼此互相砥礪、切磋琢磨、一同成長。在 MB517 研究室裡，感謝可愛的學弟妹們：振宇、信凱、孟純、君敏以及光陵、彥呈、榮欽、瑜萱的照顧與陪伴，在難過與失意時能有你們的陪伴，讓我倍感溫馨，謝謝你們。同時也感謝陪伴我在課餘時間共同討論課業的晴晴、芋晴、依芳、鈺培、育慧，和妳們一起討論課業使我受益匪淺。因為你們大家，使我的博士求學生涯更加精采與美麗。

最後感謝自始至終在背後支持著我的家人：父 巫樹霖先生、母 吳素味女士以及姐 靜怡、兄 世昌，在這幾年的求學生涯不斷的給我信心與鼓勵。僅以完成此論文，獻給所有關心我及愛我的人們，感謝這一生有你們的參與。

巫佳煌　　謹致

中華民國一〇一年六月十三日

List of Contents

page

List of Tables

page

List of Figures

Chapter 1

Introduction

Queueing system represents an example of a much broader class of interesting dynamic systems. Waiting in line is an exhausting activity in our life. How much time is spent in one's daily activities waiting in some form of a queue: for breakfast; stopped at a traffic light; slowed down on the highways and freeways; delayed at the entrance to one's parking facility; queued for access to an elevator; standing in line for the morning coffee; holding the telephone as it rings, and so on. The list is endless, and too often also are the queues. Therefore, queueing theory is a practical subject and plays an important role in scientific disciplines. In Section 1.1, we describe the background of the queueing theory. Section 1.2 is devoted to introduce theoretical analysis techniques. In Section 1.3, we relate our problem to earlier works in the literature. Section 1.4 shows the description of the queueing models in this thesis. At the end of this chapter, the scope of the thesis is presented in Section 1.5.

1.1 Background

Erlang in 1909, published "The Theory of Probabilities and Telephone Conversations" who was also responsible for the notion of stationary equilibrium. Erlang introduced so-called balance-of-state equations for the first consideration of the optimization of a queueing system. Many valuable applications of the queueing theory such as traffic flow, scheduling, and facility design are well documented in the literatures. Queueing theory originated as a very practical subject that has largely arisen since the close of World War II. The development of the practice of queueing theory must not be restricted by a lack of closed-form solutions, and problem solvers must be able to put the developed theory to good use.

A queueing system can be described as customers arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. Queueing theory was developed to provide models to predict the behavior of systems that attempt to provide service for randomly arising demands. Mathematically, queueing theory deals with the consequence of two basic types of random processes, called arrival processes and service time processes, as they interact under various assumptions concerning the structure of the waiting system. For a queueing processes, six characteristics provide an adequate description : (1) arrival pattern of customers, (2) service pattern of servers, (3)queue discipline, (4) system

capacity, (5) number of service channels, and (6) number of service stages.

The process of arrivals is stochastic, so it is necessary to know the customer arrival process (batch or bulk arrivals) and the reaction of a customer upon entering the system (balking, reneged, or retrial). More importantly, a probability distribution is needed to describe the sequence of customer service times. The service process may depend on the number of customers waiting for service which is so-called state-dependent service. In general, customers arrive and depart at irregular intervals; hence queue length will assume no definitive pattern unless arrivals and service are deterministic. Thus it follows that a probability distribution for queue lengths will be the result of two separate processes - arrivals and services.

The most common discipline that can be observed in everyday life is first come, first served (FCFS). Another discipline as last come, first served (LCFS) is applicable to many inventory systems. Other priority disciplines as preemptive and non-preemptive case can also be implemented in various situations. Usually, system capacity is assumed infinite. A customer is forced to balk if the system capacity is limited and full. Number of service channels means the number of servers in the system. A multi-channel queueing system may have a single queue or allows a queue for each channel (multiple queues). In a multi-stage queueing system, a customer may requests several stages of services (optional service) or has feedback behavior.

1.2 Literature Review

Recently, there have been more studies to multi-server queueing models are investigated because queues with multiple servers are more flexible and applicable in practice than single server models. There are numerous literatures that deal with the system characterization and optimization problem on the queues with second optional service channel, vacation policy, or customer retrial behavior. Queueing models with server vacations are effective tools for performance analysis of manufacturing systems, local area networks, and data communication systems. Excellent surveys on the single server vacation models have been reported by Doshi [29], Takagi [54] and Ke *et al.* [36]. The variations and extensions of these vacation models were developed by several researchers such as Lee *et al.* [42, 43], Choudhury [15, 16], Ke and Chu [34] and many others.

1. queues with SOS channel

A pioneering work in the queue with *SOS* channel was proposed by Madhi [48]

who first introduced the concept of a second optional service. It is assumed that all customers need the first essential service but a part of them may requests the second optional service at the first essential service completion. Madan [47] studied an M/G/1 queue with a second optional service using the supplementary variable technique, in which he considered a general service time distribution for the *FES* service and an exponential service time distribution for the *SOS*. Madan [47] also cited some important applications of this model in many real-life situations. Later, the above model with general service time distribution was discussed by Al-Jararha and Madan [2]. Choudhury and Madan [22] and Choudhury and Paul [23] studied the queue size distribution at a random epoch as well as at a departure epoch for an $M^{[x]}/G/1$ queueing system with a *SOS* channel and different considerations under *N*-policy. The reliability measures were examined by Wang [59] for the ordinary M/G/1 queue with channel breakdowns and *SOS*. Ke [32] investigated a batch arrival $M^{[x]}/G/1$ queueing system with *J* optional services. Choudhury and Tadj [24] generalized this type of model by introducing the concept of a server breakdown and a delay-repair-period. More studied results can be surveyed in Choudhury and Tadj [25], Choudhury *et al*. [26], Choudhury and Deka [19], Ke *et al*. [35], Wang and Li [60], Wang *et al*. [61], Wu *et al*. [63], and Yang *et al*. [64].

2. queues with Bernoulli vacation policy

The M/M/*c* queue with servers' vacations was introduced by Levy and Yechiali [44]. Keilson and Servi [37] firstly investigated an oscillating random walk models for GI/G/1 vacation system with Bernoulli schedules. Bernoulli vacation means that when the service of a customer is completed, the server may leave for a vacation of random interval with probability p or to serve the next customer with probability $1-p$ (Choudhury and Madan [21, 22]). A numbers of papers (Tadj *et al*. [53], Madan *et al*. [46], Choudhury [17, 18]) have appeared in queueing literature in which the server provides to each heterogeneous service with Bernoulli schedule vacation (BSV). Sherman and Kharoufeh [51] developed the optimal Bernoulli routing in an unreliable M/G/1 retrial queue. They showed that the system exhibits a dual stability structure and characterized the optimal Bernoulli routing policy.

3. retrial queues

Review of retrial queue literature could be found in Yang and Templetion [65], Falin and Templeton [30] and Artalejo [4]. Retrial queueing system is characterized by the feature that the arriving customers who on encountering the busy server will join a retrial queue called orbit when all servers are busy and unavailable. An arbitrary

customer in the orbit generates a stream of repeated requests that is independent of the rest of customers in the orbit. This situation arises in telephone switching systems, telecommunication networks and computer systems. A number of applications of retrial queues in science and engineering can be found in Kulkarni and Liang [39] Many interesting studies have been devoted to an approximate approach to the stationary probabilities for system states (Artalejo and Choudhury [6], Bright and Taylor [12], Stepanov [52], Breuer *et al*. [11] and Chakravarthy and Dudin [13]). Gomez-Corral [31] gave a detailed bibliographical guide to the analysis of retrial queues through matrix analytic techniques. Amador and Artalejo [3] refer to a busy period and present a detailed computational analysis of four new performance measures: the successful retrials, the blocked retrials, the successful primary arrivals, and the blocked primary arrivals. Kim *et al*. [38] studied the BMAP/PH/N retrial queueing system operating in Markovian random environment. The main performance measures of the system were derived and some numerical example illustrations were presented. Then, the finite source MAP/PH/N retrial G-queue operating in a random environment was investigated by Wu *et al*. [62]. Formulae for important performance measures are derived. These results can model the Ethernet system appropriately.

The monotonicity properties of an unreliable M/G.1 retrial queue was investigated by Taleb and Aissani [55] by using the general theory of stochastic ordering. An analysis of the energetic version of retrial $M^{[x]}$/G/1 queue with vacation under quite general assumptions about parametric distributions was provided by Aissani [1]. The computation and optimization problem of a multi-server retrial queue with geometric loss and feedback was investigated by Lin and Ke [45]. For an M/M/*c* retrial queue with PH distribution of retrial time, Yang and Dug [66] presented an approximation which have some different features from the previous literature and can be useful for more complicated queueing system.

It is worth noting that the truncation models seem to be the most convenient method for obtaining reliable numerical solutions for the M/M/*c* retrial queue. Neuts and Rao [50] and Artalejo and Pozo [7] proposed several models in this direction and provided efficient approximate solutions to the stationary distribution of the M/M/*c* retrial queue. Artalejo *et al*. [8-10] presented an algorithmic analysis of the maximum number of customers in orbit (and in the system) during a busy period. Artalejo [5] presented a bibliography on retrial queues made during the past decade 2000-2009. Tien and Ram [58] provided an efficient method to compute the rate matrix for retrial queues with large number of servers using characteristic matrix polynomial technique. Furthermore, Tien [56, 57] also presented new and efficient computation algorithms

for the multi-server retrial queues with various conditions.

Recently, the queueing retrial models with *SOS* channel or and various vacation policies are discussed. Choudhury [17, 18] investigated the M/G/1 and $M^{[x]}$/G/1 queue with two phases of heterogeneous service and Bernoulli vacation schedule which operate under various retrial policies. Choudhury and Deka [19] dealt with the steady-state behavior of $M^{[x]}$/G/1 retrial queue with second optional service, unreliable srever and Bernoulli admission mechanism. Furthermore, Ke and Chang [33] derived the mathematical model of $M^{[x]}/(G_1,G_2)/1$ retrial queue under Bernoulli vacation schedules with general repeated attempts and starting failures. Later, Langaris and Dimitriou [40] investigated a single-server queueing with n-phases of service and $(n-1)$ types of retrial customers. Any conditions mentioned earlier can be considered to be assumptions of a queueing system. Choudhury *et al*. [27] investigated an $M^{[x]}$/G/1 queue with two phase service and Bernoulli vacation schedule under multiple vacation policy. Lately, Dimitriou and Langaris [28] discussed a repairable queueing model with two-phase service, start-up times and retrial customers.

Existing works with optional service or Bernoulli vacation policy, including those above, mainly focused on single-server queue. Therefore, in this thesis, we deal with four queueing models with various considerations. The first two are M/M/c queue with second optional service channel and M/M/c queue with *SOS* channel and customer retrial behavior. Then, an M/M/c queue with modified Bernoulli single vacation (BSV) policy is considered. Finally, an M/M/c/BSV retrial queue is investigated.

1.3 Theoretical Analysis Technique

In this section, we introduce two theoretical analysis techniques: matrix-geometric method and Quasi-Newton method. Furthermore, some methods implemented in calculations and computations are also presented in detail.

1.3.1. Matrix-geometric method

Neuts [49] introduced the matrix-geometric method which establish a transition matrix whose entries become matrices. For a quasi-birth-death (QBD) process, the infinitesimal generator matrix $\mathbf{Q}$ can be rewritten in a block-matrix form with tri-diagonal structure. After formulating the $\mathbf{Q}$ values for a specific problem, the steady-state solution can be determined analytically via the equation $\mathbf{\Pi Q} = \mathbf{0}$ where $\mathbf{\Pi}$ denotes the steady-state probability vector. The QBD process describes a

generalization of the birth-death process. As with the birth-death process movements between it moves skip free up and down. For an infinite capacity queueing system, a matrix $\mathbf{R}$, called rate matrix, is an important component as deriving the steady-state probabilities recursively. The rate matrix is the nonnegative solution of a matrix-quadratic equation in the following

$$\mathbf{R}^2\mathbf{C}_c + \mathbf{R}\mathbf{A}_c + \mathbf{B} = \mathbf{0},$$

where matrices $\mathbf{C}_c$, $\mathbf{A}_c$ and $\mathbf{B}$ are sub-matrices of the infinitesimal generator $\mathbf{Q}$ of the queueing system. In this dissertation, we will use the property as follow to get steady-state solution

$$\mathbf{\Pi}_{i+1} = \mathbf{\Pi}_i\mathbf{R}, \quad i \geq 0.$$

As a result, the developing of rate matrix is a significant object in the investigation of a queue with infinite capacity. The rate matrix $\mathbf{R}$ can be obtained explicitly in close-form by using recursive technique via computer software. When the solution of $\mathbf{R}$ becomes more complex and difficult to be obtained, Neuts [49] provided some algorithms such as linear progression algorithm and sequence convergence algorithm to approximate the rate matrix $\mathbf{R}$.

1.3.2. Quasi-Newton method

Constantly, the analytic study of the optimization problem will be an arduous task because of the high complexity. Therefore, some heuristic algorithms to obtain the approximate solution are included. In this dissertation, the Quasi-Newton method is employed to find the heuristic solution of the optimization problem with continuous decision variables. It is noted that the derivative of the object function with respect to input parameters indicates the direction which the object function increases. That is, the better (optimal) solution can be found along the opposite direction of the gradient. (see Chong and Zak [14]). The procedures of Quasi-Newton method are described as below:

Algorithm : Quasi-Newton Method

Step 1 Set a initial trial solution $\mathbf{x}^{(0)}$ for object function F, and compute $\vec{\nabla}F(\mathbf{x}^{(0)})$.

Step 2 While the norm of gradient $\left|\vec{\nabla}F(\mathbf{x}^{(i)})\right| > \varepsilon$ (tolerance) do Steps 3-4.

Step 3 Compute the cost Hessian matrix at point $\mathbf{x}^{(i)}$ denoted by $\mathbf{H}(\mathbf{x}^{(i)})$.

Step 4 Find the new trial solution $\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} - [\mathbf{H}(\mathbf{x}^{(i)})]^{-1}\vec{\nabla}F(\mathbf{x}^{(i)})$.

Step 5 Compute $\vec{\nabla}F(\mathbf{x}^{(i+1)})$ and back to Step 2.

Comparison with gradient methods, Quasi-Newton method use the second derivative (Hessian) and it has order of convergence at least 2 (see Chone and Zak [14]). That is, for quadratic function, Quasi-Newton method converges in one step. However, it may not have descent property even though it may diverge if the initial trial solution does not start close the optimal solution sufficiently. Moreover, the heuristic solution found by the Quasi-Newton method may be local optimal solution rather than global optimal solution.

1.4 Problem Statement

In this dissertation, we investigate the optimal problem of an M/M/*c* (retrial) queue with second optional service (*SOS*) channel or Bernoulli single vacation policy. In day to day life, one encounters numerous examples of queueing models where all arriving customers need an essential service but only some require an additional optional service. For example, a manufacturing industrial system for a pump that manufactures different kinds of pumps which require shafts of various dimensions is considered. The arrival of shafts from the turning center to the computer numerical control (CNC) copy turning center follows a random process, which the center owns multiple CNC machines. The mechanics set up the template in these CNC machines to perform the copy turning process shafts (*i.e.*, the first essential service). The good quality shafts items are kept in the storage and are sold. Some of the processed (served) shafts are defective and need to be rework (re-served) to meet the required specification (*i.e.*, the second optional service). Furthermore, in reality, the customers do not always waiting in the queue but retry to enter the system later when the system is full-loading. This is so-called the customer retrial behavior. In addition, the server may take a vacation at each service completion. For example, consider a production process with a number of machines. It may so happen that the production process either needs to be temporarily stopped for overhauling and maintenance of the system after each service completion or continue the service for the next unit/customer in the queue. Hence, the servers may take a vacation with certain probability which is called Bernoulli vacation policy.

We assume that arrivals of customers follow a Poisson process with rate λ. There are c servers provide service to all arriving customers for *FES*. Service times of *FRS* channel are independent and identically distributed (*i.i.d.*) random variables obeying a exponential distribution function with service rate μ_1. As soon as *FES* of a customer is completed, a customer may leave the system with probability $1-\theta$ or

may opt for *SOS* with probability θ $(0 \le \theta \le 1)$, at the completion of which the customer departs from the system and the next customer, if any, from the queue is taken up for his *FES*. Service times of *SOS* channel are *i.i.d.* random variables having a exponential distribution with service rate μ_2. When an arriving customer finding all servers are busy will joins the orbit and make repeated attempts in random intervals having length exponentially distribution with retrial rate σ. This manner continues until the customer is eventually served. We assume that there exists an upper bound N on the number of customers in the orbit that are allowed to conduct retrials (Neuts and Rao [50], Artalejo and Pozo [7]). The server may take a vacation of random length with probability p or continue to serve the next customer, if any with probability q ($q = 1 - p$). The vacation times are also exponentially distributed.

It is also assumed that arriving customers form a single waiting line based on the FCFS (first-come, first-served) discipline. One server can serve one only and only one customer at a time. The service process, the arrival process and the vacation process are eventually independent. A customer who arrives and finds the server busy or on vacation must wait in the queue until a server is available. At the vacation completion, the server backs to server the customers waiting in the queue or stays idly in the system. That is, single vacation policy.

1.5 Scope of Dissertation

The main purposes of this dissertation are to analyze: (i) the M/M/*c* (retrial) queue with second optional service channel; and (ii) the M/M/*c* (retrial) queue with Bernoulli single vacation policy. This dissertation is organized by six chapters as follows:

Chapter 1 is an introduction, which introduces the background of the queueing theory. Some earlier studies and literatures on the multi-server queue with retrial behaviors and vacation policy are included. Several techniques and methods relevant to this study are presented.

In Chapter 2, we study the optimization of the M/M/*c* queue with second optional service channel. The matrix-geometric method is employed to derive the steady-state probability vector. One algorithm to obtain the approximate rate matrix is provided. The exact and explicit expressions of some important system performances are given by using the matrix-analytical method. Next, the expected cost function per unit time is constructed by the system performances. We determine the optimal number of

servers and the optimal service rates to minimize the expected cost per unit time. In addition, a sensitivity analysis is also investigated. Finally, some numerical results are provided to illustrate the optimization procedures.

In Chapter 3, we more consider the retrial behavior of the customer then extend the queueing system investigated in Chapter 2 into an M/M/*c* retrial queue with *SOS* channel. The arriving customer joins the orbit and retries to enter the system for service later. The entries of state-transition matrix are listed explicitly. An algorithm is provided to solve the steady-state equation system recursively. The expresses of the system performance are given. The effect of the system parameters on the system performances is studied. Some numerical examples and graphs are presented.

In Chapter 4, we consider an M/M/*c* queue with a modified Bernoulli single vacation policy. Under Bernoulli vacation policy, the server may take a vacation at the service completion of a customer with a certain probability. Particularly, we modify the tradition Bernoulli vacation that the vacation may occur only when the server is idle after service completion. At the vacation completion, the server serves the customers waiting in the queue or stays idle in the system, that is, single vacation policy. The closed-form expression of the rate matrix is derived explicitly. Some results about the special case of single server are provided. For this queueing system, the optimal number of servers, the optimal service rate and the optimal vacation rate are investigated numerically.

In Chapter 5, the customer retrial behavior is included in the model considered in Chapter 4. Similarly, the stability condition, the rate matrix, and the steady-state probability are derived by using matrix-analytical technique. The system performance expressions are also presented. The optimal number of servers, the optimal vacation rate, and the optimal service rate are determined to minimize the expected cost per unit time.

Chapter 6 presents some conclusions based on results of the investigation, and recommendations for the future investigations.

Chapter 2

M/M/*c* Queueing System with Second Optional Service Channel

In day to day life, second optional services are very commonly observed in some queueing system (see Madan [47]). For example, all customers come to shops which sell coffee beans will buy coffee beans or grains but only some of them want to utilize a grinding facility service. All ships arriving at a port may need unloading service on arrival but only some of them may require re-loading service soon after the unloading. All cars arriving at a gas station need gas refueling but only some of them require a car wash services after the refueling.

In this chapter, we study the optimization of the multi-server queueing system with *SOS*. All arriving customers arrive to demand the *FES*. After the completion of the *FES*, a customer may leave the system with probability $(1-\theta)$ or may instantly go for a *SOS* with probability θ ($\theta \in [0,1]$). The customers arrive according to a Poisson process. Service times of the *FES* and *SOS* channels are assumed to be exponentially distributed. There are c channels (servers) that provide the first essential service as well as the second optional service to arriving customers. Each channel can serve only one customer and provides only one of essential service or second optional service at a time.

This chapter is organized as follows: In Section 2.1, we give some basic assumptions of the queue under study and give some notations. Section 2.2, the steady-state equations are obtained and represented in matrix form. In Section 2.3, the stability condition is derived. An algorithm to find the rate matrix is provided. In Section 2.4, the stationary probabilities are gained by implementing a recursive procedure. In Section 2.5, some explicit expressions of important system performance measures are derived. Finally, numerical results are given in Section 2.6.

2.1 Assumptions and Notations

We assume that arrivals of customers follow a Poisson process with rate λ. A single server is needed to serve all arriving customers for the *FES*. The service times of the *FES* channel are independent and identically distributed (*i.i.d*) random variables obeying an exponential distributions with mean $1/\mu_1$. As soon as the *FES* of a customer is completed, a customer may leave the system with probability $1-\theta$ or

opt for a *SOS* provided by the same server with probability θ ($\theta \in [0,1]$), at the completion of which the customer departs from the system and the next customer, if any, from the queue is taken up for his *FES*. The service times of the *SOS* channel are another independent and identically distributed (*i.i.d*) random variables having an exponential distributions with mean $1/\mu_2$. Furthermore, the same server is assumed to serve both service channels. Customers who upon entry into the channel facility, find that all channels are busy have to wait in the queue until a channel becomes available. It is also assumed that arriving customers form a single waiting line based on the FCFS (first-come, first-served) discipline. Various stochastic processes involved in the system are assumed to be independent of each other. We will represent this queue as the M/M/*c* with *SOS* channel, where the first symbol denotes the inter-arrival time distribution for customer, the second symbol denotes service time distributions for both *FRS* and *SOS* channels, and the third symbol denotes number of channels that providing services.

In this chapter, the following notations and probabilities are used.

λ — mean arrival rate

μ_1 — mean service rate of *FES* channel

μ_2 — mean service rate of *SOS* channel

θ — probability that a customer may opt for the *SOS*

c — number of channels (servers)

$\mathbf{\Pi}$ — steady-state probability vector

$\mathbf{Q}$ — infinitesimal generator

$\mathbf{I}$ — identity matrix

$\mathbf{e}$ — identity column vector (a column vector with all elements equal to 1)

$\mathbf{F}$ — irreducible generator

$\mathbf{x}$ — invariant probability

$\mathbf{R}$ — rate matrix

L_1 — expected number of customers in the *FES* channel

L_2 — expected number of customers in the *SOS* channel

$E[I]$ — expected number of idle servers

$E[B]$ — expected number of busy servers

L_s — expected number of customers in the system

F — cost function

2.2 M/M/*c* Queue with *SOS* channel

For an infinite capacity M/M/*c* queueing system with *SOS* channel. The state of the system are described by the pair (i, j), $i = 0,1,2,...$ and $j = 0,1,2,...,c$, where i and j denote the number of customers in the *FES* and *SOS* channels, respectively. If $(i + j) \le c$, the customers upon to the server will get service immediately. Otherwise ($(i + j) > c$), the new arriving customer must wait in the queue until a server becomes available. In steady-state, we define the following notations:

$P_{i,j} \equiv$ probability that there are i customers in the *FES* channel and there are j customers in the *SOS* channel, where $i = 0,1,2,...$ and $j = 0,1,2,...,c$.

Referring to the state-transition-rate diagram shown in Figure 2.1 and using the birth-and-death process, the steady-state equations governing the queueing system are

(i) $j = 0$

$$\lambda P_{0,0} = (1-\theta)\mu_1 P_{1,0} + \mu_2 P_{0,1}, \tag{2.1}$$

$$(\lambda + i\mu_1)P_{i,0} = \lambda P_{i-1,0} + (i+1)(1-\theta)\mu_1 P_{i+1,0} + \mu_2 P_{i,1}, \quad 1 \le i \le c-1, \tag{2.2}$$

$$(\lambda + c\mu_1)P_{i,0} = \lambda P_{i-1,0} + c(1-\theta)\mu_1 P_{i+1,0} + \mu_2 P_{i,1}, \quad c \le i. \tag{2.3}$$

(ii) $1 \le j \le c-1$

$$(\lambda + j\mu_2)P_{0,j} = \theta\mu_1 P_{1,j-1} + (1-\theta)\mu_1 P_{1,j} + (j+1)\mu_2 P_{0,j+1}, \tag{2.4}$$

$$\begin{aligned}(\lambda + i\mu_1 + j\mu_2)P_{i,j} &= \lambda P_{i-1,j} + (i+1)\theta\mu_1 P_{i+1,j-1} + (i+1)(1-\theta)\mu_1 P_{i+1,j} \\ &\quad + (j+1)\mu_2 P_{i,j+1}, \quad 1 \le i \le c-j-1,\end{aligned} \tag{2.5}$$

$$\begin{aligned}[\lambda + (c-j)\mu_1 + j\mu_2]P_{i,j} &= \lambda P_{i-1,j} + (c+1-j)\theta\mu_1 P_{i+1,j-1} + (c-j)(1-\theta)\mu_1 P_{i+1,j} \\ &\quad + (j+1)\mu_2 P_{i,j+1}, \quad c-j \le i.\end{aligned} \tag{2.6}$$

(iii) $j = c$

$$(\lambda + c\mu_2)P_{0,c} = \theta\mu_1 P_{1,c-1}, \tag{2.7}$$

$$(\lambda + c\mu_2)P_{i,c} = \lambda P_{i-1,c} + \theta\mu_1 P_{i+1,c-1}, \quad 1 \le i. \tag{2.8}$$

There is no way of solving equations (2.1)-(2.8) in a recursive manner to develop the explicit expressions for the steady-state probabilities $P_{i,j}$, where $i = 0,1,2,...$ and $j = 0,1,2...,c$. Alternatively, the infinitesimal generator **Q** describing the M/M/*c* queueing system with *SOS* channel is of the block-tri-diagonal form:

$$\mathbf{Q} = \begin{array}{c} \\ \ell(0) \\ \ell(1) \\ \ell(2) \\ \vdots \\ \ell(c-1) \\ \ell(c) \\ \ell(c+1) \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccccccc} \ell(0) & \ell(1) & \ell(2) & \cdots & \cdots & \ell(c-1) & \ell(c) & \ell(c+1) & \ell(c+2) & \cdots \end{array} \\ \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & & & & \\ & & \ddots & \ddots & \ddots & & & & & \\ & & & & & \mathbf{C}_{c-1} & \mathbf{A}_{c-1} & \mathbf{B} & & \\ & & & & & & \mathbf{C}_c & \mathbf{A}_c & \mathbf{B} & \\ & & & & & & & \mathbf{C}_c & \mathbf{A}_c & \mathbf{B} \\ & & & & & & & & \ddots & \ddots & \ddots \end{bmatrix} \end{array}. \quad (2.9)$$

Each entry of the matrix $\mathbf{Q}$ is a square matrix of order $c+1$ listed as follows:

$$\mathbf{B} = \lambda \mathbf{I}, \quad (2.10)$$

$$\mathbf{A}_i = \begin{bmatrix} a_{i,0} & & & & \\ \mu_2 & a_{i,1} & & & \\ & 2\mu_2 & a_{i,2} & & \\ & & \ddots & \ddots & \\ & & & c\mu_2 & a_{i,c} \end{bmatrix}, \ i = 0, \ldots, c, \quad (2.11)$$

$$\mathbf{C}_i = \begin{bmatrix} c_{i,0} & d_{i,0} & & & \\ & c_{i,1} & d_{i,1} & & \\ & & c_{i,2} & d_{i,2} & \\ & & & \ddots & \ddots \\ & & & & c_{i,c-1} \\ & & & & 0 \end{bmatrix}, \ i = 1, \ldots, c, \quad (2.12)$$

where $\mathbf{I}$ is the identity matrix of order $c+1$, and

$$a_{i,j} = \begin{cases} -(\lambda + i\mu_1 + j\mu_2), & 1 \le i + j \le c, \\ -[\lambda + (c-j)\mu_1 + j\mu_2], & i + j > c. \end{cases} \quad (2.13)$$

$$c_{i,j} = \begin{cases} i(1-\theta)\mu_1, & 1 \le i + j \le c, \\ (c-j)(1-\theta)\mu_1, & i + j > c. \end{cases} \quad (2.14)$$

$$d_{i,j} = \begin{cases} i\theta\mu_1, & 1 \le i + j \le c, \\ (c-j)\theta\mu_1, & i + j > c. \end{cases} \quad (2.15)$$

Consequently, the steady-state equations (2.1)-(2.8) can be represented in matrix form using the above matrices. The stationary probability vector of $\mathbf{Q}$ is denoted by symbol $\mathbf{\Pi} = [P_0, P_1, P_2, ..., P_{c-1}, P_c, P_{c+1}, ...]$ where $P_i = [P_{i,0}, P_{i,1}, ..., P_{i,c}]$ is a row vector with dimension $c+1$.

2.3 Matrix-geometric Property

Before handling the steady-state equation system, the stability of the queueing system should be confirmed. It implies that the unique solution of the steady-state equation system $\mathbf{\Pi Q} = \mathbf{0}$ exists. Next, we would derive the sufficient and necessary stability condition.

2.3.1. Stability condition

Let

$$\mathbf{F} = \mathbf{C}_c + \mathbf{A}_c + \mathbf{B}, \tag{2.16}$$

is an irreducible generator. $\mathbf{x} = [x_0, x_1, ..., x_c]$ is the invariant probability of $\mathbf{F}$. Then $\mathbf{x}$ satisfies the two conditions

$$\mathbf{xF} = \mathbf{0} \quad \text{and} \quad \mathbf{xe} = 1, \tag{2.17}$$

where $\mathbf{e}$ is a column vector with dimension $c+1$ and all its elements equal to one. Expand $\mathbf{xF} = \mathbf{0}$ implies

$$c\theta\mu_1 x_0 = x_1\mu_2, \tag{2.18a}$$

$$-(c-i+1)\theta\mu_1 x_{i-1} + [(c-i)\theta\mu_1 + i\mu_2]x_i - (i+1)\mu_2 x_{i+1} = 0, \quad 1 \le i \le c-1, \tag{2.18b}$$

$$\theta\mu_1 x_{c-1} = c\mu_2 x_c. \tag{2.18c}$$

Equation (2.18) implies that $x_1 = c\theta\mu_1 / \mu_2 x_0$, and the following recursive equation

$$x_{i+1} = \frac{(c-i)\theta\mu_1}{(i+1)\mu_2} x_i, \ i = 1, ..., c-1. \tag{2.18d}$$

Then, we have

$$x_{i+1} = \frac{(c-i)\theta\mu_1}{(i+1)\mu_2} x_i = \binom{c}{i+1} \left(\frac{\theta\mu_1}{\mu_2}\right)^{i+1} x_0, \ i = 1, ..., c-1. \tag{2.18e}$$

Also using the condition $x_0 + x_1 + ... + x_c = 1$, the probability x_0 is determined as

$$x_0 = \left[\sum_{i=0}^{c} \binom{c}{i} \left(\frac{\theta\mu_1}{\mu_2} \right)^i \right]^{-1} = \left(1 + \frac{\theta\mu_1}{\mu_2} \right)^{-c}, \tag{2.18f}$$

By Theorem 3.1.1 in Neuts [49], the sufficient and necessary stability condition is

$$\mathbf{xBe} < \mathbf{xC}_c\mathbf{e}, \tag{2.19}$$

Substituting $\mathbf{B}$ and $\mathbf{C}_c$ into equation (2.19) and using (2.18f) to get

$$\mu_1(c - L_2) > \lambda, \tag{2.20a}$$

which is equivalent to

$$\frac{\lambda}{\mu_1(c - L_2)} < 1, \tag{2.20b}$$

where

$$\begin{aligned} L_2 &= x_1 + 2x_2 + \ldots + cx_R \\ &= \sum_{i=1}^{c} i x_i = \sum_{i=1}^{c} i \binom{c}{i} \left(\frac{\theta\mu_1}{\mu_2} \right)^i x_0 = \frac{c\theta\mu_1}{\mu_2} x_0 = \frac{c\theta\mu_1}{\mu_2} \left(1 + \frac{\theta\mu_1}{\mu_2} \right)^{-c}, \end{aligned} \tag{2.21}$$

denotes the expected number of customers in the *SOS* channel. Note that $\theta = 0$ or $\mu_2 \to \infty$ (*i.e.*, L_2=0), equation (2.21) can be reduced to the stability condition for the ordinary M/M/*c* queueing system without *SOS* channel.

2.3.2. Linear progression algorithm

When the stability condition is satisfied, the steady-state equation system $\mathbf{\Pi Q} = \mathbf{0}$ has a unique solution. Our aim is to obtain the steady-state vector $\mathbf{\Pi}$ by means of the matrix analytic method and normalization. By applying the matrix geometric method, the steady-state probabilities $[P_{c+1}, P_{c+2}, P_{c+3}, \ldots]$ can be obtained as $P_i = P_c \mathbf{R}^{i-c}$, $i \geq c+1$, where $\mathbf{R}$ is the minimal nonnegative solution to the matrix quadratic equation

$$\mathbf{R}^2\mathbf{C}_c + \mathbf{R}\mathbf{A}_c + \mathbf{B} = \mathbf{0}. \tag{2.22}$$

The matrix $\mathbf{R}$ is a very important matrix needed in the evaluation of the performance measures of a QBD process. It is known as the rate matrix of the Markov chain $\mathbf{Q}$. Developing a closed-form solution for the rate matrix by taking the nonlinear equation (2.22) is very difficult because the matrix structure of $\mathbf{A}_c$, $\mathbf{B}$, and $\mathbf{C}_c$ is not consistent. In the following, we will develop some matrix analytic properties to approximate the rate matrix $\mathbf{R}$.

Let us decompose the level space into two groups as $\ell(J)=\{\ell(0),\ell(1),\ldots,\ell(c)\}$ and $\ell(K)=\{\ell(c+1),\ell(c+2),\ldots\}$. The QBD model of this thesis is partially level-dependent up to a certain level (group $\ell(J)$) and thereafter becomes a infinite level-independent (group $\ell(K)$). It is well-known that an infinite level-independent QBD has the matrix-geometric form which can be solved from the matrix quadratic equation (Latouche and Ramaswami [41]). The level-independent structure in our thesis can be solved by Cramer's rule. Thus, we can use the finite level-dependent algorithm first and then the algorithm of infinite level-independent QBDs to derive the state probabilities.

It is note from the matrix (9) that starting from level $\ell(c)$ the matrices $\mathbf{C}_{c-1}$ and $\mathbf{A}_{c-1}$ change to $\mathbf{C}_c$ and $\mathbf{A}_c$, respectively, which implies that the process holds an infinite level-independent QBD with group $\ell(K)$. First, we reduce the QBD-$\mathbf{Q}$ into a finite level-dependent QBD-$\mathbf{Q}^*$ as :

$$\mathbf{Q}^* = \begin{array}{c} \\ \ell(0) \\ \ell(1) \\ \ell(2) \\ \vdots \\ \ell(c) \\ \ell(c+1) \end{array} \begin{array}{c} \begin{array}{cccccc} \ell(0) & \ell(1) & \cdots & \cdots & \ell(c) & \ell(c+1) \end{array} \\ \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & 0 & \cdots & 0 & 0 \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & \cdots & 0 & 0 \\ 0 & \mathbf{C}_2 & \mathbf{A}_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{A}_c & \mathbf{B} \\ 0 & 0 & 0 & \cdots & \mathbf{C}_c & \mathbf{H} \end{bmatrix} \end{array}. \tag{2.23}$$

From Neuts [49], the matrix $\mathbf{H}$ in (2.23) represents the transitions between the states belonging to the imaginary level group $\ell(K)$. The boundary steady-state probability vector P_{c+1} based on $\ell(c+1)$ is given by solving the following equations

$$P_c\mathbf{B}+P_{c+1}\mathbf{H}=P_{c+1}, \quad \text{(from QBD-}\mathbf{Q}^*\text{)} \tag{2.24a}$$

$$P_c\mathbf{B}+P_{c+1}\mathbf{A}_c+P_{c+2}\mathbf{C}_c=P_{c+1}. \quad \text{(from QBD-}\mathbf{Q}\text{)} \tag{2.24b}$$

Solving equations (2.24), we obtain

$$\mathbf{H}=\mathbf{A}_c+\mathbf{R}\mathbf{C}_c. \tag{2.25}$$

Substituting (2.25) into equation (2.23), it yields the following system of linear equation

$$\mathbf{\Lambda Q}^{*} = [P_0, P_1, P_2, ..., P_{c+1}] \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & 0 & \cdots & 0 & 0 \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & \cdots & 0 & 0 \\ 0 & \mathbf{C}_2 & \mathbf{A}_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{A}_c & \mathbf{B} \\ 0 & 0 & 0 & \cdots & \mathbf{C}_c & \mathbf{H} \end{bmatrix} = \mathbf{0}, \qquad (2.26)$$

where $P_i = [P_{i,0}, P_{i,1}, P_{i,2}, ..., P_{i,c}]$, $i = 0,1,2,...,c+1$. By the arguments of Latouche and Ramaswami [41], there exists an infinitesimal generator $\mathbf{U}$ of the transient continuous-time Markov chain that is restricted to level $\ell(c+2)$ before it reaches $\ell(c+1)$ from group level $\ell(c)$. It is given by

$$\mathbf{U} = \mathbf{A}_c + \mathbf{B}(-\mathbf{U})^{-1}\mathbf{C}_c = \mathbf{A}_c + \mathbf{BG} = \mathbf{A}_c + \mathbf{RC}_c = \mathbf{H},$$

where

$$\mathbf{R} = \mathbf{B}(-\mathbf{U})^{-1}, \quad \mathbf{G} = (-\mathbf{U})^{-1}\mathbf{C}_c.$$

Based on the analysis above, we summarize an algorithm to obtain the approximation for the rate matrix $\mathbf{R}$. (see Latouche and Ramaswami [41]).

Algorithm: Linear Progression Algorithm

Step 1 $\mathbf{G} = (-\mathbf{A}_c)^{-1}\mathbf{C}_c$.

Step 2 while $\|\mathbf{e} - \mathbf{Ge}\| \geq \delta$ (tolerance) do Steps 3-4.

Step 3 set $\mathbf{H} = \mathbf{A}_c + \mathbf{BG}$.

Step 4 set $\mathbf{G} = (-\mathbf{H})^{-1}\mathbf{C}_c$.

Step 5 Assign $\mathbf{R} = \mathbf{B}(-\mathbf{H})^{-1}$.

2.4 Probability Computation

By solving equation (2.24) recursively, the recursive relationship between steady-state probability vectors is given as below:

$$P_0 = P_1\mathbf{C}_1(-\mathbf{A}_0)^{-1} = P_1\phi_1, \tag{2.27a}$$
$$P_1 = P_2\mathbf{C}_2[-(\phi_1\mathbf{B}+\mathbf{A}_1)]^{-1} = P_2\phi_2, \tag{2.27b}$$
$$P_2 = P_3\mathbf{C}_3[-(\phi_2\mathbf{B}+\mathbf{A}_2)]^{-1} = P_3\phi_3, \tag{2.27c}$$
$$\vdots$$
$$P_{c-1} = P_c\mathbf{C}_c[-(\phi_{c-1}\mathbf{B}+\mathbf{A}_{c-1})]^{-1} = P_c\phi_c, \tag{2.27d}$$
$$P_c = P_{c+1}\mathbf{C}_c[-(\phi_c\mathbf{B}+\mathbf{A}_c)]^{-1} = P_{c+1}\phi_{c+1}, \tag{2.27e}$$
$$P_c\mathbf{R}[\phi_{c+1}\mathbf{B}+\mathbf{H}] = \mathbf{0}. \tag{2.27f}$$

where $\phi_1 = \mathbf{C}_1(-\mathbf{A}_0)^{-1}$, $\phi_2 = \mathbf{C}_2[-(\phi_1\mathbf{B}+\mathbf{A}_1)]^{-1}$, …, $\phi_i = \mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1}$, and $\phi_{c+1} = \mathbf{C}_c[-(\phi_c\mathbf{B}+\mathbf{A}_c)]^{-1}$. Consequently, the levels $P_i\,(0 \le i \le c-1)$ state probabilities of equation (2.27) can be written in terms of P_c as $P_0 = P_c\prod_{i=c}^{1}\phi_i$, $P_1 = P_c\prod_{i=c}^{2}\phi_i$, …, $P_{c-1} = P_c\prod_{i=c}^{c}\phi_i$, and the rest of the steady-state vector $[P_c, P_{c+1}, P_{c+2}, ...]$ can be determined recursively using $P_i = P_c\mathbf{R}^{i-c}$, for $i \ge c$. Once the level probability P_c is obtained, the steady-state solutions $[P_0, P_1, P_2, ..., P_{c-1}, P_c, P_{c+1}, ...]$ can be determined. The steady-state probability P_c can be solved by $(2.27f)$ and the following normalization equation

$$\begin{aligned}\sum_{n=0}^{\infty} P_n\mathbf{e} &= [P_0 + P_1 + ... + P_{c-1} + P_c + P_{c+1} + P_{c+2} + ...]\mathbf{e} \\ &= [P_c\prod_{i=c}^{1}\phi_i + P_c\prod_{i=c}^{2}\phi_i + ... + P_c\prod_{i=c}^{c}\phi_i + P_c + P_c\mathbf{R} + P_c\mathbf{R}^2 + ...]\mathbf{e} \\ &= P_c[\sum_{k=1}^{c}\prod_{i=c}^{k}\phi_i + \mathbf{I} + \mathbf{R}(\mathbf{I}-\mathbf{T})^{-1}]\mathbf{e} = 1.\end{aligned} \tag{2.28}$$

Solving equations $(2.27f)$ and (2.28) in accordance with Cramer's rule, we obtain P_c. Next, computing the prior state probabilities $[P_0, P_1, P_2, ..., P_{c-1}]$ from (2.27) and obtaining $[P_{c+1}, P_{c+2}, ...]$ by the formula $P_i = P_c\mathbf{R}^{i-c}$, $i \ge c+1$.

2.5 System Performance Measures

The system performance measures, such as the expected number of customers in the *FES* channel (denoted by L_1), the expected number of customers in the *SOS* channel (denoted by L_2), the expected number of customers in the system (denoted by L_s), the expected number of idle servers (denoted by $E[I]$) and the expected number of busy servers in the system (denoted by $E[B]$), can be evaluated from the steady-state probabilities $P_i = [P_{i,0}, P_{i,1}, P_{i,2}, ..., P_{i,R}]$. The expressions for L_1, L_2, L_s, $E[I]$, and $E[B]$ are given by

$$L_1 = \sum_{i=1}^{\infty} iP_i\mathbf{e}, \tag{2.29}$$

$$L_2 = \sum_{i=0}^{\infty} P_i\mathbf{u}, \tag{2.30}$$

$$L_s = L_1 + L_2, \tag{2.31}$$

$$E[I] = \sum_{i=0}^{c-1} P_i\mathbf{v}_i, \tag{2.32}$$

$$E[B] = c - E[I]. \tag{2.33}$$

where $\mathbf{u} = [0,1,2,...,c]^{\mathrm{T}}$ and $\mathbf{e} = [1,...,1]^{\mathrm{T}}$ are column vectors with dimension $c+1$. $\mathbf{v}_i$ is also a column vector with dimension $c+1$ with the j^{th} elements is $\max(0, c-i-j+1)$. The summation in (2.29) and (2.30) has an infinite number of terms and its computation is cumbersome. We provide another explicit formula for L_s which simplifies the computational procedure.

$$\begin{aligned} L_s &= L_1 + L_2 \\ &= \sum_{i=1}^{c-1} iP_i\mathbf{e} + [cP_c + (c+1)P_c\mathbf{R} + ...]\mathbf{e} + \sum_{i=0}^{c-1} P_i\mathbf{u} + [P_c + P_c\mathbf{R} + ...]\mathbf{u} \\ &= \sum_{i=1}^{c-1} iP_i\mathbf{e} + cP_c(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e} + P_R\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e} + \sum_{i=0}^{c-1} P_i\mathbf{u} + P_c(\mathbf{I}-\mathbf{R})^{-1}\mathbf{u} \\ &= \sum_{i=1}^{c-1} P_i(i\mathbf{e}+\mathbf{u}) + P_c(\mathbf{I}-\mathbf{R})^{-1}(c\mathbf{e}+\mathbf{u}) + P_c\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e}. \end{aligned} \tag{2.34}$$

For an infinite capacity M/M/c queueing system with second optional service channel, the numerical results of L_s are obtained by considering the following three cases with different values of c.

Case 1. $\mu_1 = 15$, $\mu_2 = 5$, $\theta = 0.05$, vary the values of λ from 0.5 to 10.

Case 2. $\lambda = 10$, $\mu_1 = 15$, $\theta = 0.05$, vary the values of μ_2 from 2.5 to 10.

Case 3. $\lambda = 10$, $\mu_2 = 5$, $\theta = 0.05$, vary the values of μ_1 from 15 to 25.

Results for L_s are depicted in Figures 2.2-2.4 for Cases 1-3, respectively. One sees from Figure 2.1 that L_s drastically increases as λ increases for $c = 1$, while L_s slightly increases as λ increases for $c \geq 2$. From Figures 2.3 and 2.4 we can see that L_s drastically decreases as μ_1 or μ_2 increases for $c = 1$, while L_s is not sensitive to μ_1 or μ_2 for $c \geq 2$.

2.6 Numerical Results

In this section, we construct the total expected cost function per customer per unit

time based on the system performance measures presented in the previous section. Our main objective is to determine the optimum number of server c, say c^*, and the optimal value of the service rate $\mu=(\mu_1,\mu_2)$, say $\mu^*=(\mu_1^*,\mu_2^*)$, simultaneously, so that the expected cost function is minimized. To do this, we define the following cost elements:

$C_h \equiv$ cost per unit time per customer present in the system,
$C_1 \equiv$ cost per unit time when one server is busy,
$C_2 \equiv$ cost per unit time of providing a service rate μ_1,
$C_3 \equiv$ cost per unit time of providing a service rate μ_2,
$C_4 \equiv$ fixed cost for purchase of one server.

Using these cost elements listed above, the expect cost function $F(c,\mu_1,\mu_2)$ is given by

$$F(c,\mu_1,\mu_2)=C_hL_s+C_1E[B]+C_2\mu_1+C_3\mu_2+C_4c. \tag{2.35}$$

The cost function in (2.35) are assumed to be linear in the mean number of indicated quantity, and it would have been a hard task to develop analytic results for the optimum value (c^*,μ_1^*,μ_2^*) because the expected cost function is highly complex and non-linear in terms of (c,μ_1,μ_2). In the next section, we firstly use the Quasi-Newton method to find the optimal value of continuous variable (μ_1,μ_2), say (μ_1^*,μ_2^*), and then use the direct search method to search the optimal value of discrete variable c, say c^*. For practice, the number of servers is bounded by a positive integer $c_U \geq 1$. We want to find the joint optimal value (μ_1^*,μ_2^*) for each given c in the feasible set $\{1,2,\ldots,c_U\}$. The cost minimization problem can be illustrated mathematically as

$$F(c,\mu_1^*,\mu_2^*)=\min_{(\mu_1,\mu_2)}\{F(c,\mu_1,\mu_2)|c\},\quad c=1,2,...,c_U, \tag{2.36}$$

subject to equation (2.20), the stability condition. For the problem of (2.36), it is difficult to show the convexity of $F(c,\mu_1,\mu_2)$ in (μ_1,μ_2).We note that the derivative of the cost function F with respect to (μ_1,μ_2) indicates the direction which the cost function increases. It means that, the optimal value (μ_1^*,μ_2^*) can be found along this opposite direction of the gradient (see Chong and Zak [15]). That is, for a fixed *c*, the Quasi-Newton method is employed to search (μ_1,μ_2) until the minimum value of $F(c,\mu_1,\mu_2)$ is achieved, say $F(c,\mu_1^*,\mu_2^*)$. To demonstrate the valid and the process of the optimization method, some examples are performed in Table 2.1 by considering the following cost parameters

$$C_h=\$250/\text{customer/unit-time},\quad C_1=\$180/\text{server/ unit-time},$$

$C_2 = \$15$ / unit-time, $C_3 = \$30$ / unit-time, and $C_4 = \$60$ /server.

Under other given parameters, one can find from Table 2.1 that the minimum expected cost per unit time of **1682.21** is achieved at $(\mu_1^*, \mu_2^*) = (\mathbf{27.3756, 14.0267})$ by using 6 iterations, which is $c = 3$ based on Case (i) with initial value $(\mu_1, \mu_2) = (20, 10)$. Based on Case (ii) with $c = 2$ and initial value $(\mu_1, \mu_2) = (20, 20)$, the minimum expected cost per unit time of **1737.30** is achieved at $(\mu_1^*, \mu_2^*) = (\mathbf{28.8310}, \mathbf{18.7206})$ by using 6 iterations.

After we obtain the joint optimal value (μ_1^*, μ_2^*) of the continuous variable (μ_1, μ_2), we will use the direct search method to obtain the optimal c such that the expected cost function $F(c, \mu_1^*, \mu_2^*)$ attains a minimum, say $F(c^*, \mu_1^*, \mu_2^*)$. Therefore, the cost minimization problem can be illustrated mathematically as

$$F(c^*, \mu_1^*, \mu_2^*) = \min_{c \in \{1,2,\ldots,c_U\}} \left\{ F(c, \mu_1^*, \mu_2^*) \right\}. \tag{2.37}$$

The procedure to find the optimal solution is described in the following. A numerical example is shown in Table 2.2 based on (i) $(\lambda, \theta) = (15, 0.5)$ and (ii) $(\lambda, \theta) = (20, 0.8)$. Based on Table 2.2, it is noted that the optimal value $(c^*, \mu_1^*, \mu_2^*) = (\mathbf{3, 22.86016, 11.64466})$ and the corresponding minimum cost $F^* = \mathbf{1463.830}$ for Case (i). For Case (ii), $(c^*, \mu_1^*, \mu_2^*) = (\mathbf{4, 25.40649, 16.13801})$ and $F^* = \mathbf{1891.530}$ are optimal. Finally, we perform a sensitivity investigation to the optimal value (c^*, μ_1^*, μ_2^*) based on changes in specific values of the system parameters. The numerical results are shown in Table 2.3 for various values of θ and λ. We find that (i) c^* increases as λ or θ increases; and (ii) μ_1^* (μ_2^*) increases as λ (θ) increases. Moreover, the minimum expected cost increases as λ or θ increases.

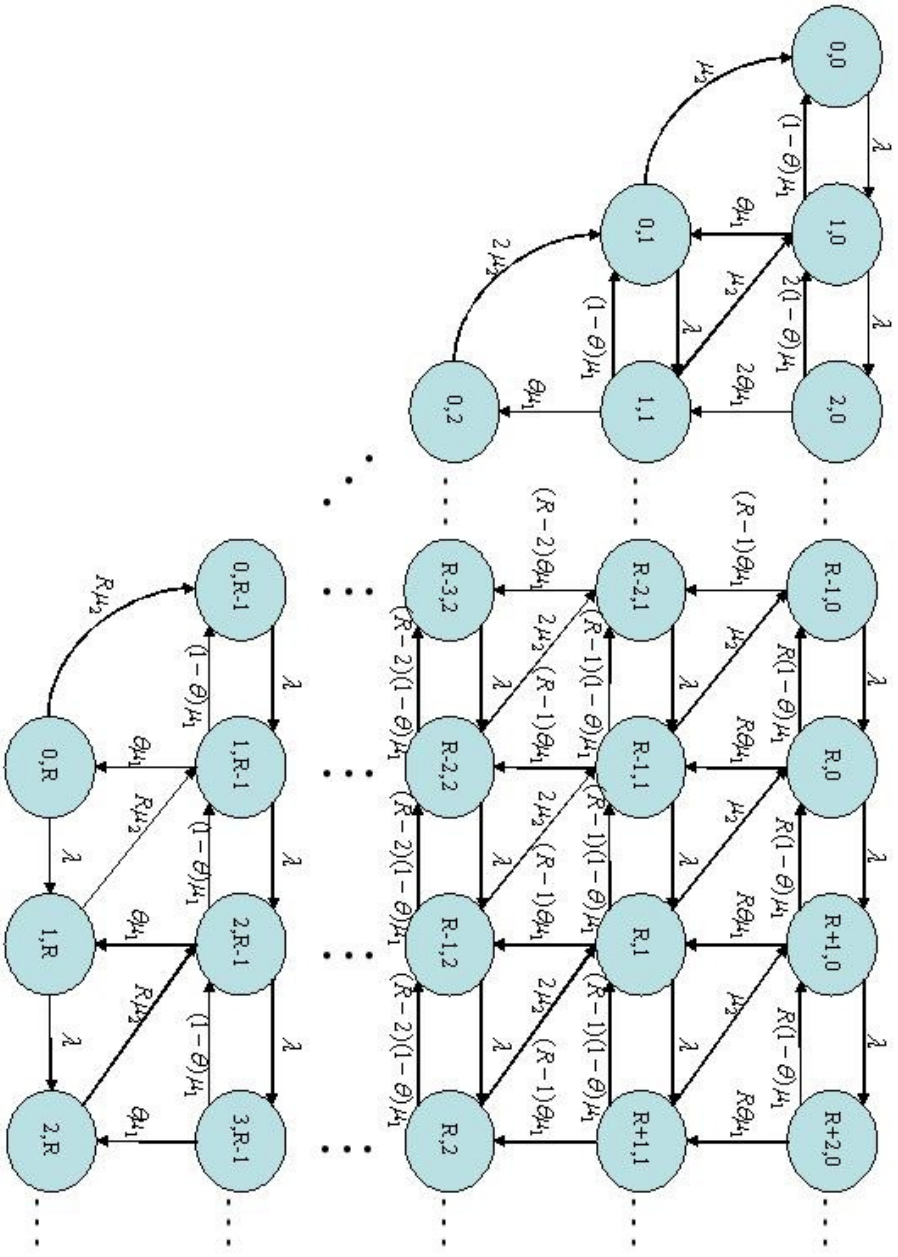

Figure 2.1. Steady-transition-rate diagram for an M/M/*c* queueing system with second optional service channel.

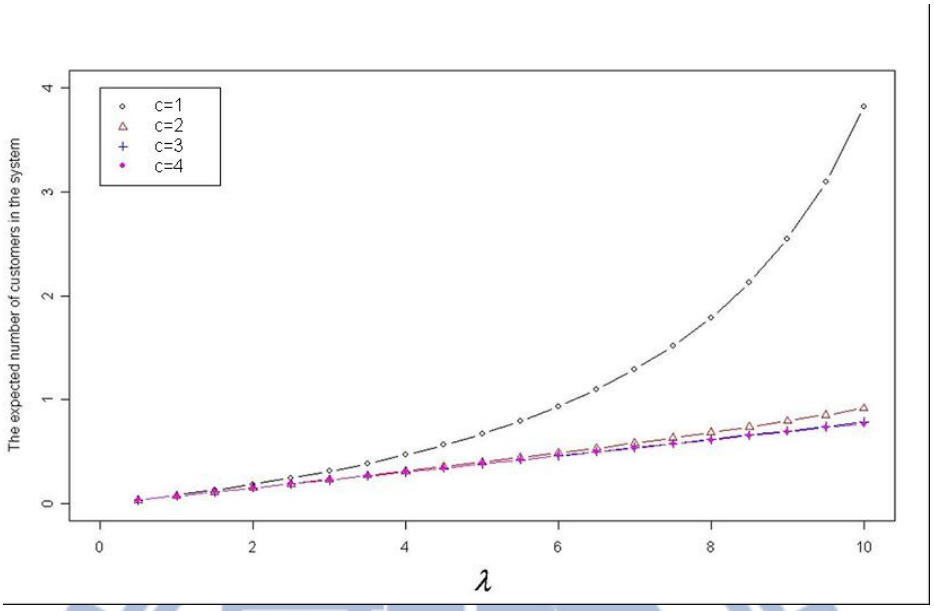

Figure 2.2. The expected number of customers in the system versus λ.

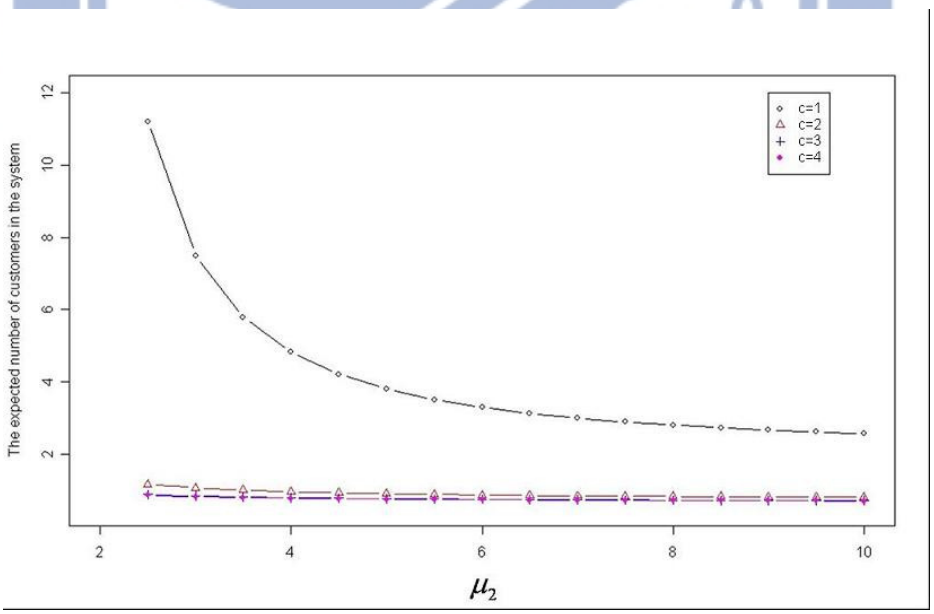

Figure 2.3. The expected number of customers in the system versus μ_2.

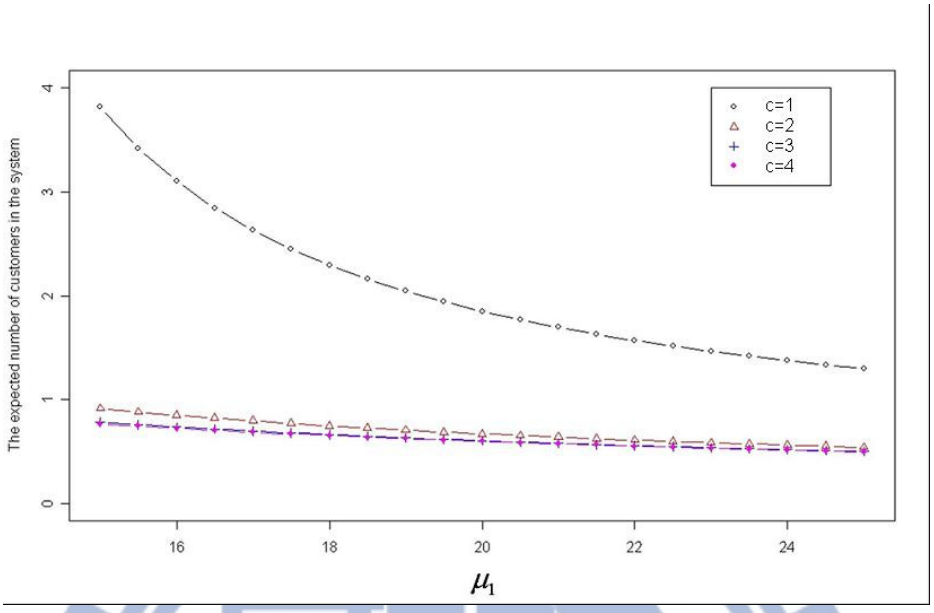

Figure 2.4. The expected number of customers in the system versus μ_1.

Table 2.1. The illustrations of the implementation of Quasi-Newton method

Case (i): $(\lambda, \theta) = (20, 0.5)$ with $c = 3$ and initial value $(\mu_1, \mu_2) = (20, 10)$

Iterations	0	1	2	3	4	5	6
$F(c, \mu_1, \mu_2)$	1862.22	1735.76	1689.68	1682.43	1682.21	1682.22	1682.21
μ_1	20	22.7766	25.5320	27.0701	27.3668	27.3756	**27.3756**
μ_2	10	11.4360	12.9115	13.8155	14.0192	14.0267	**14.0267**
$\frac{\partial F}{\partial \mu_1}$	-32.3746	-12.3033	-3.53768	-0.51339	-0.01504	-0.00001	-3×10^{-9}
$\frac{\partial F}{\partial \mu_2}$	-74.8311	-28.6228	-8.49048	-1.33248	-0.04521	-0.00005	-4.7×10^{-9}
L_s	2.88890	2.22232	1.83577	1.67455	1.64478	1.64379	1.64379
$E[B]$	2.000002	1.75253	1.55783	1.46265	1.44412	1.44350	1.44350
Hessian	$\begin{bmatrix} 7.990 & 7.096 \\ 7.096 & 38.39 \end{bmatrix}$	$\begin{bmatrix} 3.354 & 2.075 \\ 2.075 & 15.52 \end{bmatrix}$	$\begin{bmatrix} 1.828 & 0.803 \\ 0.803 & 8.026 \end{bmatrix}$	$\begin{bmatrix} 1.386 & 0.501 \\ 0.501 & 5.812 \end{bmatrix}$	$\begin{bmatrix} 1.317 & 0.456 \\ 0.456 & 5.441 \end{bmatrix}$	$\begin{bmatrix} 1.315 & 0.455 \\ 0.455 & 5.429 \end{bmatrix}$	$\begin{bmatrix} 1.315 & 0.455 \\ 0.455 & 5.428 \end{bmatrix}$

Case (ii): $(\lambda,\theta)=(15, 0.8)$ with $c=2$ and initial value $(\mu_1,\mu_2)=(20, 20)$

Iterations	0	1	2	3	4	5	6
$F(c,\mu_1,\mu_2)$	1829.50	1760.25	1739.61	1737.33	1737.30	1737.30	1737.30
μ_1	20	23.8016	27.0887	28.6094	28.8273	28.8310	**28.8310**
μ_2	20	19.2062	18.8294	18.7303	18.7207	18.7206	**18.7206**
$\frac{\partial F}{\partial \mu_1}$	-30.3036	-10.9797	-2.84444	-0.32356	-0.00538	-1.8×10^{-6}	0.
$\frac{\partial F}{\partial \mu_2}$	-6.92424	-3.14200	-1.01269	-0.13222	-0.00232	-9.9×10^{-7}	$-3\times10^{(-10)}$
L_s	2.26602	1.72458	1.73603	1.66634	1.65691	1.65674	1.65674
$E[B]$	1.35000	1.25501	1.19104	1.16498	1.16134	1.16128	1.16128
Hessian	$\begin{bmatrix}8.634 & 3.172\\ 3.172 & 6.468\end{bmatrix}$	$\begin{bmatrix}3.525 & 1.609\\ 1.609 & 5.696\end{bmatrix}$	$\begin{bmatrix}1.936 & 1.008\\ 1.008 & 5.239\end{bmatrix}$	$\begin{bmatrix}1.522 & 0.829\\ 0.829 & 5.042\end{bmatrix}$	$\begin{bmatrix}1.472 & 0.807\\ 0.807 & 5.012\end{bmatrix}$	$\begin{bmatrix}1.471 & 0.807\\ 0.807 & 5.012\end{bmatrix}$	$\begin{bmatrix}1.471 & 0.807\\ 0.807 & 5.012\end{bmatrix}$

Table 2.2. The optimal value (μ_1,μ_2) and the corresponding minimum expected cost

(i) $(\lambda,\theta)=(15, 0.5)$

c	**Initial Value**	**Coverage Value** (μ_1^*,μ_2^*)	**Iteration**	**Cost***
$c=1$	[30, 25]	[44.20521, 24.33688]	7	2022.146
$c=2$	[20, 20]	[27.50290, 14.50211]	6	1527.743
$c=3$	[15, 15]	[22.86016, 11.64466]	6	1463.830
$c=4$	[15, 10]	[21.33382, 10.71376]	6	1492.969
$c=5$	[15, 10]	[20.88151, 10.44900]	5	1545.927

(ii) $(\lambda,\theta)=(20, 0.8)$

c	**Initial Value**	**Coverage Value** (μ_1^*,μ_2^*)	**Iteration**	**Cost***
$c=1$	[50, 30]	[61.14970, 40.31473]	9	2890.717
$c=2$	[40, 30]	[35.80379, 23.29807]	8	2056.578
$c=3$	[30, 25]	[28.23610, 18.09640]	8	1896.310
$c=4$	[25, 20]	[25.40649, 16.13801]	5	1891.530
$c=5$	[20, 15]	[24.38956, 15.44162]	5	1933.145

Table 2.3. The optimal value (c^*, μ_1^*, μ_2^*) and it's minimum expected value $F(c^*, \mu_1^*, \mu_2^*)$ for various value of λ and θ.

(λ, θ)	(5, 0.2)	(10, 0.2)	(20, 0.2)	(5, 0.8)	(10, 0.8)	(20,0.8)
c^*	2	2	3	2	3	4
(μ_1^*, μ_2^*)	[13.0953, 4.35200]	[19.9021, 6.80977]	[26.3424, 8.64436]	[13.7175, 8.80645]	[18.2622, 11.6276]	[25.4065, 16.1380]
$F(c^*, \mu_1^*, \mu_2^*)$	729.6488	1011.985	1391.119	976.8809	1356.801	1897.530
L_s	0.690286	0.983412	1.346797	0.958229	1.326524	1.864544
$E[B]$	0.611596	0.796155	1.221962	0.818713	1.235596	1.778650

(λ, θ)	(10, 0.2)	(10, 0.5)	(10, 0.8)	(20, 0.2)	(20, 0.5)	(20, 0.8)
c^*	2	3	3	3	3	4
(μ_1^*, μ_2^*)	[19.9021, 6.80977]	[17.9854, 9.09991]	[18.2622, 11.6276]	[26.3424, 8.64436]	[27.37559, 14.02674]	[25.4065, 16.1380]
$F(c^*, \mu_1^*, \mu_2^*)$	1011.985	1215.012	1356.801	1391.119	1682.213	1897.530
L_s	0.983412	1.173003	1.326524	1.346797	1.643788	1.864544
$E[B]$	0.796155	1.105460	1.235596	1.221962	1.443501	1.778650

Chapter 3

M/M/*c* Retrial Queue with Second Optional Service Channel

In some cases, service stations and clients do not know the information of each other. Therefore, the customers will not enter the queueing system immediately even if there are idle servers. This situation arises in many telephony switching systems, telecommunication networks and computer systems. The retrial queueing systems play important roles in the analysis of these problems. Consider a Web system, all user login for the service of internet network browse, some of them will require the optional file transmission upload / download service. As the Web system is fully loaded, the user will retry after a random period of time.

In this chapter, we consider the queueing system investigated in chapter 2 with customer retrial behavior. An arriving primary customer finding one or more servers available (free) obtains the *FES* service immediately. On the other hand, he joins to the orbit and tries to get the service later on if all servers are busy and unavailable. Each customer staying in the orbit makes the repeated attempts in random intervals and is independently of the other customers. Upon requesting service from the orbit, customers finds all servers busy always rejoins the orbit; this manner continues until he is eventually served. An arbitrary customer in the orbit generates a stream of repeated requests that is independent of the rest of customers in the orbit.

This chapter is organized as follows: Basic assumptions and notations of the queueing model are given in Section 3.1. In Section 3.2, the mathematical model and the state-transition matrix are provided. In Section 3.3, the stability condition for this model is derived. A sequence approximation of the rate matrix is performed. Then, the steady-state solutions are obtained using recursively procedure. Section 3.4 devoted to develop the implicit expressions of the important system performances. Finally, Section 3.5 presents the optimization results and some numerical illustrations.

3.1 Assumptions and Notations

An M/M/*c* retrial queue with second optional service (*SOS*) is investigated. The service times of the first essential service (*FES*) and the second optional service (*SOS*) have an exponential distribution with mean $1/\mu_1$ and $1/\mu_2$, respectively. As soon as the first essential service of a customer is completed, a customer may leave the system

with probability (1-θ) or may opt for the second optional service with probability θ ($0 \le \theta \le 1$), at the completion of which the customer departs from the system and the next customer, if any, from the queue is taken up for his first essential service (see Figure 3.1). Each channel can serve only one customer at a time and it also provides only one of essential service or second optional service at a time. Furthermore, each customer staying in the orbit makes the repeated attempts in random intervals having length exponentially distributed with parameter σ, independently of the other customers.

A state of the system is a pair (i, j, k), where i and k denote the number of servers busy in the *FES* and *SOS*, respectively. j is the number of customers in the orbit (sources of repeated demands). The system can be described by a continuous parameter Markov chain on the state space $\{(i, j, k);\ 0 \le i \le c,\ 0 \le j,\ 0 \le k \le c - i\}$. From Figure 3.2, the customers which upon the server will get services immediately as $i + k < c$ (*i.e.* there are available servers). The new arriving customer who finds all c servers busy ($i + k \ge c$) always rejoins the retrial group (orbit), this operation continuous until they are eventually served. In steady-state, we define

$P_{i,j}^{k} \equiv$ probability that there are i and k servers busy in the *FES* and *SOS*, respectively, and j customers in orbit, where $0 \le i + k \le c$, $j \ge 0$.

In this chapter, the following notations and symbols are used.

λ — mean arrival rate

μ_1 — mean service rate of *FES* channel

μ_2 — mean service rate of *SOS* channel

θ — probability that a customer may opt for the *SOS*

σ — mean retrial rate

c — number of channels (servers)

Π — steady-state probability vector

Q — infinitesimal generator

I — identity matrix

e — identity column vector (a column vector with all elements equal to 1)

F — irreducible generator

x — invariant probability

P_F — probability that all servers are busy

R — rate matrix

$E[FES]$ — expected number of customers in the *FES* channel

$E[SOS]$ — expected number of customers in the *SOS* channel

$E[Orbit]$ — expected number of idle servers

σ_1^* — the overall rate of retrials

σ_2^* — the rate of retrials that are successful

FR — the fraction of retrials that are successful

$E(T)$ — mean busy period

P_V — vain retrials

F — cost function

It is assumed that there exists an upper bound N on the number of customers in the orbit that are allowed to conduct retrials (see Neuts and Rao [50] and Artalejo and Pozo [9]). This implies that the probability of a repeated attempt during $(t, t+dt)$, given that j customers in the orbit at time t, is $\sigma_j dt + o(dt)$, where $\sigma_j = \min\{j, N\}\sigma$. Moreover, we assume that the process of primary arrivals, service times and inter-retrial times are mutually independent.

3.2 M/M/*c* Retrial Queue with *SOS* Channel

For an M/M/*c* retrial queue with *SOS* channel, refer to Figure 3.2, the infinitesimal generator $\mathbf{Q}$ of the Markov chain has the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{C}_N & \mathbf{A}_{N-1} & \mathbf{B} & \\ & & & & \mathbf{C}_N & \mathbf{A}_N & \mathbf{B} \\ & & & & & \mathbf{C}_N & \mathbf{A}_N & \mathbf{B} \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{3.1}$$

The entries $\mathbf{B}$, $\mathbf{A}_j (j>0)$, and $\mathbf{C}_j (j>1)$ are block-diagonal matrices of order $(c+1)(c+2)/2$. Matrices $\mathbf{B}$ and $\mathbf{C}_j$ can be partitioned as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_0 & & & & \\ & \mathbf{b}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{b}_{c-1} & \\ & & & & \mathbf{b}_c \end{bmatrix} \text{ and } \mathbf{C}_j = \begin{bmatrix} \mathbf{c}_0 & & & & \\ & \mathbf{c}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{c}_{c-1} & \\ & & & & \mathbf{c}_c \end{bmatrix}, \quad j = 1, 2, ...,$$

where sub-matrices $\mathbf{b}_i$ and $\mathbf{c}_i$ are square matrices of order $(c+1-i)$ with

$$\begin{cases} \mathbf{b}_i[c+1-i, c+1-i] = \lambda, \\ 0 \qquad\qquad\qquad e.w. \end{cases} \text{ and } \begin{cases} \mathbf{c}_i[k, k+1] = \sigma_j = \min\{j, N\}\sigma, \ 1 \le k \le c-i, \\ 0 \qquad\qquad\qquad\qquad e.w. \end{cases}$$

$\mathbf{A}_j$ can be partitioned as

$$\mathbf{A}_j = \begin{bmatrix} \mathbf{Y}_j^0 & \mathbf{X}^0 & & & & \\ \mathbf{Z}^1 & \mathbf{Y}_j^1 & \mathbf{X}^1 & & & \\ & \mathbf{Z}^2 & \mathbf{Y}_j^2 & \mathbf{X}^2 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{Z}^{c-1} & \mathbf{Y}_j^{c-1} & \mathbf{X}^{c-1} \\ & & & & \mathbf{Z}^c & \mathbf{Y}_j^c \end{bmatrix}, \quad j = 0,1,2,...,$$

where $\mathbf{X}^i$ is a $(c+1-i)\times(c-i)$ matrix with $\mathbf{X}^i[k+1,k] = k\theta\mu_1,\ 1 \le k \le c-i$, $\mathbf{Z}^i$ is a $(c-i)\times(c+1-i)$ matrix with $\mathbf{Z}^i[k,k] = i\mu_2,\ 1 \le k \le c-i$, and $\mathbf{Y}_j^i$ is a square matrix of order $(c+1-i)$ with elements

$$\begin{cases} \mathbf{Y}_j^i[k,k+1] = \lambda,\ 1 \le k \le c-i, \\ \mathbf{Y}_j^i[k+1,k] = k(1-\theta)\mu_1,\ 1 \le k \le c-i, \\ \mathbf{Y}_j^i[1,1] = -[\lambda + (i-1)\mu_2 + \sigma_j], \\ \mathbf{Y}_j^i[k,k] = -[\lambda + (k+1)\mu_1 + i\mu_2 + \sigma_j],\ 2 \le k \le c-i, \\ \mathbf{Y}_j^i[c+1-i, c+1-i] = -[\lambda + i\mu_2 + (c-i)\mu_1]. \end{cases}$$

For instance, for c=3, the sub-matrices of $\mathbf{B}$ are

$$\mathbf{b}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix}, \ \mathbf{b}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix}, \ \mathbf{b}_2 = \begin{bmatrix} 0 & 0 \\ 0 & \lambda \end{bmatrix}, \ \mathbf{b}_3 = \lambda .$$

The sub-matrices of $\mathbf{C}_1$ are

$$\mathbf{c}_0 = \begin{bmatrix} 0 & \sigma & 0 & 0 \\ 0 & 0 & \sigma & 0 \\ 0 & 0 & 0 & \sigma \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ \mathbf{c}_1 = \begin{bmatrix} 0 & \sigma & 0 \\ 0 & 0 & \sigma \\ 0 & 0 & 0 \end{bmatrix}, \ \mathbf{c}_2 = \begin{bmatrix} 0 & \sigma \\ 0 & 0 \end{bmatrix}, \ \mathbf{c}_3 = 0 .$$

For $\mathbf{A}_1$, the first sup-diagonal sub-matrices are

$$\mathbf{X}_0 = \begin{bmatrix} 0 & 0 & 0 \\ \theta\mu_1 & 0 & 0 \\ 0 & 2\theta\mu_1 & 0 \\ 0 & 0 & 3\theta\mu_1 \end{bmatrix}, \quad \mathbf{X}_1 = \begin{bmatrix} 0 & 0 \\ \theta\mu_1 & 0 \\ 0 & 2\theta\mu_1 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 0 \\ \theta\mu_1 \end{bmatrix}.$$

The diagonal sub-matrices are

$$\mathbf{Y}_0 = \begin{bmatrix} -(\lambda+\sigma) & \lambda & & \\ (1-\theta)\mu_1 & -(\lambda+\mu_1+\sigma) & \lambda & \\ & 2(1-\theta)\mu_1 & -(\lambda+2\mu_1+\sigma) & \lambda \\ & & 3(1-\theta)\mu_1 & -(\lambda+3\mu_1) \end{bmatrix},$$

$$\mathbf{Y}_1 = \begin{bmatrix} -(\lambda+\mu_2+\sigma) & \lambda & \\ (1-\theta)\mu_1 & -(\lambda+\mu_1+\mu_2+\sigma) & \lambda \\ & 2(1-\theta)\mu_1 & -(\lambda+2\mu_1+\mu_2) \end{bmatrix},$$

$$\mathbf{Y}_2 = \begin{bmatrix} -(\lambda+2\mu_2+\sigma) & \lambda \\ (1-\theta)\mu_1 & -(\lambda+\mu_1+2\mu_2) \end{bmatrix}, \quad \mathbf{Y}_3 = -(\lambda+3\mu_2).$$

The first sub-diagonal sub-matrices are

$$\mathbf{Z}_1 = \begin{bmatrix} \mu_2 & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & 0 \\ 0 & 0 & \mu_2 & 0 \end{bmatrix}, \quad \mathbf{Z}_2 = \begin{bmatrix} 2\mu_2 & 0 & 0 \\ 0 & 2\mu_2 & 0 \end{bmatrix}, \quad \mathbf{Z}_3 = \begin{bmatrix} 3\mu_2 & 0 \end{bmatrix}.$$

After the derivation of the mathematical model, the steady-state can be represented in matrix form.

3.3 Steady-state Results

In this section, we will derive the stability condition and obtain the steady-state probability vectors by using recursive technique. Let $\mathbf{\Pi} = [\Pi_0, \Pi_1, \Pi_2, ...]$ with $\Pi_i = [P_{0,i}^0, P_{1,i}^0, ..., P_{c,i}^0, P_{0,i}^1, P_{1,i}^1, ..., P_{c-1,i}^1, P_{0,i}^0, ..., P_{0,i}^{c-1}, P_{1,i}^{c-1}, P_{0,i}^c]$, $i = 0, 1, 2, ...$ be the unique solution to $\mathbf{\Pi Q} = \mathbf{0}$ and $\mathbf{\Pi e} = 1$, where $\mathbf{e}$ is a column vector with all elements equal to 1. An efficient algorithm is developed to calculate the stationary probabilities

by matrix-geometric method.

3.3.1. Stability condition

It is known that the stationary probability exists if and only if

$$\mathbf{xBe} < \mathbf{xC}_N\mathbf{e}, \tag{3.2}$$

where $\mathbf{x} = [x_0^0, x_1^0, ..., x_c^0, ..., x_0^{c-1}, x_1^{c-1}, x_0^c]$ is the invariant probability of the matrix $\mathbf{F} = \mathbf{C}_N + \mathbf{A}_N + \mathbf{B}$. $\mathbf{x}$ satisfies $\mathbf{xF} = \mathbf{0}$ and $\mathbf{xe} = 1$ where $\mathbf{e}$ is a column vector with dimension $(c+1)(c+2)/2$ and all elements equal to one. Solving two equations simultaneously, we have

$$x_i^k = \frac{c!\mu_2^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu_1^i\theta^{c-k}} x_0^c, \ 0 \le i+k \le c. \tag{3.4}$$

$$x_0^c = \left[\sum_{k=0}^{c} \sum_{i=0}^{c-k} \frac{c!\mu_2^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu_1^i\theta^{c-k}} \right]^{-1}. \tag{3.5}$$

Substituting $\mathbf{B}$ and $\mathbf{C}_N$ into equation (5) and doing some routine manipulations, then we have

$$N\sigma(1-P_F) > \lambda P_F, \tag{3.8}$$

where

$$\begin{aligned} P_F &= \sum_{i=0}^{c} x_i^{c-i} = \sum_{i=0}^{c} \frac{c!\mu_2^i}{i!(c-i)!\mu_1^i\theta^i} x_0^c \\ &= \left(1 + \frac{\mu_2}{\theta\mu_1}\right)^c \left[\sum_{k=0}^{c} \sum_{i=0}^{c-k} \frac{c!\mu_2^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu_1^i\theta^{c-k}} \right]^{-1}. \end{aligned} \tag{3.7}$$

denotes the probability that all server are busy (i.e. $i+k=c$). That is, the system will be stable if the expected successful retrial rate is greater then the expected arrival rate of "orbit".

3.3.2. Rate matrix

By matrix-geometric property, it is noted that the vector $\mathbf{\Pi} = [\Pi_0, \Pi_1, \Pi_2, \Pi_3, ...]$ has the following properties

$$\Pi_{N+k} = \Pi_N \mathbf{R}^k, \text{ for } k \ge 1. \tag{3.9}$$

The matrix $\mathbf{R}$, called rate matrix, is the unique non-negative solution with spectral

radius less than one of the equation

$$\mathbf{R}^2\mathbf{B} + \mathbf{R}\mathbf{A}_N + \mathbf{C}_N = \mathbf{0}. \tag{3.10}$$

From Neuts [49] and Latouche and Ramaswami [41], it is known that a approximation solution of the rate matrix $\mathbf{R}$ can be gain by the converge of sequence $\lim_{n\to\infty}\mathbf{R}_n$, where the sequence $\{\mathbf{R}_n\}$ is defined by

$$\mathbf{R}_0 = \mathbf{0}, \text{ and } \mathbf{R}_{n+1} = -\mathbf{B}\mathbf{A}_N^{-1} - \mathbf{R}_n^2\mathbf{C}_N\mathbf{A}_N^{-1}, \text{ for } n \geq 0. \tag{3.11}$$

The sequence $\{\mathbf{R}_n\}$ is monotone so that $\mathbf{R}$ can be evaluated from (3.6) by successive substitutions.

3.3.3. Recursive solver

Under the stability condition, the stationary probability vector $\mathbf{\Pi}$ of $\mathbf{Q}$ exists. In the above section, we deal with the steady-state equations by representing it in matrix form. This steady-state probability vector $\mathbf{\Pi} = [\Pi_0, \Pi_1, \Pi_2, \Pi_3, ...]$ is given by

$$\Pi_0\mathbf{A}_0 + \Pi_1\mathbf{C}_1 = \mathbf{0}, \tag{3.7a}$$

$$\Pi_{i-1}\mathbf{B} + \Pi_i\mathbf{A}_i + \Pi_{i+1}\mathbf{C}_{i+1} = \mathbf{0}, \ 1 \leq i \leq N-1, \tag{3.7b}$$

$$\Pi_{N-1}\mathbf{B} + \Pi_N\mathbf{A}_N + \Pi_N\mathbf{R}\mathbf{C}_N = \mathbf{0}, \tag{3.7c}$$

$$\Pi_N\mathbf{R}^{i-1-N}\mathbf{B} + \Pi_N\mathbf{R}^{i-N}\mathbf{A}_N + \Pi_N\mathbf{R}^{i+1-N}\mathbf{C}_N = \mathbf{0}, \ N+1 \leq i, \tag{3.7d}$$

$$\sum_{i=0}^{\infty}\Pi_i\mathbf{e} = 1. \tag{3.8}$$

After doing some routine manipulations to equation (3.7a)-(3.7c), we have

$$\begin{aligned} \Pi_0 &= \Pi_1\mathbf{C}_1(-\mathbf{A}_0)^{-1} = \Pi_1\phi_1, \\ \Pi_{i-1} &= \Pi_i\mathbf{C}_i[-(\phi_{i-1}\mathbf{B} + \mathbf{A}_{i-1})]^{-1} = \Pi_i\phi_i, \ 2 \leq i \leq N, \end{aligned} \tag{3.9}$$

and

$$\Pi_N\phi_N\mathbf{B} + \Pi_N\mathbf{A}_N + \Pi_N\mathbf{R}\mathbf{C}_N = \mathbf{0}. \tag{3.10}$$

Consequently, $\Pi_i\ (0 \leq i \leq N-1)$ in equation (3.9) can be written as product form in terms of Π_N and the rest steady-state vector $[\Pi_N, \Pi_{N+1}, \Pi_{N+2}, ...]$ can be determined recursively as $\Pi_i = \Pi_N\mathbf{R}^{i-N}$, for $i \geq N$. Once the steady-state probability Π_N is obtained, the steady-state solutions $[\Pi_0, \Pi_1, \Pi_2, ..., \Pi_{N-1}, \Pi_N, \Pi_{N+1}, ...]$ are determined. The steady-state probability Π_N can be solved by equation (3.10) with the following normalization equation

$$\sum_{i=0}^{\infty}\Pi_i\mathbf{e}=[\Pi_0+\Pi_1+...+\Pi_{N-1}+\Pi_N+\Pi_{N+1}+\Pi_{N+2}+...]\mathbf{e}$$
$$=[\Pi_N\prod_{i=N}^{1}\phi_i+\Pi_N\prod_{i=N}^{2}\phi_i+...+\Pi_N\prod_{i=N}^{N}\phi_i+\Pi_N+\Pi_N\mathbf{R}+\Pi_N\mathbf{R}^2+...]\mathbf{e}. \quad (3.11)$$
$$=\Pi_N[\sum_{k=1}^{N}\prod_{i=N}^{k}\phi_i+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1$$

The symbol $\mathbf{I}$ denotes the identity matrix with suitable size. Solving equations (3.10) and (3.11) in accordance with Cramer's rule, we obtain Π_N. Then the prior state probabilities $[\Pi_0,\Pi_1,\Pi_2,\ ...,\Pi_{N-1}]$ are computed from (3.9) recursively and $[\Pi_{N+1},\Pi_{N+2},\Pi_{N+3},...]$ are gained by the formula $\Pi_i=\Pi_N\mathbf{R}^{i-N}$, $i\geq N+1$. We summarize the solution procedure of steady-state probabilities as below:

Algorithm: Recursive Solver

Step 1. Set $\phi_1=\mathbf{C}_1(-\mathbf{A}_0)^{-1}$.

Step 2. For *i* from 2 to *N*, set $\phi_i=\mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1}$.

Step 3. For *k* from 1 to *N*, set $\Phi_k=\prod_{i=N}^{k}\phi_i$.

Step 4. Solving $\Pi_N\phi_N\mathbf{B}+\Pi_N\mathbf{A}_N+\Pi_N\mathbf{R}\mathbf{C}_N=\mathbf{0}$, $\Pi_N[\sum_{k=1}^{N}\Phi_k+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1$ and obtain the steady-state probability Π_N.

Step 5. Construct steady-state probability Π_i as follows:

(a) if $0\leq i\leq N$, assign $\Pi_i=\Pi_N\Phi_{i+1}$,

(b) if $N+1\leq i$, assign $\Pi_{i+1}=\Pi_iR$.

3.4 System Performance Measures

The system performance measures, such as the expected number of customers in the *FES* channel (denoted by *E*[*FES*]), the expected number of customers in the *SOS* channel (denoted by *E*[*SOS*]), and the expected number of customers in orbit (denoted by *E*[*Orbit*]), can be evaluated from the steady-state probabilities $\Pi_j=[P^0_{0,j},P^0_{1,j},...,P^0_{c,j},P^1_{0,j},P^1_{1,j},...,P^1_{c-1,j},...,P^{c-1}_{0,j},P^{c-1}_{1,j},P^c_{0,j}]$. The expressions for *E*[*FES*], *E*[*SOS*], and *E*[*Orbit*] are given by

$$
\begin{aligned}
E[FES] &= \sum_{i=0}^{\infty} \Pi_i \mathbf{v} = \sum_{i=0}^{N-1} \Pi_i \mathbf{v} + \Pi_N \mathbf{v} + \Pi_N \mathbf{R}\mathbf{v} + \Pi_N \mathbf{R}^2 \mathbf{v} + \ldots \\
&= \sum_{i=0}^{N-1} \Pi_N \Phi_{i+1} \mathbf{v} + \Pi_N \mathbf{v} + \Pi_N \mathbf{R}\mathbf{v} + \Pi_N \mathbf{R}^2 \mathbf{v} + \ldots \\
&= \Pi_N [\sum_{i=1}^{N} \Phi_i + (\mathbf{I} - \mathbf{R})^{-1}] \mathbf{v},
\end{aligned}
\tag{3.12}
$$

$$
\begin{aligned}
E[SOS] &= \sum_{i=0}^{\infty} \Pi_i \mathbf{J} = \sum_{i=0}^{N-1} \Pi_i \mathbf{J} + \Pi_N \mathbf{J} + \Pi_N \mathbf{R}\mathbf{J} + \Pi_N \mathbf{R}^2 \mathbf{J} + \ldots \\
&= \sum_{i=0}^{N-1} \Pi_N \Phi_{i+1} \mathbf{J} + \Pi_N (\mathbf{I} - \mathbf{R})^{-1} \mathbf{J} = \Pi_N [\sum_{i=1}^{N} \Phi_i + (\mathbf{I} - \mathbf{R})^{-1}] \mathbf{J},
\end{aligned}
\tag{3.13}
$$

$$
\begin{aligned}
E[Orbit] &= \sum_{i=1}^{\infty} i \Pi_i \mathbf{e} \\
&= \sum_{i=1}^{N-1} i \Pi_N \Phi_{i+1} \mathbf{e} + N \Pi_N \mathbf{e} + (N+1) \Pi_N \mathbf{R}\mathbf{e} + (N+2) \Pi_N \mathbf{R}^2 \mathbf{e} + \ldots \\
&= \sum_{i=2}^{N} (i-1) \Pi_N \Phi_i \mathbf{e} + \Pi_N [N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}] \mathbf{e} \\
&= \Pi_N [\sum_{i=2}^{N} (i-1) \Phi_i + N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}] \mathbf{e},
\end{aligned}
\tag{3.14}
$$

where

$$
\mathbf{v} = [\underbrace{0,1,\ldots,c}_{\#=c+1}, \underbrace{0,1,\ldots,c-1}_{\#=c}, \ldots, \underbrace{0,1}_{\#=2}, 0] \quad \text{and} \quad \mathbf{J} = [\underbrace{0,0,\ldots,0}_{\#=c+1}, \underbrace{1,1,\ldots,1}_{\#=c}, \ldots, \underbrace{c-1,c-1}_{\#=2}, c]
$$

are column vectors with dimension $(c+1)(c+2)/2$. For an M/M/*c* retrial queue with second optional service channel, the numerical results of *E*[*Orbit*] are obtained by considering the following three cases with different values of *c*

Case 1. N =30, $\lambda = 5$, $\mu_2 = 10$, $\theta = 0.5$, $\sigma = 5$, vary μ_1 from 10 to 20.

Case 2. N =30, $\lambda = 5$, $\mu_1 = 10$, $\theta = 0.5$, $\sigma = 5$, vary μ_2 from 10 to 20.

Case 3. N =30, $\mu_1 = 20$, $\mu_2 = 15$, $\theta = 0.5$, $\sigma = 5$, vary λ from 5 to 10.

Results of *E*[*Orbit*] are depicted in Figure 3.3 for Case 1-3, respectively. From the Figure, one sees that *E*[*Orbit*] drastically decreases (increase) as μ_1 or μ_2 (or λ) increases (decrease) for c =1, while *E*[*Orbit*] is not sensitive to μ_1 or μ_2 (or λ) for $c \geq 2$. Furthermore, there are several general descriptors of retrial queues, some of which are listed below:

1. The overall rate of retrials

$$\sigma_1^* = \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k = \sum_{j=1}^{N} j\sigma \Pi_j \mathbf{e} + \sum_{j=N+1}^{\infty} N\sigma \Pi_N \mathbf{R}^{j-N} \mathbf{e} \quad (3.15)$$

$$= \sum_{j=1}^{N} j\sigma \Pi_j \mathbf{e} + N\sigma \Pi_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1} \mathbf{e} = \sigma[\sum_{j=1}^{N} j\Pi_j + N\Pi_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}$$

$$= \sigma \Pi_N [\sum_{j=1}^{N-1} j\Phi_{j+1} + N(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}.$$

2. The rate of retrials that are successful

$$\sigma_2^* = \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k. \quad (3.16)$$

3. The fraction of retrials that are successful

$$FR = \frac{\sigma_2^*}{\sigma_1^*}. \quad (3.17)$$

4. The marginal distribution of the number of busy servers

$$p(i,k) = \sum_{j=0}^{\infty} P_{i,j}^k, \quad 0 \le i+k \le c. \quad (3.18)$$

5. Busy period : The busy period T of a retrial queue is defined as the period that starts at the epoch when an arriving customer finds an empty system (all servers are idle and no customer in the orbit) and ends at the departure epoch at which the system is empty again.

   The mean busy period

$$E(T) = \frac{1}{\lambda}(\frac{1}{P_{0,0}^0} - 1) = \frac{1}{\lambda}(\frac{1}{\Pi_N \Phi_1[1]} - 1) \quad (3.19)$$

   where $\Pi_N \Phi_1[1]$ denotes the first element of $\Pi_N \Phi_1$.

6. Vain retrials : A vain retrial is an unsuccessful retrial when all servers are busy.

   The steady-state probability of vain retrial P_V

$$P_V = \frac{\sum_{j=1}^{\infty} \sum_{i+k=c} P_{i,j}^k}{\sum_{j=1}^{\infty} \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k} = \frac{\sum_{j=1}^{\infty} \sum_{i+k=c} P_{i,j}^k}{1 - \Pi_0 \mathbf{e}}. \quad (3.20)$$

To understand how system performance measures listed above vary with N, we also perform a numerical investigation to the measures based on changing the value of

N. The numerical illustration is graphically presented in Figure 3.4. From Figure 3.4, it is clear that increasing the retrial rate beyond a certain point does not result in a commensurate improvement in the system performance, which is according with the result of Neuts and Rao [50].

3.5 Numerical Results

We construct a total expected cost function per unit time, in which the number of servers (c) is a discrete decision variable, and the service rates $\mu=(\mu_1,\mu_2)$ are continuous decision variables. Let us define the following cost elements:

$C_h \equiv$ cost per unit time per customer present in orbit,
$C_1 \equiv$ cost per unit time when one server is busy in *FES* channel
$C_2 \equiv$ cost per unit time of providing an service rate μ_1,
$C_3 \equiv$ cost per unit time of providing an service rate μ_2,
$C_4 \equiv$ fixed cost for purchase one server.

Based on the definition of the cost parameters, the total expected cost function per unit time is given by

$$F(c,\mu_1,\mu_2)=C_hE[Orbit]+C_1E[B]+C_2\mu_1+C_3\mu_2+C_4c\,. \tag{3.21}$$

The main objective is to determine the optimal number of servers c^*, and the optimal value of the service rate $\mu^*=(\mu_1^*,\mu_2^*)$, simultaneously which minimize the cost function. The analytic study of the optimization behavior of the expected cost function is an arduous task to undertake since the decision variable appears in an expression which is a highly complex and non-linear in terms of (c,μ_1,μ_2). We firstly use direct search method to find the optimal value of the number of servers, say c^*, when μ_1 and μ_2 are fixed. Next, we fix c^* and use the Quasi-Newton method to search/adjust the optimal value of (μ_1,μ_2), say (μ_1^*,μ_2^*). In practical application, an upper bound *U* is imposed on *c*. We can successively substitute $c=1,2,\ldots,U$ into the cost function. The optimum value c^* can be determined by satisfying the following inequality

$$F(c^*-1\,|\,\mu_1,\mu_2)>F(c^*\,|\,\mu_1,\mu_2)<F(c^*+1\,|\,\mu_1,\mu_2)\,. \tag{3.22}$$

To demonstrate that the cost function is really convex in c and the solution gives a minimum, some numerical examples are performed based on the preceding formulation. For convenience, the number $N=30$ is chosen and the following cost

elements are adopted

C_h =\$25/customer/unit time, C_1 =\$120/server/unit time, C_2 =\$15/unit time,

C_3 =\$30/unit time, and C_4 =\$180/server.

Under other parameters are given, we observe from Table 3.1 that the optimal number of servers c^* and its corresponding minimum cost increase as θ or λ increases, and decrease as σ increases. After we obtain c^*, Quasi-Newton method is employed to search (μ_1, μ_2) until the minimum value of $F(c^*, \mu_1, \mu_2)$ is achieved, say $F(c^*, \mu_1^*, \mu_2^*)$. To find the joint optimal value (μ_1^*, μ_2^*) for a given c^*, we should show the convexity of $F(c^*, \mu_1, \mu_2)$. However, this work is difficult to implement. Two examples are presented to illustrate the optimization procedure shown in Table 3.2. From Table 3.2, we can see that the minimum expected cost per day of **1003.92** is achieved at (μ_1^*, μ_2^*) = (**23.4453**, **8.02222**) by using 5 iterations, which is c^* =1 based on Case (i) with initial value (μ_1, μ_2) = (20, 10). Based on Case (ii), c^* is 4 and the minimum expected cost per day of **1674.11** is achieved at (μ_1^*, μ_2^*) = (**16.8630**, **10.7441**) is achieved using 5 iterations.

We now perform a sensitivity investigation to the optimal value (c^*, μ_1^*, μ_2^*) based on changes in specific values of the system parameters. The numerical results are shown in Table 3.3 for various values of λ, θ, and σ by considering the initial value (μ_1, μ_2) of (20, 10). From Table 3.3, we find that (i) c^* increases as λ or θ increases and is insensitive to the change of σ; and (ii) μ_1^* (μ_2^*) increases as θ (λ) increases and decreases as σ increases.

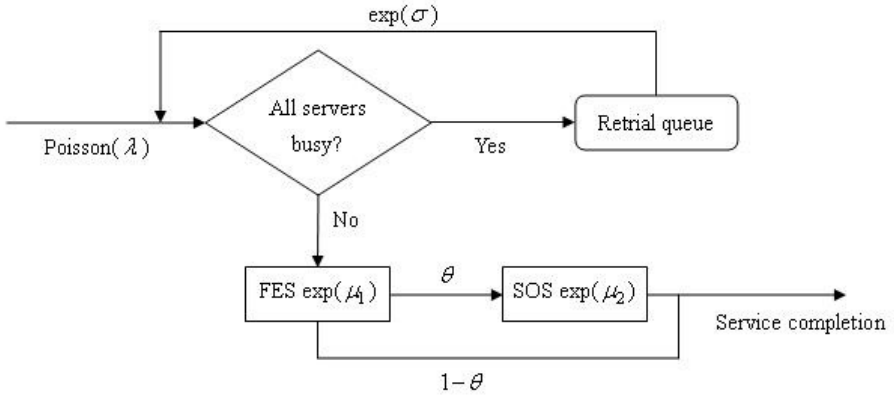

Figure 3.1. The general structure of M/M/*c* retrial queue with second optional service.

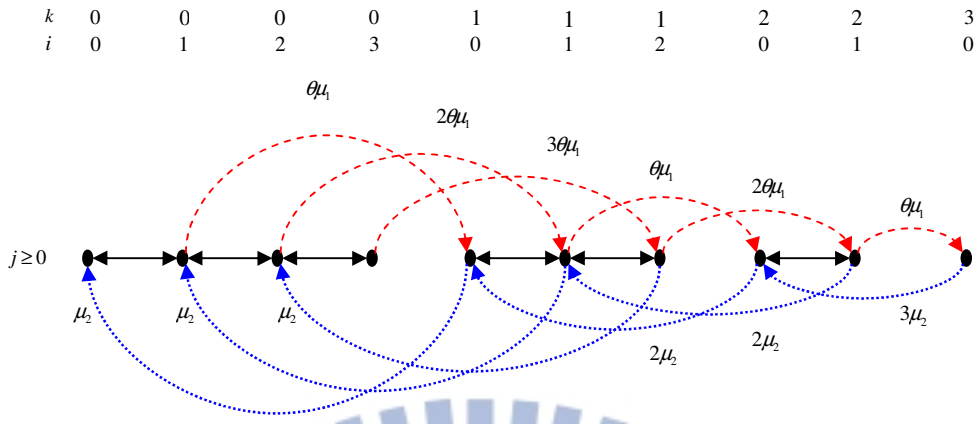

Figure 3.2. State-transition-rate diagram for an M/M/3 retrial queue with *SOS*.

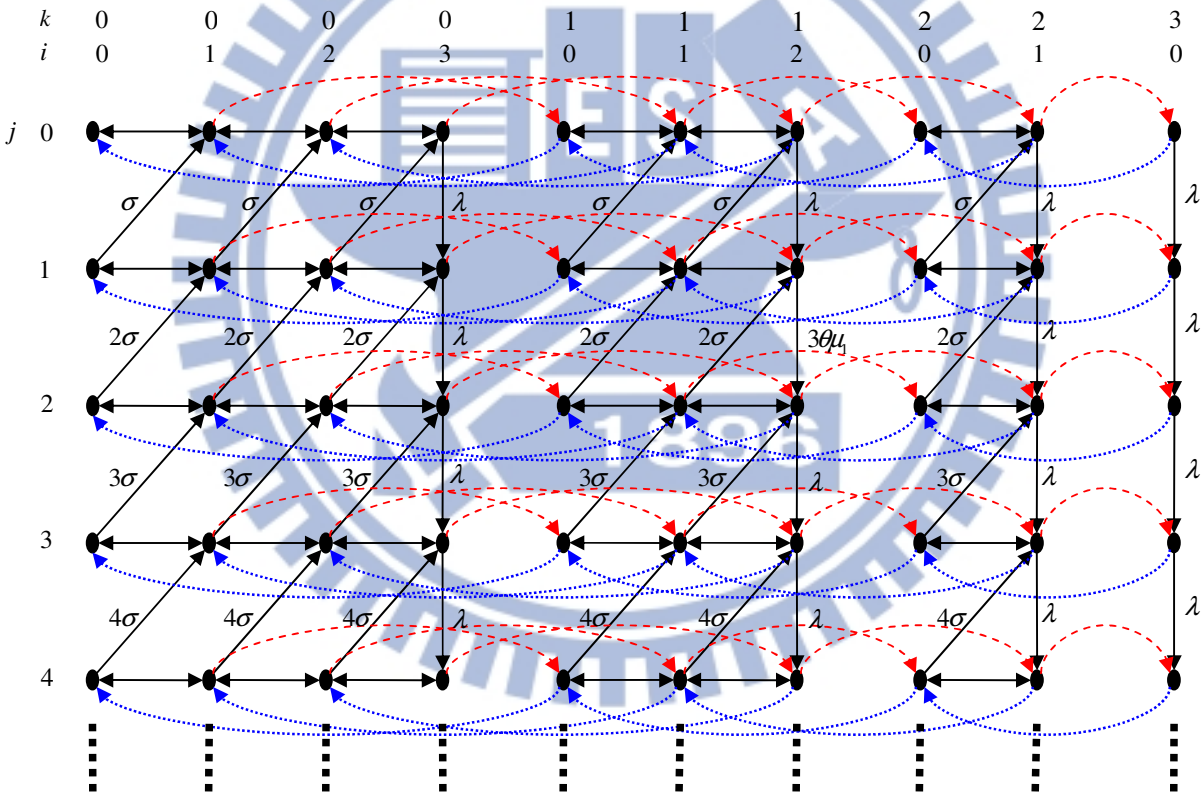

Figure 3.2 (cont.). State-transition-rate diagram for an M/M/3 retrial queue with *SOS*.

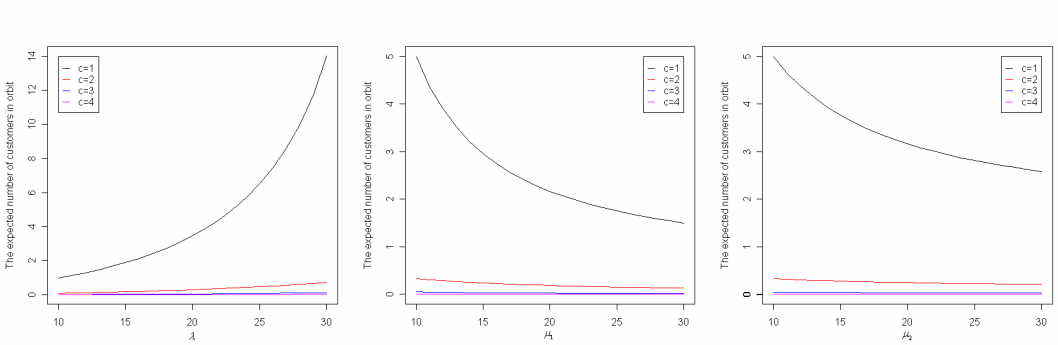

Figure 3.3. The expected number of customers in orbit versus λ, μ_1 and μ_2

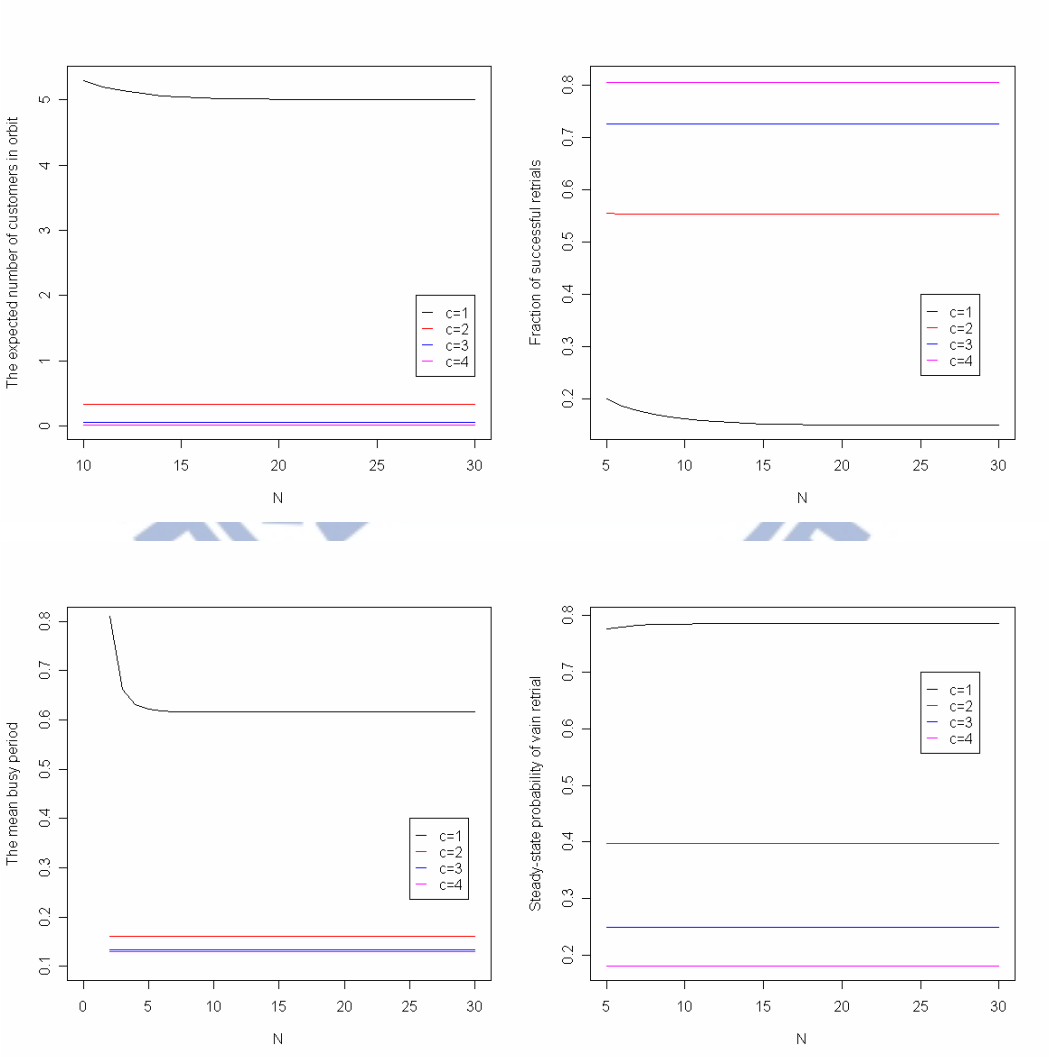

Figure 3.4. The system performance measures versus the truncated parameter *N*.

Table 3.1. The cost function associated with number of servers and values of λ.

$(\mu_1, \mu_2, \lambda, \theta, \sigma)$	c=1	c=2	c=3	c=4	c=5	c=6
(20, 10, 5, 0.2, 5)	**840.75**	1003.57	1182.15	1362.01	1542.00	1722.00
(20, 10, 10, 0.2, 5)	**1026.54**	1056.44	1225.94	1404.29	1584.04	1764.00
(20, 10, 15, 0.2, 5)	N/A*	**1133.07**	1274.45	1447.72	1626.32	1806.05
(20, 10, 20, 0.2, 5)	N/A	**1277.00**	1332.65	1493.81	1669.37	1848.30
(20, 10, 5, 0.2, 10)	**834.019**	1002.95	1182.086	1362.01	1542.00	1722.00
(20, 10, 10, 0.2, 10)	**968.171**	1051.67	1225.14	1404.17	1584.02	1764.00
(20, 10, 15, 0.2, 10)	N/A	**1115.42**	1271.07	1446.99	1626.18	1806.03
(20, 10, 20, 0.2, 10)	N/A	**1222.10**	1323.03	1491.42	1668.78	1848.16
(20, 10, 5, 0.2, 15)	**831.776**	1002.74	1182.07	1362.01	1542.00	1722.00
(20, 10, 10, 0.2, 15)	**948.724**	1050.06	1224.87	1404.12	1584.02	1764.00
(20, 10, 15, 0.2, 15)	N/A	**1109.48**	1269.93	1446.75	1626.13	1806.02
(20, 10, 20, 0.2, 15)	N/A	**1203.68**	1319.78	1490.61	1668.58	1848.12

$(\mu_1, \mu_2, \lambda, \theta, \sigma)$	c=1	c=2	c=3	c=4	c=5	c=6
(20, 10, 5, 0.8, 5)	**930.322**	1043.87	1218.86	1398.12	1578.02	1758.00
(20, 10, 10, 0.8, 5)	N/A	**1179.97**	1306.96	1478.47	1656.55	1836.11
(20, 10, 15, 0.8, 5)	N/A	4365.50	**1431.25**	1567.86	1737.90	1915.07
(20, 10, 20, 0.8, 5)	N/A	N/A	1778.63	**1683.62**	1827.33	1996.97
(20, 10, 5, 0.8, 10)	**907.107**	1041.82	1218.53	1398.07	1578.01	1758.00
(20, 10, 10, 0.8, 10)	N/A	**1158.61**	1303.03	1477.52	1656.32	1836.06
(20, 10, 15, 0.8, 10)	N/A	2668.80	**1411.62**	1562.80	1736.40	1914.64
(20, 10, 20, 0.8, 10)	N/A	N/A	1668.51	**1665.55**	1821.65	1995.03
(20, 10, 5, 0.8, 15)	**899.369**	1041.12	1218.42	1398.06	1578.01	1758.00
(20, 10, 10, 0.8, 15)	N/A	**1151.44**	1301.70	1477.20	1656.25	1836.05
(20, 10, 15, 0.8, 15)	N/A	2347.02	**1405.00**	1561.08	1735.88	1914.49
(20, 10, 20, 0.8, 15)	N/A	N/A	**1632.44**	1659.44	1819.73	1994.38

* "N/A" denotes system is unstable (i.e., the stable condition does not hold)

Table 3.2. The illustration of the implementation process of Newton-Quasi method

Case (i): $(\lambda,\theta,\sigma)=(10, 0.2, 5)$ with initial value $(\mu_1,\mu_2)=(20,10)$

Iterations	0	1	2	3	4	5
$F(c^*,\mu_1,\mu_2)$	1026.54	1007.83	1004.06	1003.92	1003.92	**1003.92**
c^*	1	1	1	1	1	1
μ_1	20	22.6527	23.3038	23.4405	23.4453	**23.4453**
μ_2	10	7.55324	7.92219	8.01767	8.02221	**8.02222**
$\frac{\partial F}{\partial \mu_1}$	-8.24475	-4.34979	-0.69766	-0.25822	-0.00004	-4.0×10^{-10}
$\frac{\partial F}{\partial \mu_2}$	9.22640	-10.6671	-1.80336	-0.07392	-0.00014	-1.5×10^{-9}
$E[Orbit]$	6.50153	7.06782	6.20192	6.02626	6.01906	6.01905
$E[BS]^*$	0.70000	0.70623	0.68157	0.67606	0.67583	0.67583
Hessian	$\begin{bmatrix}5.8322 & 2.9535\\ 2.9535 & 6.9730\end{bmatrix}$	$\begin{bmatrix}4.0699 & 4.6075\\ 4.6075 & 20.781\end{bmatrix}$	$\begin{bmatrix}2.9736 & 3.0487\\ 3.0487 & 14.521\end{bmatrix}$	$\begin{bmatrix}2.7867 & 2.7845\\ 2.7845 & 13.382\end{bmatrix}$	$\begin{bmatrix}2.7796 & 2.7738\\ 2.7738 & 13.334\end{bmatrix}$	$\begin{bmatrix}2.7805 & 2.7742\\ 2.7742 & 13.334\end{bmatrix}$

Case (ii): $(\lambda,\theta,\sigma)=(20, 0.8, 5)$ with initial value $(\mu_1,\mu_2)=(20,10)$

Iterations	0	1	2	3	4	5
$F(c^*,\mu_1,\mu_2)$	1683.62	1675.80	1674.15	1674.11	1674.11	**1674.11**
c^*	4	4	4	4	4	4
μ_1	20	15.7834	16.6998	16.8593	16.8630	**16.8630**
μ_2	10	10.7383	10.7446	10.7442	10.7441	**10.7441**
$\frac{\partial F}{\partial \mu_1}$	4.88607	-3.28908	-0.42979	-0.00954	-0.000005	1.3×10^{-9}
$\frac{\partial F}{\partial \mu_2}$	-2.97931	-2.06125	-0.25997	-0.00563	-0.000005	1.0×10^{-9}
$E[Orbit]$	2.06473	2.64154	2.35613	2.31366	2.31270	2.31269
$E[BS]$	2.60000	2.75714	2.68674	2.67547	2.67521	2.67521
Hessian	$\begin{bmatrix}1.3534 & 1.1115\\ 1.1115 & 10.383\end{bmatrix}$	$\begin{bmatrix}3.5744 & 2.1813\\ 2.1813 & 9.9203\end{bmatrix}$	$\begin{bmatrix}2.6986 & 1.6548\\ 1.6548 & 8.9978\end{bmatrix}$	$\begin{bmatrix}2.5796 & 1.5845\\ 1.5845 & 8.8708\end{bmatrix}$	$\begin{bmatrix}2.5761 & 1.5820\\ 1.5820 & 8.8674\end{bmatrix}$	$\begin{bmatrix}2.5770 & 1.5823\\ 1.5823 & 8.8671\end{bmatrix}$

*" $E[BS]$ " denotes the number of busy servers in the system $\equiv E[FES]+E[SOS]$.

Table 3.3. The optimal value (c^*, μ_1^*, μ_2^*) and the minimum expected cost value for various value of λ, θ, and σ, while c^* is obtained at initial value $(\mu_1, \mu_2)=(20,10)$.

(λ,θ,σ)	(5, 0.2, 10)	(10, 0.2, 10)	(20, 0.2, 10)	(5, 0.8, 10)	(10, 0.8, 10)	(20,0.8, 10)
c^*	1	1	2	1	2	4
(μ_1^*, μ_2^*)	[11.8535, 4.26058]	[22.0254, 7.71166]	[22.7810, 7.77980]	[14.7456, 9.53810]	[15.1755, 9.76186]	[16.2154, 10.3483]
$F(c^*, \mu_1, \mu_2)$	628.502	947.158	1200.47	866.965	1129.98	1652.39
E[*Orbit*]	2.56395	4.79291	3.93227	3.54492	2.88299	1.80663
E[*BS*]	0.65653	0.71337	1.39208	0.75845	1.47847	2.77955

(λ,θ,σ)	(10, 0.2, 15)	(10, 0.5, 15)	(10, 0.8, 15)	(20, 0.2, 15)	(20, 0.5, 15)	(20,0.8, 15)
c^*	1	2	2	2	3	3
(μ_1^*, μ_2^*)	[21.4641, 7.60174]	[13.8213, 7.19819]	[14.9257, 9.61603]	[22.2164, 7.65119]	[18.2749, 9.41276]	[19.6974, 12.6312]
$F(c^*, \mu_1, \mu_2)$	925.598	1009.21	1118.22	1181.57	1417.74	1561.528
E[*Orbit*]	4.32420	2.23082	2.62489	3.52100	2.41688	2.93133
E[*BS*]	0.72899	1.41814	1.50193	1.42303	2.15678	2.28207

(λ,θ,σ)	(10, 0.2, 5)	(10, 0.2, 10)	(10, 0.2, 15)	(10, 0.8, 5)	(10, 0.8, 10)	(10,0.8, 15)
c^*	1	1	1	2	2	2
(μ_1^*, μ_2^*)	[23.4453, 8.02222]	[22.0254, 7.71166]	[21.4641, 7.60174]	[15.8259, 10.1461]	[15.1755, 9.76186]	[14.9257, 9.61603]
$F(c^*, \mu_1, \mu_2)$	1003.92	947.158	925.598	1161.21	1129.98	1118.22
E[*Orbit*]	2.31269	4.79291	4.32420	3.55994	2.88299	2.62489
E[*BS*]	2.67521	0.71337	0.72899	1.42036	1.47847	1.50193

Chapter 4

M/M/*c* Queue with Modified Bernoulli Vacation Policy

In some transport systems in which a ferry driver or a locomotive driver may take a vacation after every round of trip. In the restaurant, the waiter may go to the restroom when there is no guest waiting for taking their order. For a production system, it may so happen that the process either needs to be stopped for overhauling and maintenance of the system after usual processing (see Choudhury and Madan [22]). The overhauling may be represented as a Bernoulli vacation time in our system.

In this chapter, matrix analytic method is used to analyze an infinite capacity multi-server queue with modified Bernoulli vacation under a single vacation policy. In traditional Bernoulli vacation policy, servers may take a vacation at the completion of service with probability p or continuous to serve the next customer with probability $1-p$. For the modified Bernoulli vacation policy, the server will keep providing service to the customer if there are customers still waiting in the queue. At this time, the vacation behavior will not occur, that is, $p=0$. The modified Bernoulli vacation policy is more suitable for the real situation. When the servers complete the vacation period, they stay idly for the next new arrival or serve the customers in the system, if any. That is, the single vacation policy.

This chapter is organized as follows: Section 4.1 gives some basic assumptions and notations of the queue. In Section 4.2, the mathematical analyses of the state-transition matrix are presented. In Section 4.3, the stability condition, the closed-form expression of rate matrix, and the steady-state probability are derived. In Section 4.4, the explicit expressions of some important system performance are obtained. The special case of single server is also discussed. The optimization results and numerical examples are performed in Section 4.5.

4.1 Assumptions and Notations

An infinite capacity M/M/*c* queueing system with modified Bernoulli vacation under a single vacation policy is considered. Conveniently, we represent this multi-server system with modified Bernoulli vacation as M/M/*c*/MBSV queueing system. Customers arrive according to a Poisson process with parameter λ. Their service are provided by c servers, in which the service times are assumed to be exponentially distributed with mean $1/\mu$. It is assumed that customers arrive at the

system form a single waiting line and served in the order of their arrivals; that is, the first-come, first-served discipline. Each server can serve only one customer at a time, and that the service is independent of the arrival of the customers. At each service completion instant of a server, the server inspects the system state and decides whether leave for a vacation or not. If the number of customers in the system is less than the number of busy/working servers, the server may take a vacation of random length with probability p or continue to serve the next customer, if any with proability q $(=1-p)$. The vacation times are exponentially distribution with mean $1/\eta$. If the number of customers in the system is more than the number of busy servers, the server always keep working/serving for the next customers waiting in the queue, that is, $p=0$. At the end of the vacation, the server remains idle until the first arriving customer, that is, the single vacation policy.

In this chapter, the following notations and symbols are used.

λ — mean arrival rate

μ — mean service rate

p — probability that a server may opt for Bernoulli vacation

η — vacation rate

c — number of channels (servers)

$\mathbf{\Pi}$ — steady-state probability vector

$\mathbf{Q}$ — infinitesimal generator

$\mathbf{I}$ — identity matrix

$\mathbf{e}$ — identity column vector (a column vector with all elements equal to 1)

$\mathbf{F}$ — irreducible generator

$\mathbf{x}$ — invariant probability

$\mathbf{R}$ — rate matrix

L_s — expected number of customers in the system

L_q — expected number of customers in the queue

$E[V]$ — expected number of vacation servers

$E[I]$ — expected number of idle servers

$E[B]$ — expected number of busy servers

F — cost function

4.2 M/M/*c* Queue with Bernoulli Vacation

For an M/M/*c*/MBSV queueing system, the state of this queueing system can be described by the pair (i,n), $i=0,1,2,...,c$, $n=0,1,2,...$, where i denotes the number of vacation servers, and n denotes the number of customers in the system. According to system assumptions, the server keeps busy and serves the next customer waiting in the queue at a service completion instant if the number of customers in the system is greater than the number of total working (no vacation) servers, *i.e.* $(n \geq c-i+1)$. Otherwise $(n \leq c-i)$, one server may goes on vacation with probability p $(p>0)$ or resumes service with probability $q=1-p$ at a service completion instant. In steady-state, we define

$P_i(n) \equiv$ probability that there are n customers in the system when there are i vacation servers, where $i=0,1,2,...,c$ and $n=0,1,2,...$

and $\mathbf{\Pi}=[\Pi_0,\Pi_1,...,\Pi_c,\Pi_{c+1},...]$ with $\Pi_n=[P_0(n),P_1(n),...,P_c(n)]$, $n=0,1,2,...$ denotes the steady-state probability vector. The infinitesimal generator $\mathbf{Q}$ of the QBD describing the M/M/*c* queueing system with modified Bernoulli vacation is of the form

$$\mathbf{Q}=\begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mathbf{C}_{c-1} & \mathbf{A}_{c-1} & \mathbf{B} & \\ & & & & \mathbf{C}_c & \mathbf{A}_c & \mathbf{B} \\ & & & & & \mathbf{C}_{c+1} & \mathbf{A}_c & \mathbf{B} \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{4.1}$$

The entries $\mathbf{B}$, $\mathbf{A}_j$ $(0 \leq j \leq c)$, and $\mathbf{C}_j$ $(1 \leq j \leq c+1)$ are matrices of order $(c+1)$.

$$\mathbf{A}_j=\begin{bmatrix} a_{j,1} & & & & & \\ \eta & a_{j,2} & & & & \\ & 2\eta & a_{j,3} & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & (c-1)\eta & a_{j,c} \\ & & & & & c\eta \quad a_{j,c+1} \end{bmatrix}, j=0,1,...,c.$$

The matrix $\mathbf{B} = \lambda \mathbf{I}_{(c+1)\times(c+1)}$, where $\mathbf{I}_{(c+1)\times(c+1)}$ is the identity matrix of order c+1. $\mathbf{A}_j$ $(0 \le j \le c)$ is also square matrix with dimension $(c+1)\times(c+1)$ shown above with diagonal elements $a_{j,k} = -[\lambda + \min(j, c+1-k)\mu + (k-1)\eta]$, $k = 1, 2, ..., c+1$. Then the matrix $\mathbf{C}_j$ $(1 \le j \le c+1)$ is list below

$$\mathbf{C}_j = \begin{bmatrix} \left.\begin{matrix} jq\mu & jp\mu & & & \\ & jq\mu & jp\mu & & \\ & & \ddots & \ddots & \\ & & & jq\mu & jp\mu \end{matrix}\right\} \#=(c+1-j) \\ \left.\begin{matrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix}\right\} \#=j \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \begin{bmatrix} \\ \\ \\ (j-1)\mu & & & \\ & (j-2)\mu & & \\ & & \ddots & \\ & & & \mu \\ & & & 0 \end{bmatrix}, \; j = 1, 2, .., c+1.$$

The steady-state equations system can be represented as $\mathbf{\Pi Q} = \mathbf{0}$.

4.3 Steady-state Results

To ensure that the unique solution of $\mathbf{\Pi Q} = \mathbf{0}$ exists. The stability condition of this queueing system should be derived. Certainly, obtaining rate matrix is necessary before employing the matrix geometric method. The convergence property of the rate matrix is proofed. We also discuss the special case of single server.

4.3.1. Stability condition

Form Neuts [49], the steady-state probability vector exists if and only if

$$\mathbf{xBe} < \mathbf{xC}_{c+1}\mathbf{e}, \tag{4.2}$$

where $\mathbf{e}$ is a column vector with dimension $(c+1)$ and all elements equal to one. The vector $\mathbf{x} = [x_0, x_1, ..., x_c]$ is the invariant probability of the matrix $\mathbf{F} = \mathbf{C}_{c+1} + \mathbf{A}_c + \mathbf{B}$. It satisfies two conditions $\mathbf{xF = 0}$ and $\mathbf{xe = 1}$.

Substituting $\mathbf{B}$ and $\mathbf{C}_{c+1}$ into equation (4.2) and doing some routine manipulations, then we have $x_0 = 1$ and $x_i = 0$, $i = 1, 2, ..., c$. The stability condition is given as

$$\lambda < \sum_{i=0}^{c} (c-i) x_i \mu = c\mu, \tag{4.3}$$

which agrees with the stability condition of the ordinary M/M/*c* queueing system.

4.3.2. Rate matrix

It is noted that the vector $\mathbf{\Pi} = [\Pi_0, \Pi_1, ..., \Pi_c, \Pi_{c+1}, ...]$ has the following properties

$$\Pi_{c+k} = \Pi_c \mathbf{R}^k, \text{ for } k \geq 1 \tag{4.4}$$

which is the matrix-geometric property. The matrix $\mathbf{R}$ is the unique non-negative solution with spectral radius less than one of the equation

$$\mathbf{B} + \mathbf{R}\mathbf{A}_c + \mathbf{R}^2\mathbf{C}_{c+1} = \mathbf{0}. \tag{4.5}$$

It is necessary to solve the rate matrix $\mathbf{R}$ of (4.5), in order to obtain the steady-state solution vector $\mathbf{\Pi} = [\Pi_0, \Pi_1, ..., \Pi_c, \Pi_{c+1}, ...]$. Based on the structures of matrices, $\mathbf{B}$, $\mathbf{A}_c$ and $\mathbf{C}_{c+1}$ which are represented as the lower triangular matrix, thus the matrix solution $\mathbf{R}$ is also the lower triangular matrix. (see Latouche and Ramaswami [41])

Doing some arduous algebraic derivations and arrangement, we develop the explicit formula for matrix $\mathbf{R}$ as follows:

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & & & & \\ r_{2,1} & r_{2,2} & & & \\ \vdots & \vdots & \ddots & & \\ \vdots & \vdots & & \ddots & \\ r_{c,1} & r_{c,2} & \cdots & \cdots & r_{c,c} \\ r_{c+1,1} & r_{c+1,2} & \cdots & \cdots & r_{c+1,c} \quad r_{c+1,c+1} \end{bmatrix}, \tag{4.6}$$

where $r_{i,i} = \dfrac{\theta_i - \sqrt{\theta_i^2 - 4(c+1-i)\lambda\mu}}{2(c+1-i)\mu}$, for $1 \leq i \leq c$, $r_{c+1,c+1} = \dfrac{\lambda}{\lambda + c\eta}$,

$$r_{i,j} = \frac{(c+1-j)\mu \sum_{k=j+1}^{i-1} r_{i,k} r_{k,j} + j r_{i,j+1}\eta}{\lambda + (c+1-j)(1 - r_{i,i} - r_{j,j})\mu + (j-1)\eta}, \text{ for } 2 \leq i \leq c+1,\ i > j,$$

$r_{i,j} = 0$, for $i < j$ and $\theta_i = \lambda + (c+1-i)\mu + (i-1)\eta$.

Note that $r_{i,i}$ is the corresponding eigen-values of the rate matrix $\mathbf{R}$ and the spectral radius of $\mathbf{R}$, $sp(\mathbf{R})$ is less than one if $\rho = \lambda(c\mu)^{-1} < 1$ (stability condition).

Proof. Firstly, the first diagonal element

$$r_{1,1} = \frac{\lambda + c\mu - \sqrt{(\lambda + c\mu)^2 - 4c\lambda\mu}}{2c\mu} = \frac{\lambda + c\mu - |\lambda - c\mu|}{2c\mu} = \lambda(c\mu)^{-1} = \rho < 1.$$

For $2 \le i \le c$, the diagonal element $r_{i,i}$ is got from the quadratic equation

$$f(x) = (c+1-i)\mu x^2 - \theta_i x + \lambda = 0.$$

It should be noted that there exists exact one real root in (0, 1) because

$$f(0) = \lambda > 0,$$
$$f(1) = (c+1-i)\mu - \theta_{i-1} + \lambda = -(i-1)\eta < 0.$$

Finally, $r_{c+1,c+1} = \lambda / (\lambda + c\eta) < 1$. Consequently, all diagonal elements (eigen-values) of rate matrix $\mathbf{R}$ are less than 1. Therefore, the spectral radius of rate matrix $\mathbf{R}$, $sp(\mathbf{R}) = \max_{1 \le i \le c+1} \{r_{i,i}\}$ is less than 1. That is, the convergence property is ensured if the stability condition holds. By using the rate matrix $\mathbf{R}$, we can solve the steady-state probability more efficiently.

4.3.3. Probability computation

Under the stability condition, by solving the equation $\mathbf{\Pi Q} = \mathbf{0}$ with the normalization condition, we obtain

$$\Pi_0 \mathbf{A}_0 + \Pi_1 \mathbf{C}_1 = \mathbf{0}, \tag{4.7a}$$

$$\Pi_{i-1} \mathbf{B} + \Pi_i \mathbf{A}_i + \Pi_{i+1} \mathbf{C}_{i+1} = \mathbf{0}, \ 1 \le i \le c-1, \tag{4.7b}$$

$$\Pi_{c-1} \mathbf{B} + \Pi_c \mathbf{A}_c + \Pi_c \mathbf{R} \mathbf{C}_{c+1} = \mathbf{0}, \tag{4.7c}$$

$$\Pi_c \mathbf{R}^{i-1-c} \mathbf{B} + \Pi_c \mathbf{R}^{i-c} \mathbf{A}_c + \Pi_c \mathbf{R}^{i+1-c} \mathbf{C}_{c+1} = \mathbf{0}, \ c+1 \le i, \tag{4.7d}$$

$$\sum_{i=0}^{\infty} \Pi_i \mathbf{e} = 1. \tag{4.8}$$

After doing routine substitutions to (4.7a)-(4.7c), we have

$$\begin{aligned} \Pi_0 &= \Pi_1 \mathbf{C}_1 (-\mathbf{A}_0)^{-1} = \Pi_1 \phi_1, \\ \Pi_{i-1} &= \Pi_i \mathbf{C}_i [-(\phi_{i-1} \mathbf{B} + \mathbf{A}_{i-1})]^{-1} = \Pi_i \phi_i, \ 2 \le i \le c, \end{aligned} \tag{4.9}$$

and

$$\Pi_c \phi_c \mathbf{B} + \Pi_c \mathbf{A}_c + \Pi_c \mathbf{R} \mathbf{C}_{c+1} = \mathbf{0}. \tag{4.10}$$

Consequently, Π_i $(0 \le n \le c-1)$ in equation (4.9) can be written in terms of Π_c as $\Pi_i = \Pi_c \prod_{i=c}^{i+1} \phi_i$, $i = 0,1,2,...,c-1$ where $\phi_1 = \mathbf{C}_1 (-\mathbf{A}_0)^{-1}$ and $\phi_i = \mathbf{C}_i [-(\phi_{i-1} \mathbf{B} + \mathbf{A}_{i-1})]^{-1}$, $2 \le i \le c$. The rest steady-state vectors $\Pi_c, \Pi_{c+1}, ...$ can be calculated recursively as $\Pi_i = \Pi_c \mathbf{R}^{i-c}$, for $i \ge c$. Once Π_c is determined, the steady-state solutions $\mathbf{\Pi} = [\Pi_0, \Pi_1, ..., \Pi_c, \Pi_{c+1}, ...]$ are obtained. The vector Π_c is given by solving equation (4.10) with the following normalization condition.

$$\sum_{i=0}^{\infty}\Pi_i\mathbf{e}=[\Pi_0+\Pi_1+\ldots+\Pi_{c-1}+\Pi_c+\Pi_{c+1}+\Pi_{c+2}+\ldots]\mathbf{e}$$
$$=[\Pi_c\prod_{i=c}^{1}\phi_i+\Pi_c\prod_{i=c}^{2}\phi_i+\ldots+\Pi_c\prod_{i=c}^{c}\phi_i+\Pi_c+\Pi_c\mathbf{R}+\Pi_c\mathbf{R}^2+\ldots]\mathbf{e} \quad (4.11)$$
$$=\Pi_c[\sum_{n=1}^{c}\prod_{i=c}^{n}\phi_i+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1.$$

Solving equations (4.10) and (4.11) in accordance with Cramer's rule, we obtain Π_c. Then the prior state probabilities $[\Pi_0,\Pi_1,\Pi_2,\ ...,\Pi_{c-1}]$ are computed from (4.9) and $[\Pi_{c+1},\Pi_{c+2},\Pi_{c+3},...]$ are gained by the formula $\Pi_i=\Pi_c\mathbf{R}^{i-c}$, $i\geq c+1$.

4.4 System Performance Measures

There are several general descriptors (system performance measures) of the M/M/*c*/MBSV queueing system, such as the expected number of customers in the system (denoted by L_s), the expected number of customers in the queue (denoted by L_q), the expected number of busy, idle and vacation servers (denoted by $E[B]$, $E[I]$ and $E[V]$, respectively). The expressions for these system performance measures are given by

$$L_s=\sum_{i=1}^{\infty}i\Pi_i\mathbf{e}=\sum_{i=1}^{c-1}i\Pi_i\mathbf{e}+c\Pi_c\mathbf{e}+(c+1)\Pi_c\mathbf{R}\mathbf{e}+\ldots$$
$$=\sum_{i=1}^{c-1}i\Pi_c\Phi_{i+1}\mathbf{e}+c\Pi_c(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e}+\Pi_c\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e} \quad (4.12)$$
$$=\Pi_c\left[\sum_{i=1}^{c-1}i\Phi_{i+1}+c(\mathbf{I}-\mathbf{R})^{-1}+\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\right]\mathbf{e}.$$

$$L_q=\Pi_1\begin{bmatrix}0\\ \vdots\\ 0\\ 1\end{bmatrix}+\Pi_2\begin{bmatrix}0\\ \vdots\\ 1\\ 2\end{bmatrix}+\cdots+\Pi_c\begin{bmatrix}0\\ \vdots\\ c-1\\ c\end{bmatrix}+\Pi_c\mathbf{R}\left(\begin{bmatrix}0\\ \vdots\\ c-1\\ c\end{bmatrix}+\mathbf{e}\right)+\ldots$$
$$=\sum_{i=1}^{c-1}\Pi_c\Phi_{i+1}\mathbf{u}_i+\Pi_c(\mathbf{I}-\mathbf{R})^{-1}\mathbf{u}_c+\Pi_c\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e} \quad (4.13)$$
$$=\Pi_c\left[\sum_{i=1}^{c-1}\Phi_{i+1}\mathbf{u}_i+(\mathbf{I}-\mathbf{R})^{-1}\mathbf{u}_c+\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e}\right].$$

$$E[V]=\sum_{i=0}^{\infty}\Pi_i\begin{bmatrix}0\\1\\\vdots\\c\end{bmatrix}=\Pi_c\left[\sum_{i=1}^{c}\Phi_i+(\mathbf{I}-\mathbf{R})^{-1}\right]\begin{bmatrix}0\\1\\\vdots\\c\end{bmatrix}. \tag{4.14}$$

$$\begin{aligned}E[I]&=\Pi_0\begin{bmatrix}c\\c-1\\\vdots\\0\end{bmatrix}+\Pi_1\begin{bmatrix}c-1\\c-2\\\vdots\\0\end{bmatrix}+\Pi_2\begin{bmatrix}c-2\\c-3\\\vdots\\0\end{bmatrix}+\ldots+\Pi_{c-1}\begin{bmatrix}1\\0\\\vdots\\0\end{bmatrix}\\&=\Pi_c\Phi_1\mathbf{v}_1+\Pi_c\Phi_2\mathbf{v}_2+\ldots+\Pi_c\Phi_c\mathbf{v}_c=\Pi_c\sum_{i=1}^{c}\Phi_i\mathbf{v}_i,\end{aligned} \tag{4.15}$$

$$E[B]=c-E[V]-E[I], \tag{4.16}$$

where

$$\mathbf{u}_i=[\overbrace{0,\ldots,0}^{\#=c+1-i},\overbrace{1,2,\ldots,i}^{\#=i}]^{\mathrm{T}}\quad\text{and}\quad\mathbf{v}_i=[\overbrace{c-i,c-i-1,\ldots,1,0}^{\#\,=\,c-i+1}\overbrace{0,\ldots,0}^{\#\,=\,i}]^{\mathrm{T}}$$

are column vector with dimensional $(c+1)$. To understand how system performance measures (such as L_s and $E[B]$) listed above vary with λ, μ and η, we now perform some numerical investigation to the measures based on changing the value of system parameters. For computation, we let p=0.5. The numerical results of L_s are obtained by considering the following three cases with different values of c.

Case 1. μ = 5.5, η=2.0, vary λ from 2.0 to 5.0.

Case 2. λ = 2.0, η=2.0, vary μ from 2.5 to 5.5.

Case 3. λ = 2.0, μ=3.0, vary η from 1.0 to 4.0.

Results of L_s are depicted in Figures 4.1-4.3 for Case 1-3, respectively. Figure 4.1 reveals that (i) L_s increases quickly as λ increases for c=1, and (ii) L_s slightly increases as λ increases for $c\geq 2$. We observes from Figure 4.2 that (i) L_s drastically decreases as μ increases for c=1, and (ii) L_s slightly decreases as μ increases for $c\geq 2$. One sees from Figure 4.3 that L_s slightly decreases as η increases. We also interest in the effect of different parameters on the expected number of busy servers ($E[B]$). The following three cases are considered:

Case 4. $E[B]$ versus λ from 2.0 to 5.0 when μ=5.5 and η=2.0.

Case 5. $E[B]$ versus μ from 2.5 to 5.5 when λ=2.0 and η=2.0.

Case 6. $E[B]$ versus η from 1.0 to 4.0 when λ=2.0 and μ=3.0.

The numerical illustrations of the expected number of busy servers are graphically presented in Figures 4.4-4.6 for Case 4-6, respectively. We observe from Figures 4.4-4.5 that $E[B]$ increases as λ increases or μ decreases. Figure 4.6 reports $E[B]$ is a constant is independent of η. From the investigation, it is interesting that $E[B]$ nearly equals to λ/μ. However, it is very difficult to proof the results. In the next section, we will provide the proof of single server case ($c=1$).

4.4.1. Special case of single server

As a particular case, the M/M/1/MBSV queueing system, in which the server may take a vacation if server is free at service completion instant, steady-state equations in states (0,0), (0,1), and (1,0) are given by:

$$\begin{aligned} \lambda P_0(0) &= q\mu P_0(1) + \eta P_1(0), \\ (\lambda+\eta)P_1(0) &= p\mu P_0(1), \end{aligned} \tag{4.17}$$

which implies

$$\lambda[P_0(0)+P_1(0)] = \mu P_0(1). \tag{4.18}$$

For the single server case, the sub-matrices are as following:

$$\mathbf{B} = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}, \ \mathbf{A}_c = \begin{bmatrix} -(\lambda+\mu) & 0 \\ \eta & -(\lambda+\eta) \end{bmatrix}, \text{ and } \mathbf{C}_{c+1} = \begin{bmatrix} \mu & 0 \\ 0 & 0 \end{bmatrix}.$$

Substituting $\mathbf{B}$, $\mathbf{A}_c$, $\mathbf{C}_{c+1}$ into $\mathbf{B}+\mathbf{R}\mathbf{A}_c+\mathbf{R}^2\mathbf{C}_{c+1}=\mathbf{0}$ and solving the quadratic equation above, we have

$$\mathbf{R} = \begin{bmatrix} \frac{\lambda}{\mu} & 0 \\ \frac{\lambda}{\mu} & \frac{\lambda}{\lambda+\eta} \end{bmatrix}. \tag{4.19}$$

Also, equation (4.20) can be obtained from (4.6). For the case of single server, the steady-state distribution $\Pi_1 = [P_0(1), P_1(1)]$ satisfies $\Pi_0\mathbf{B}+\Pi_1\mathbf{A}_1+\Pi_1\mathbf{R}\mathbf{C}_2=\mathbf{0}$ as following

$$\begin{aligned} &[P_0(0), P_1(0)]\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} + [P_0(1), P_1(1)]\begin{bmatrix} -(\lambda+\mu) & 0 \\ \eta & -(\lambda+\eta) \end{bmatrix} \\ &\quad + [P_0(1), P_1(1)]\begin{bmatrix} \frac{\lambda}{\mu} & 0 \\ \frac{\lambda}{\mu} & \frac{\lambda}{\lambda+\eta} \end{bmatrix}\begin{bmatrix} \mu & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned} \tag{4.20}$$

which leads to

$$\frac{\lambda p\mu}{\lambda+\eta}P_0(1)=(\lambda+\eta)P_1(1). \tag{4.21}$$

Using the normalization condition (4.11) to obtain Π_1

$$[P_0(1),P_1(1)](\mathbf{I}-\mathbf{R})^{-1}\begin{bmatrix}1\\1\end{bmatrix}+[P_0(0),P_1(0)]\begin{bmatrix}1\\1\end{bmatrix}=1. \tag{4.22}$$

Substituting (4.18), (4.19) and (4.21) into (4.22), we get $P_0(1)$ and $P_1(1)$ as follows

$$P_0(1)=\frac{\lambda(\lambda+\eta)(\mu-\lambda)\eta}{(p\lambda^2+\eta\lambda+\eta^2)\mu^2}, \tag{4.23}$$

and

$$P_1(1)=\frac{\lambda^2 p\eta(\mu-\lambda)}{(\lambda+\eta)(p\lambda^2+\eta\lambda+\eta^2)\mu}. \tag{4.24}$$

After the gaining of Π_1, the rest steady-state probability vectors $\Pi_2,\Pi_3,\Pi_4,...$ can be obtained recursively with $\Pi_2=\Pi_1\mathbf{R}$, $\Pi_3=\Pi_2\mathbf{R}$, …, and so on. The expected number of busy servers is

$$\begin{aligned}E[B]&=[P_0(1),P_1(1)](\mathbf{I}-\mathbf{R})^{-1}\begin{bmatrix}1\\0\end{bmatrix}=[P_0(1),\frac{\lambda p\mu}{(\lambda+\eta)^2}P_0(1)]\begin{bmatrix}\frac{\mu}{\mu-\lambda} & 0\\ \frac{\lambda(\mu+\eta)}{\eta(\mu-\lambda)} & \frac{\lambda+\eta}{\eta}\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}\\ &=[P_0(1),\frac{\lambda p\mu}{(\lambda+\eta)^2}P_0(1)]\begin{bmatrix}\frac{\mu}{\mu-\lambda}\\ \frac{\lambda(\mu+\eta)}{\eta(\mu-\lambda)}\end{bmatrix}=\frac{\mu(p\lambda^2+\eta\lambda+\eta^2)}{\eta(\lambda+\eta)(\mu-\lambda)}P_0(1)=\rho.\end{aligned} \tag{4.25}$$

It is interest that the result of (4.25) for the M/M/1/MBSV queueing system, in which the server may take a vacation if server is free at service completion instant, is the same as that of the ordinary M/M/1 queue. Furthermore, the steady-state equation (4.17) implies that the probability of system empty while the server is not on vacation, is given by

$$P_0(0)=\frac{(q\lambda+\eta)(\mu-\lambda)\eta}{(p\lambda^2+\eta\lambda+\eta^2)\mu}, \text{ and } P_1(0)=\frac{p(\mu-\lambda)\eta}{(p\lambda^2+\eta\lambda+\eta^2)\mu}. \tag{4.26}$$

As $p=0$ ($q=1$) or $\eta\to\infty$, $P_0(0)=1-\rho$, $P_0(1)=\rho(1-\rho)$, and $\rho P_0(0)=P_0(1)$, which are consistent with the result of the ordinary M/M/1 queue.

4.5 Numerical Results

In this section, we construct the total expected cost function per unit time based on the system performance measures for the M/M/*c*/MBSV queueing system, in which the number of servers (c) is a discrete decision variable, and the service rate (μ) and the vacation rate (η) are continuous decision variables. Our main objective is to find the optimum number of servers c^*, and the optimum values of service rate and vacation rate (μ^*, η^*) simultaneously to minimum the cost function. Let us define the following cost elements:

$C_h \equiv$ holding cost per unit time per customer present in the system,
$C_s \equiv$ cost per unit time of providing an service rate μ,
$C_v \equiv$ cost per unit time when one server is on vacation,
$C_r \equiv$ cost per unit time of providing an vacation rate η,
$C_p \equiv$ fixed cost for purchasing one server.

Using the definition of the cost parameters listed above, the total expected cost function per unit time is given by:

$$F(c,\mu,\eta) = C_h L_s + C_s \mu + C_v E[V] + C_r \eta + C_p c, \tag{4.27}$$

where L_s and $E[V]$ are defined previously. The analytic study of the optimization behavior of the expected cost function would have been an arduous task to undertake since the decision variables appear in an expression which is a highly nonlinear and complex and non-linear in terms of (c,μ,η). We firstly use the Quasi-Newton method to find the optimal value of continuous variable (μ,η), say (μ^*,η^*), and then use direct search method to search the optimal value of discrete variable c, say c^*. For practice use, the number of servers is bounded by a positive integer $c_U \geq 1$. We want to find the joint optimal value (μ^*, η^*) for each given c in the feasible set $\{1,2,\ldots,c_U\}$. The cost minimization problem can be illustrated mathematically as

$$F(c,\mu^*,\eta^*) = \min_{(\mu,\eta)\ \text{s.t.(4.3)}} \{F(c,\mu,\eta)|c\}, \quad c = 1,2,...,c_U. \tag{4.28}$$

For a fixed *c*, Quasi-Newton method is employed to search (μ,η) until the minimum value of $F(c,\mu,\eta)$ is achieved, say $F(c,\mu^*,\eta^*)$. To demonstrate the valid and the procedure of optimization solution, we perform some examples shown in Table 1 by considering the following cost parameters as

C_h=\$90/customer/unit time, C_s=\$15 /unit time,

C_v=\$30/server, C_r=\$45 /unit time, and C_p=\$120/server

From Table 4.1, we can see that the minimum expected cost per unit time of **838.457** is achieved at $(\mu^*,\eta^*)=$(**17.5903**, **4.30120**) by using 6 iterations, which is $c=1$ based on Case (i) with initial value $(\mu,\eta)=$(15, 2.0). Based on Case (ii), the initial value $(c,\mu,\eta)=$(3, 10, 2) and the minimum expected cost per unit time of **935.612** is achieved at $(\mu^*,\eta^*)=$(**15.2171**, **2.74098**) by using 6 iterations. After we obtain the joint optimal value (μ^*,η^*) of the continuous variable (μ,η), we will use the direct search method to obtain the optimal c such that the expected cost function $F(c,\mu^*,\eta^*)$ attains a minimum, say $F(c^*,\mu^*,\eta^*)$. Therefore, the cost minimization problem can be illustrated mathematically as

$$F(c^*,\mu^*,\eta^*)=\min_{c\in\{1,2,\ldots,c_U\}}\left\{F(c,\mu^*,\eta^*)\right\}. \tag{4.29}$$

The procedure to find the optimal solution is described in the following. A numerical example is shown in Table 4.2 based on (i) $(\lambda, p)=$(15, 0.5) and (ii) $(\lambda, p)=$(20, 0.8). It is noted that the optimal value $(c^*,\mu^*,\eta^*)=$(2, 15.284, 3.7983) and the corresponding minimum cost F^* = 895.4944 for Case (i). For Case (ii), $(c^*,\mu^*,\eta^*)=$(2, 18.731, 4.8242) and F^* = 1071.252 are optimal.

Finally, we perform a sensitivity investigation to the optimal values (c^*,μ^*,η^*). For various values of λ and p, the minimum expected cost $F(c^*,\mu^*,\eta^*)$ and the system performance measures L_s, and $E[V]$ at the optimum values (c^*,μ^*,η^*) are shown in Table 4.3. From the Table, it is seen that (i) c^* is insensitive to λ or p; (ii) μ^* increases as λ increases; and (iii) η^* increases as λ or p increases. Moreover, the minimum expected cost increases $F(c^*,\mu^*,\eta^*)$ as λ or p increases.

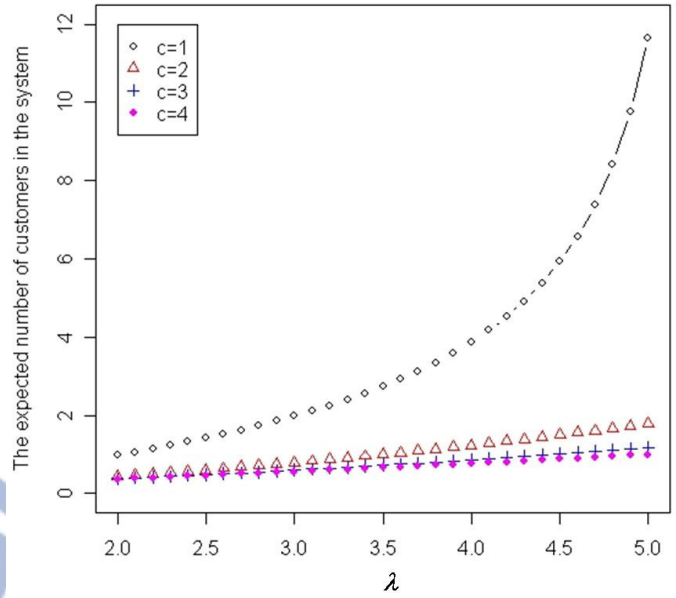

Figure 4.1. The effect of λ on the expected number of customers in the system.

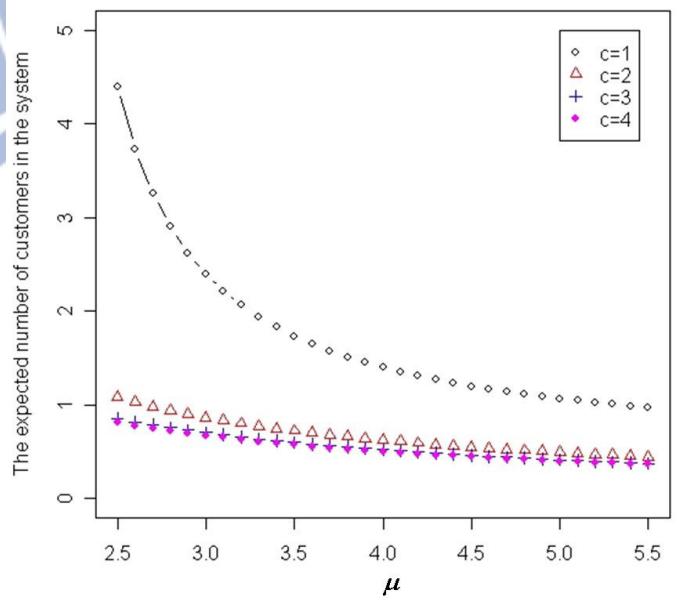

Figure 4.2. The effect of μ on the expected number of customers in the system.

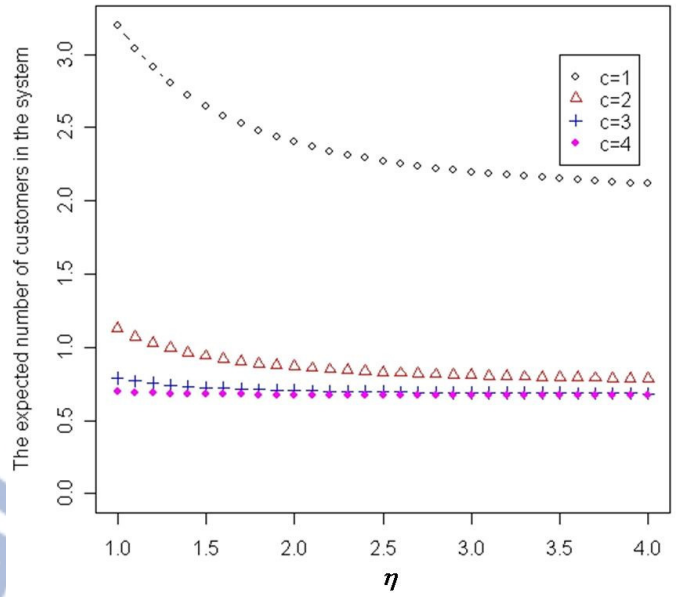

Figure 4.3. The effect of η on the expected number of customers in the system.

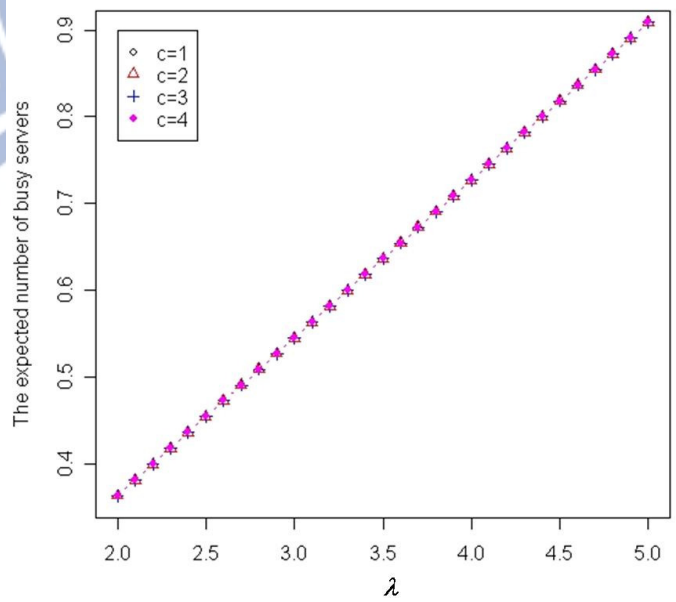

Figure 4.4. The expected number of busy servers versus λ.

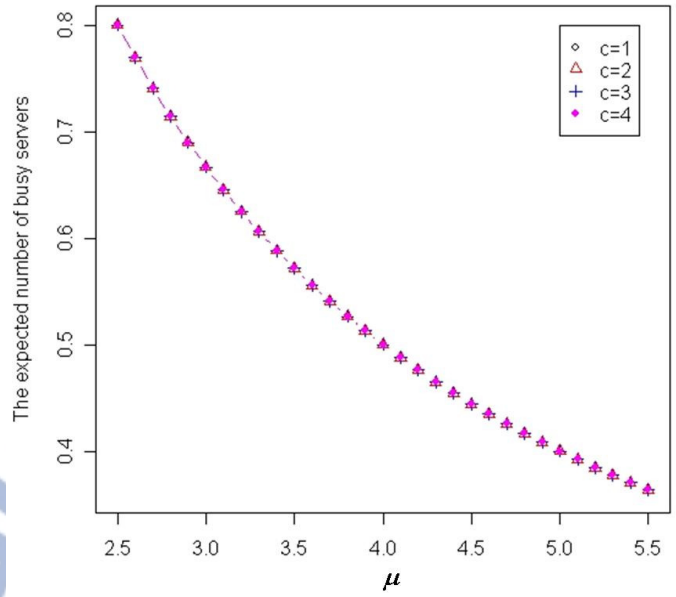

Figure 4.5. The expected number of busy servers versus μ.

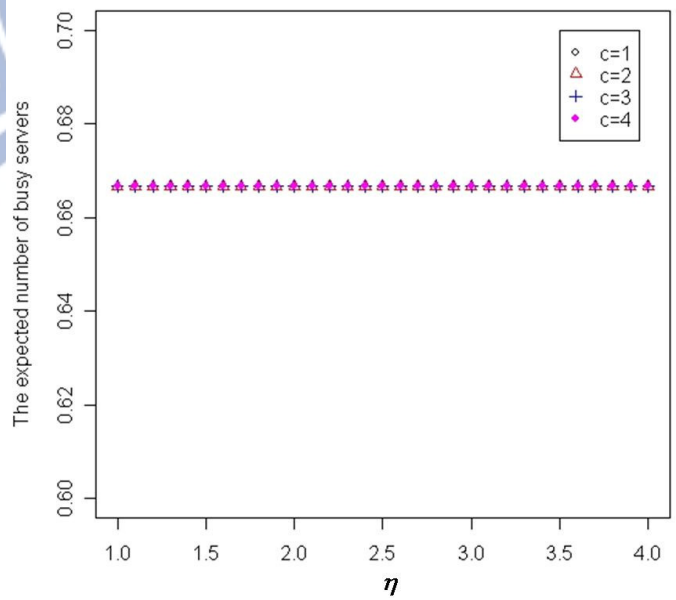

Figure 4.6. The expected number of busy servers versus η.

Table 4.1. The illustration of the implement process of Quasi-Newton method

Case (i): $(\lambda, p) = (10, 0.5)$ with initial value $(c, \mu, \eta) = (1, 15, 2.0)$

Iterations	0	1	2	3	4	5	6
$F(c,\mu,\eta)$	987.973	882.065	845.430	838.786	838.458	838.457	**838.457**
μ	15	16.4035	17.3194	17.5741	17.5901	17.5903	**17.5903**
η	2.0	2.78381	3.59146	4.13419	4.29150	4.30117	**4.30120**
$\frac{\partial F}{\partial \mu}$	-19.9189	16.4035	-1.11143	-0.05537	-0.00008	1.57×10^{-8}	-8×10^{-11}
$\frac{\partial F}{\partial \eta}$	-176.914	2.78381	-21.1504	-4.00835	-0.22011	-0.00075	-8.5×10^{-9}
L_s	6.05405	4.24438	3.28115	2.89676	2.81312	2.80833	2.80831
Hessian	$\begin{bmatrix} 14.26 & -0.114 \\ -0.114 & 225.9 \end{bmatrix}$	$\begin{bmatrix} 6.754 & -0.086 \\ -0.086 & 83.88 \end{bmatrix}$	$\begin{bmatrix} 4.511 & -0.069 \\ -0.069 & 39.00 \end{bmatrix}$	$\begin{bmatrix} 4.070 & -0.062 \\ -0.062 & 25.48 \end{bmatrix}$	$\begin{bmatrix} 4.046 & -0.06 \\ -0.06 & 22.76 \end{bmatrix}$	$\begin{bmatrix} 4.046 & -0.06 \\ -0.06 & 22.60 \end{bmatrix}$	$\begin{bmatrix} 4.046 & -0.06 \\ -0.06 & 22.60 \end{bmatrix}$

Case (ii): $(\lambda, p) = (20, 0.2)$ with initial value $(c, \mu, \eta) = (3, 10, 2)$

Iterations	0	1	2	3	4	5	6
$F(c,\mu,\eta)$	1052.33	971.631	942.421	936.060	935.615	935.612	**935.612**
μ	10	11.5153	13.2741	14.6682	15.1728	15.2168	**15.2171**
η	2.0	2.46623	2.71924	2.75081	2.74176	2.74098	**2.74098**
$\frac{\partial F}{\partial \mu}$	-59.8568	-23.5812	13.2741	-1.69159	-0.12620	-0.00083	-4.8×10^{-8}
$\frac{\partial F}{\partial \eta}$	-77.6947	-24.0490	2.71924	-0.59183	-0.04460	-0.00031	4.16×10^{-9}
L_s	4.82721	3.42012	2.65292	2.31515	2.22369	2.21614	2.21609
Hessian	$\begin{bmatrix} 36.25 & 10.56 \\ 10.56 & 132.3 \end{bmatrix}$	$\begin{bmatrix} 12.73 & 4.723 \\ 4.723 & 62.22 \end{bmatrix}$	$\begin{bmatrix} 5.547 & 2.580 \\ 2.580 & 41.83 \end{bmatrix}$	$\begin{bmatrix} 3.386 & 1.861 \\ 1.861 & 38.37 \end{bmatrix}$	$\begin{bmatrix} 2.900 & 1.686 \\ 1.686 & 38.22 \end{bmatrix}$	$\begin{bmatrix} 2.862 & 1.671 \\ 1.671 & 38.21 \end{bmatrix}$	$\begin{bmatrix} 2.862 & 1.671 \\ 1.671 & 38.21 \end{bmatrix}$

Table 4.2. The optimal value (μ^*, η^*) and the corresponding minimum expected cost

(i) $(\lambda, p) = (15, 0.5)$

c	**Initial Value**	**Coverage Value** (μ^*, η^*)	**Iteration**	**Cost***
c=1	[20, 2.0]	[24.32507, 5.332980]	7	1052.297
c=2	[15, 2.0]	[15.28433, 3.798293]	6	895.4944
c=3	[10, 2.0]	[12.37270, 3.088068]	6	920.8427
c=4	[10, 2.0]	[11.00938, 2.679454]	5	998.4310
c=5	[10, 2.0]	[10.26962, 2.428360]	5	1098.187

(ii) $(\lambda, p) = (20, 0.8)$

c	**Initial Value**	**Coverage Value** (μ^*, η^*)	**Iteration**	**Cost***
c=1	[25, 5.0]	[30.75986, 6.423140]	6	1288.713
c=2	[20, 3.0]	[18.73113, 4.824175]	6	1071.252
c=3	[15, 2.0]	[14.85998, 4.032956]	6	1073.578
c=4	[10, 2.0]	[13.05122, 3.560957]	6	1137.429
c=5	[10, 2.0]	[12.06278, 3.260737]	6	1232.625

Table 4.3. The optimal value (c^*, μ^*, η^*) and F^* for various value of λ and p.

(λ, p)	(5, 0.2)	(10,0.2)	(20, 0.2)	(5, 0.8)	(10, 0.8)	(20, 0.8)
c^*	2	2	2	2	2	2
(μ^*, η^*)	[7.249477, 1.471333]	[11.60659, 2.295007]	[19.16225, 3.550663]	[7.091449, 2.326386]	[11.32231, 3.368702]	[18.73113, 4.824175]
$F(c^*, \mu^*, \eta^*)$	532.099	685.935	932.038	610.522	792.191	1071.252
L_s	1.154063	1.717796	2.565803	1.481779	2.275863	3.436747
$E[V]$	0.442712	0.465296	0.463387	0.870082	0.864552	0.796331

Chapter 5

M/M/*c* Retrial Queue with Bernoulli Vacation Policy

The multi-server retrial queue with a Bernoulli single vacation policy is considered in this chapter. Servers may take a vacation at the completion of service with probability p or continuous to serve the next customer with probability $1-p$. Such as the customer service department, the receptionist may take a rest after completing a service for a customer. For the telephone communication system, the servers and customers have no information for each other. Consequently, the retrial behavior of customers should be considered.

When the servers complete the vacation period, they stay idly for the next new arrival or serve the customers in the system, if any. That is, the single vacation policy. The stability condition is developed explicitly. For this queueing model, it is rather difficult to obtain the close-form of steady-sate probability. Hence, we use matrix-analytical method to solve the steady-state solution recursively. Conveniently, we represent this multi-server system with Bernoulli single vacation policy as M/M/*c*/BSV retrial queue.

This chapter is organized as follows. Section 5.1 gives some basic assumptions of the queue under study and notations. We develop the state-transition matrix, the stability condition, and the recursive method to obtain the steady-state solution in Section 5.2. Section 5.3 provides some important system performance measures. Finally, Section 5.4 presents the numerical results and several examples to illustrate the optimization procedures.

5.1 Assumptions and Notations

An M/M/*c*/BSV retrial queue is investigated. Primary customers arriving as a Poisson process with parameter λ. An arriving primary customer finding any available servers will get service immediately. Otherwise, he joins the orbit and attempts to enter the system later. There are c channels (servers) that provide service for the customers. The service times are assumed exponentially distribution with rate μ. Each server can serve one and only one customer at a time and the service is independent of the arrival process. At the service completion instant of as server, it may take a vacation of random interval with probability p or wait idly in the system for the next new arrival with probability q ($q=1-p$). The vacation times follow an

exponentially distribution with parameter η. Furthermore, the inter-retrial time of each customer staying in the orbit is assumed exponentially distributed with parameter σ. Upon requesting service from the orbit, customer who finds all c servers busy always rejoins the orbit; this manner continues until he is eventually served. It is assumed that the number of customers in the orbit that are allowed to conduct retrials have an upper bound N (see Neuts and Rao [50] and Artalejo and Pozo [7]). Moreover, the process of primary arrivals, service times and inter-retrial times are assumed mutually independent.

For an M/M/c/BSV retrial queue, the state of the system can be described by the pair (i, j, k), $i = 0,1,2,...,c$, $j = 0,1,2,...$, $k = 0,1,2,...,c-i$, where i denotes the number of busy server, j is the number of customers in orbit (sources of repeated demands) and k denotes the number of vacation servers. According to system assumptions, the number of customers in orbit allowed to conduct retrials is restricted to an appropriate number N ($N > c$), so the retrial rate is $\sigma_j = \min\{j, N\}\sigma$, $j \geq 0$ and one server will go on vacation with probability p ($0 \leq p \leq 1$) or resumes service with probability $q = 1 - p$ at a service completion instant. The customers upon the server will get services immediately as $i + k < c$. The new arriving customer who find all c servers busy ($i + k \geq c$) always rejoins the retrial group (orbit).

In steady-state, the steady-state probability is defined as

$P_{i,j}^{k} \equiv$ probability that there are i busy servers and j customers in orbit and k vacation servers, where $0 \leq i + k \leq c$ and $j = 0,1,2,...$.

In this chapter, the following notations and probabilities are used.

λ — mean arrival rate

μ — mean service rate

p — probability that a server may opt for Bernoulli vacation

η — vacation rate

σ — mean retrial rate

c — number of channels (servers)

$\mathbf{\Pi}$ — steady-state probability vector

$\mathbf{Q}$ — infinitesimal generator

$\mathbf{I}$ — identity matrix

$\mathbf{e}$ — identity column vector (a column vector with all elements equal to 1)

$\mathbf{F}$ — irreducible generator

$\mathbf{x}$ — invariant probability

P_F — probability that all servers are busy

$\mathbf{R}$ — rate matrix

$E[B]$ — expected number of customers in the *FES* channel

$E[V]$ — expected number of customers in the *SOS* channel

$E[Orbit]$ — expected number of idle servers

σ_1^* — the overall rate of retrials

σ_2^* — the rate of retrials that are successful

FR — the fraction of retrials that are successful

$E(T)$ — mean busy period

P_V — vain retrials

F — cost function

5.2 M/M/c Retrial Queue with Bernoulli Vacation

This paper consider a M/M/*c* retrial queue in which primary customers arriving to a Poisson process with parameter λ. An arriving primary customer finding one or more servers available (free) obtains service immediately. On the other hand, if the primary customer who finds all servers busy, he joins the orbit and tries to get the service later on. There are c channels (servers) that provide service for the arrivals, in which the service times are assumed to be exponentially distributed with mean $1/\mu$. Each server can serve only one customer at a time, and that the service is independent of the arrival of the customers. At each service completion instant of a server, the server may take a vacation of random length with probability p or wait to serve the next arrival with probability $q(1-p)$. The vacation times follow an exponentially distributed with a parameter η. Furthermore, each customer staying in the orbit makes the repeated attempts in random intervals having length exponentially distributed with parameter σ, independently of the other customers. Upon requesting service from the orbit, customer who finds all c servers busy always rejoins the orbit; this manner continues until he is eventually served. It is assumed that there exists an upper bound *N* on the number of customers in the orbit that are allowed to conduct retrials (see Neuts and Rao (1990) and Artalejo and Pozo (2002)). This implies that the probability of a repeated attempt during $(t,t+dt)$, given that j customers in the orbit at time *t*, is $\sigma_j dt + o(dt)$, where $\sigma_j = \min\{j,N\}\sigma$. Moreover, the process of primary arrivals, service times and inter-retrial times are assumed mutually

independent. Conveniently, we represent this multi-server system with Bernoulli vacation as M/M/*c*/BSV retrial queue.

5.3 Steady-state Results

The infinitesimal generator $\mathbf{Q}$ that describing the M/M/*c*/BSV retrial queueing system is

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \mathbf{C}_N & \mathbf{A}_{N-1} & \mathbf{B} & \\ & & & & \mathbf{C}_{N-1} & \mathbf{A}_N & \mathbf{B} \\ & & & & & \mathbf{C}_N & \mathbf{A}_N & \mathbf{B} \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{5.1}$$

Let $\mathbf{\Pi} = [\mathbf{\Pi}_0, \mathbf{\Pi}_1, \mathbf{\Pi}_2, ...]$ with sub-vector $\mathbf{\Pi}_i = [P_{0,i}^0, P_{1,i}^0, ..., P_{c,i}^0, P_{0,i}^1, P_{1,i}^1, ..., P_{c-1,i}^1, P_{0,i}^0, ..., P_{0,i}^{c-1}, P_{1,i}^{c-1}, P_{0,i}^c]$, $i = 0,1,2,...$ be the unique solution to $\mathbf{\Pi Q} = \mathbf{0}$ and $\mathbf{\Pi e} = 1$, where $\mathbf{e}$ is a column vector with all elements equal to 1. The entries $\mathbf{B}$, $\mathbf{A}_j (j \geq 0)$, and $\mathbf{C}_j (j \geq 1)$ are block-diagonal matrices of order $(c+1)(c+2)/2$ defined by

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_0 & & & & \\ & \mathbf{b}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{b}_{c-1} & \\ & & & & \mathbf{b}_c \end{bmatrix} \quad \text{and} \quad \mathbf{C}_j = \begin{bmatrix} \mathbf{c}_0 & & & & \\ & \mathbf{c}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{c}_{c-1} & \\ & & & & \mathbf{c}_c \end{bmatrix}, \quad j = 1,2,...,$$

where sub-matrices $\mathbf{b}_i$ and $\mathbf{c}_i$ are $(c+1-i)\times(c+1-i)$ square matrices with elements

$$\begin{cases} \mathbf{b}_i[c+1-i, c+1-i] = \lambda, \\ 0 \qquad\qquad\qquad\quad e.w, \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{c}_i[k, k+1] = \sigma_j, \ 1 \leq k \leq c-i, \\ 0 \qquad\qquad\qquad\quad e.w, \end{cases}$$

$$\mathbf{A}_j = \begin{bmatrix} \mathbf{Y}_j^0 & \mathbf{X}^0 & & & & & \\ \mathbf{Z}^1 & \mathbf{Y}_j^1 & \mathbf{X}^1 & & & & \\ & \mathbf{Z}^2 & \mathbf{Y}_j^2 & \mathbf{X}^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{Z}^{c-1} & \mathbf{Y}_j^{c-1} & \mathbf{X}^{c-1} \\ & & & & & \mathbf{Z}^c & \mathbf{Y}_j^c \end{bmatrix}, \quad j = 0,1,2,...,$$

where $\mathbf{X}^i$ is a $(c+1-i)\times(c-i)$ matrix with $\mathbf{X}^i[k+1,k] = kp\mu,\ 1 \le k \le c-i$, $\mathbf{Z}^i$ is a $(c-i)\times(c+1-i)$ matrix with $\mathbf{Z}^i[k,k] = i\eta,\ 1 \le k \le c-i$, and $\mathbf{Y}_j^i$ is a square matrix of order $(c+1-i)$ with elements

$$\begin{cases} \mathbf{Y}_j^i[k,k+1] = \lambda,\ 1 \le k \le c-i, \\ \mathbf{Y}_j^i[k+1,k] = k(1-p)\mu,\ 1 \le k \le c-i, \\ \mathbf{Y}_j^i[1,1] = -[\lambda + (i-1)\eta + \sigma_j], \\ \mathbf{Y}_j^i[k,k] = -[\lambda + (k+1)\mu + i\eta + \sigma_j],\ 2 \le k \le c-i, \\ \mathbf{Y}_j^i[c+1-i, c+1-i] = -[\lambda + i\eta + (c-i)\mu]. \end{cases}$$

5.3.1. Stability condition

By Theorem 3.1.1 of Neuts [49] that the steady-state probability vector exists if and only if

$$\mathbf{xBe} < \mathbf{xC}_N\mathbf{e}, \tag{5.2}$$

where $\mathbf{x}$ is the invariant probability of the matrix $\mathbf{F} = \mathbf{C}_N + \mathbf{A}_N + \mathbf{B}$. $\mathbf{x}$ satisfies $\mathbf{xF} = \mathbf{0}$ and $\mathbf{xe} = 1$. First we solve $\mathbf{xF} = \mathbf{0}$, where $\mathbf{x} = [x_0^0, x_1^0 ..., x_c^0, ..., x_0^{c-1}, x_1^{c-1}, x_0^c]$. We can get following $(c+1)(c+2)/2$ equations :

For $k = 0$,

$$-(\lambda + N\sigma)x_0^0 + q\mu x_1^0 + \eta x_0^1 = 0, \tag{5.3a}$$

$$(\lambda + N\sigma)x_{i-1}^0 - (\lambda + i\mu + N\sigma)x_i^0 + (i+1)q\mu x_{i+1}^0 + \eta x_i^1 = 0,\ 1 \le i \le c-1, \tag{5.3b}$$

$$(\lambda + N\sigma)x_{c-1}^0 - c\mu x_c^0 = 0. \tag{5.3c}$$

For $1 \le k \le c-1$,

$$p\mu x_1^{k-1} - (\lambda + N\sigma + k\eta)x_0^k + q\mu x_1^k + (k+1)\eta x_0^{k+1} = 0, \tag{5.4a}$$

$$(\lambda+N\sigma)x_{i-2}^{k}+ip\mu x_{i}^{k-1}-[\lambda+N\sigma+(i-1)\mu+k\eta]x_{i-1}^{k} +iq\mu x_{i}^{k}+(k+1)\eta x_{i-1}^{k+1}=0,\ 2\le i\le c-k, \tag{5.4b}$$

$$(c+1-k)\,p\mu x_{c+1-k}^{k-1}+(\lambda+N\sigma)x_{c-k-1}^{k}-[(c-k)\mu+k\eta]x_{c-k}^{k}=0. \tag{5.4c}$$

For $k=c$,

$$p\mu x_{1}^{c-1}-c\eta x_{0}^{c}=0. \tag{5.5}$$

Using a effective Maple software to solve equations (5.3a)-(5.4c), it derive the following results

$$x_{i}^{k}=\frac{c!\eta^{c-k}}{i!k!(\lambda+N\sigma)^{c-i-k}\mu^{i}p^{c-k}}x_{0}^{c},\ 0\le i+k\le c. \tag{5.6}$$

Then using the normalization condition $\mathbf{xe}=1$, x_{0}^{c} can be determined as

$$x_{0}^{c}=\left[\sum_{k=0}^{c}\sum_{i=0}^{c-k}\frac{c!\eta^{c-k}}{i!k!(\lambda+N\sigma)^{c-i-k}\mu^{i}p^{c-k}}\right]^{-1}. \tag{5.7}$$

Substituting $\mathbf{B}$ and $\mathbf{C}_N$ into equation (5.2) and doing some routine manipulations, then we have

$$N\sigma(1-P_F)>\lambda P_F, \tag{5.8}$$

where

$$\begin{aligned}P_F&=\sum_{i=0}^{c}x_{i}^{c-i}=\sum_{i=0}^{c}\frac{c!\eta^{i}}{i!(c-i)!\mu^{i}p^{i}}x_{0}^{c}\\&=\left(1+\frac{\eta}{p\mu}\right)^{c}\left[\sum_{k=0}^{c}\sum_{i=0}^{c-k}\frac{c!\eta^{c-k}}{i!k!(\lambda+N\sigma)^{c-i-k}\mu^{i}p^{c-k}}\right]^{-1},\end{aligned} \tag{5.9}$$

which is referred to the probability that all normal working (non-vacation) server are busy (i.e. $i+k=c$). That is, the system would be stable if the expected successful retrial rate is greater then the expected arrival rate of "orbit".

5.3.2. Rate matrix

By the matrix-geometric property, it is noted that the steady-state probability vector $\mathbf{\Pi}=[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,\mathbf{\Pi}_3,...]$ has the following properties

$$\mathbf{\Pi}_{N+k}=\mathbf{\Pi}_{N}\mathbf{R}^{k},\ \text{for}\ \ k\ge 1. \tag{5.10}$$

The matrix $\mathbf{R}$ is the unique non-negative solution with spectral radius less than one of the equation

$$\mathbf{B}+\mathbf{R}\mathbf{A}_N+\mathbf{R}^2\mathbf{C}_N=\mathbf{0}. \tag{5.11}$$

From Neuts [49] and Latouche and Ramaswami [41], it is known that $\mathbf{R}$ is given by $\lim_{n\to\infty}\mathbf{R}_n$, where the sequence $\{\mathbf{R}_n\}$ is defined by

$$\mathbf{R}_0=\mathbf{0}, \text{ and } \mathbf{R}_{n+1}=-\mathbf{B}\mathbf{A}_N^{-1}-\mathbf{R}_n^2\mathbf{C}_N\mathbf{A}_N^{-1}, \text{ for } n\geq 0. \tag{5.12}$$

The sequence $\{\mathbf{R}_n\}$ is monotone so that $\mathbf{R}$ could be evaluated from (5.12) by successive substitutions.

After the development of rate matrix, the stationary probability vector Π exists under the stability condition. We deal with the steady-state equations by using matrix technique. The steady-state equations are given by

$$\mathbf{\Pi}_0\mathbf{A}_0+\mathbf{\Pi}_1\mathbf{C}_1=\mathbf{0}, \tag{5.13a}$$

$$\mathbf{\Pi}_{i-1}\mathbf{B}+\mathbf{\Pi}_i\mathbf{A}_i+\mathbf{\Pi}_{i+1}\mathbf{C}_{i+1}=\mathbf{0}, \ 1\leq i\leq N-1, \tag{5.13b}$$

$$\mathbf{\Pi}_{N-1}\mathbf{B}+\mathbf{\Pi}_N\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N=\mathbf{0}, \tag{5.13c}$$

$$\mathbf{\Pi}_N\mathbf{R}^{i-1-N}\mathbf{B}+\mathbf{\Pi}_N\mathbf{R}^{i-N}\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}^{i+1-N}\mathbf{C}_N=\mathbf{0}, \ N+1\leq i, \tag{5.13d}$$

$$\sum_{i=0}^{\infty}\mathbf{\Pi}_i\mathbf{e}=1. \tag{5.14}$$

After doing some routine manipulations to equation (5.13a)-(5.13c) recursively, we have

$$\begin{aligned}\mathbf{\Pi}_0&=\mathbf{\Pi}_1\mathbf{C}_1(-\mathbf{A}_0)^{-1}=\mathbf{\Pi}_1\phi_1,\\ \mathbf{\Pi}_{i-1}&=\mathbf{\Pi}_i\mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1}=\mathbf{\Pi}_i\phi_i, \ 2\leq i\leq N,\end{aligned} \tag{5.15}$$

and

$$\mathbf{\Pi}_N\phi_N\mathbf{B}+\mathbf{\Pi}_N\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N=\mathbf{0}. \tag{5.16}$$

Consequently, $\mathbf{\Pi}_i$ $(0\leq i\leq N-1)$ in equation (5.15) can be written in terms of $\mathbf{\Pi}_N$ as $\mathbf{\Pi}_0=\mathbf{\Pi}_N\prod_{i=N}^{1}\phi_i$, $\mathbf{\Pi}_1=\mathbf{\Pi}_N\prod_{i=N}^{2}\phi_i$, …, $\mathbf{\Pi}_{N-1}=\mathbf{\Pi}_N\prod_{i=N}^{N}\phi_i$ and the rest steady-state vector $[\mathbf{\Pi}_N,\mathbf{\Pi}_{N+1},\mathbf{\Pi}_{N+2},...]$ can be determined recursively as $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{R}^{i-N}$, for $i\geq N$. Therefore, once the steady-state probability $\mathbf{\Pi}_N$ is obtained, the steady-state solutions $[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,...,\mathbf{\Pi}_{N-1},\mathbf{\Pi}_N,\mathbf{\Pi}_{N+1},...]$ are

determined. The steady-state probability $\mathbf{\Pi}_N$ can be solved by equation (5.16) with the following normalization equation

$$\begin{aligned}\sum_{i=0}^{\infty}\mathbf{\Pi}_i\mathbf{e} &= [\mathbf{\Pi}_0+\mathbf{\Pi}_1+...+\mathbf{\Pi}_{N-1}+\mathbf{\Pi}_N+\mathbf{\Pi}_{N+1}+\mathbf{\Pi}_{N+2}+...]\mathbf{e}\\ &= [\mathbf{\Pi}_N\prod_{i=N}^{1}\phi_i+\mathbf{\Pi}_N\prod_{i=N}^{2}\phi_i+...+\mathbf{\Pi}_N\prod_{i=N}^{N}\phi_i+\mathbf{\Pi}_N+\mathbf{\Pi}_N\mathbf{R}+\mathbf{\Pi}_N\mathbf{R}^2+...]\mathbf{e} \quad (5.17)\\ &= \mathbf{\Pi}_N[\sum_{k=1}^{N}\prod_{i=N}^{k}\phi_i+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1.\end{aligned}$$

where $\mathbf{I}$ denotes the identity matrix with suitable size. Solving equations (5.16) and (5.17) in accordance with Cramer's rule, $\mathbf{\Pi}_N$ can be obtained. Then, the prior state probabilities $[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,...,\mathbf{\Pi}_{N-1}]$ are computed from (5.15) and $[\mathbf{\Pi}_{N+1},\mathbf{\Pi}_{N+2},\mathbf{\Pi}_{N+3},...]$ are gained by the formula $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{R}^{i-N}$, $i\geq N+1$. The solution procedure of steady-state probabilities is summarized as below:

Algorithm: Recursive Solver

Step 1. Set $\phi_1=\mathbf{C}_1(-\mathbf{A}_0)^{-1}$.

Step 2. For *i* from 2 to *N*, set $\phi_i=\mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1}$.

Step 3. For *k* from 1 to *N*, set $\mathbf{\Phi}_k=\prod_{i=N}^{k}\phi_i$.

Step 4. Solving $\mathbf{\Pi}_N\phi_N\mathbf{B}+\mathbf{\Pi}_N\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N=\mathbf{0}$, $\mathbf{\Pi}_N[\sum_{k=1}^{N}\mathbf{\Phi}_k+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1$ and obtain steady-state probability $\mathbf{\Pi}_N$.

Step 5. Construct steady-state probability $\mathbf{\Pi}_i$ as follows:

(a) if $0\leq i\leq N$, assign $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{\Phi}_{i+1}$,

(b) if $N\leq i$, assign $\mathbf{\Pi}_{i+1}=\mathbf{\Pi}_i\mathbf{R}$.

5.4 System Performance Measures

There are several system descriptors (system performance measures) of the M/M/*c*/BSV retrial queue, such as the expected number of busy servers (denoted by $E[B]$), the expected number of vacation servers (denoted by $E[V]$), and the expected number of customers in orbit (denoted by $E[Orbit]$) can be evaluated from the steady-state probabilities. The explicit expressions for $E[B]$, $E[V]$, and $E[Orbit]$ are given by

$$
\begin{aligned}
E[B] &= \sum_{j=0}^{\infty} \mathbf{\Pi}_j \mathbf{v} = \sum_{j=0}^{N-1} \mathbf{\Pi}_j \mathbf{v} + \mathbf{\Pi}_N \mathbf{v} + \mathbf{\Pi}_N \mathbf{R} \mathbf{v} + \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{v} + \ldots \\
&= \sum_{j=0}^{N-1} \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{v} + \mathbf{\Pi}_N \mathbf{v} + \mathbf{\Pi}_N \mathbf{R} \mathbf{v} + \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{v} + \ldots \\
&= \mathbf{\Pi}_N [\sum_{j=1}^{N} \mathbf{\Phi}_j + (\mathbf{I} - \mathbf{R})^{-1}] \mathbf{v}.
\end{aligned} \tag{5.18}
$$

$$
\begin{aligned}
E[V] &= \sum_{j=0}^{\infty} \mathbf{\Pi}_j \mathbf{u} = \sum_{j=0}^{N-1} \mathbf{\Pi}_j \mathbf{u} + \mathbf{\Pi}_N \mathbf{u} + \mathbf{\Pi}_N \mathbf{R} \mathbf{u} + \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{u} + \ldots \\
&= \sum_{j=0}^{N-1} \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{u} + \mathbf{\Pi}_N (\mathbf{I} - \mathbf{R})^{-1} \mathbf{u} \\
&= \mathbf{\Pi}_N [\sum_{j=1}^{N} \mathbf{\Phi}_j + (\mathbf{I} - \mathbf{R})^{-1}] \mathbf{u}.
\end{aligned} \tag{5.19}
$$

$$
\begin{aligned}
E[Orbit] &= \sum_{j=1}^{\infty} j \mathbf{\Pi}_j \mathbf{e} \\
&= \sum_{j=1}^{N-1} j \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{e} + N \mathbf{\Pi}_N \mathbf{e} + (N+1) \mathbf{\Pi}_N \mathbf{R} \mathbf{e} + (N+2) \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{e} + \ldots \\
&= \sum_{j=2}^{N} (j-1) \mathbf{\Pi}_N \mathbf{\Phi}_j \mathbf{e} + \mathbf{\Pi}_N [N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}] \mathbf{e} \\
&= \Pi_N [\sum_{j=2}^{N} (j-1) \mathbf{\Phi}_j + N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}] \mathbf{e},
\end{aligned} \tag{5.20}
$$

where

$$
\mathbf{v} = [\underbrace{0,1,\ldots,c}_{\#=c+1}, \underbrace{0,1,\ldots,c-1}_{\#=c}, \ldots, \underbrace{0,1}_{\#=2}, 0] \quad \text{and} \quad \mathbf{u} = [\underbrace{0,0,\ldots,0}_{\#=c+1}, \underbrace{1,1,\ldots,1}_{\#=c}, \ldots, \underbrace{c-1,c-1}_{\#=2}, c]
$$

are column vectors with dimension $(c+1)(c+2)/2$. For an M/M/*c*/BSV retrial queue, the numerical results of $E[Orbit]$ are obtained by considering the following four cases with different values of *c*

Case 1. N =30, $\lambda = 5$, $\eta = 10$, $p = 0.5$, $\sigma = 5$, vary μ from 10 to 15.

Case 2. N =30, $\lambda = 5$, $\mu = 10$, $p = 0.5$, $\sigma = 10$, vary η from 10 to 15.

Case 3. N =30, $\mu = 15$, $\eta = 15$, $p = 0.5$, $\sigma = 10$, vary λ from 5 to 10.

Case 4. N =30, λ =5, $\mu = 15$, $\eta = 15$, $p = 0.5$, vary σ from 10 to 15.

Results of $E[Orbit]$ are depicted in Figures 5.1-5.4 for Case 1-4, respectively. One sees from Figure 5.1 and Figure 5.2 that $E[Orbit]$ drastically decreases as μ or η increases for $c = 1$, while $E[Orbit]$ is not sensitive to μ or η for $c \geq 2$. It reveals from Figure 5.3 that $E[Orbit]$ increases violently as λ increases for $c = 1$ while $E[Orbit]$ slightly increases as λ increases for $c \geq 2$. Figure 5.4 reports that

$E[Orbit]$ decreases as σ increases for $c=1$, while $E[Orbit]$ is not sensitive to σ for $c \geq 2$. There are several general descriptors of retrial queues, some of which are listed below:

1. The overall rate of retrials

$$\begin{aligned}\sigma_1^* &= \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k = \sum_{j=1}^{N} j\sigma \mathbf{\Pi}_j \mathbf{e} + \sum_{j=N+1}^{\infty} N\sigma \mathbf{\Pi}_N \mathbf{R}^{j-N} \mathbf{e} \\ &= \sum_{j=1}^{N} j\sigma \mathbf{\Pi}_j \mathbf{e} + N\sigma \mathbf{\Pi}_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1} \mathbf{e} = \sigma[\sum_{j=1}^{N} j\mathbf{\Pi}_j + N\mathbf{\Pi}_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e} \\ &= \sigma \mathbf{\Pi}_N [\sum_{j=1}^{N-1} j\mathbf{\Phi}_{j+1} + N(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}.\end{aligned} \tag{5.21}$$

2. The rate of retrials that are successful

$$\sigma_2^* = \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k. \tag{5.22}$$

3. The fraction of retrials that are successful

$$FR = \frac{\sigma_2^*}{\sigma_1^*}. \tag{5.23}$$

4. The marginal distribution of the number of busy servers

$$\sum_{j=0}^{\infty} P_{i,j}^k, \quad 0 \leq i+k \leq c. \tag{5.24}$$

5. Busy period : The busy period T of a retrial queue is defined as the period that starts at the epoch when an arriving customer finds an empty system (all servers are idle and no customer in the orbit) and ends at the departure epoch at which the system is empty again.

   The mean busy period

$$E(T) = \frac{1}{\lambda}(\frac{1}{P_{0,0}^0} - 1) = \frac{1}{\lambda}(\frac{1}{\mathbf{\Pi}_N \mathbf{\Phi}_1[1]} - 1), \tag{5.25}$$

   where $\mathbf{\Pi}_N \mathbf{\Phi}_1[1]$ denotes the first element of $\mathbf{\Pi}_N \mathbf{\Phi}_1$.

6. Vain retrials : A vain retrial is an unsuccessful retrial when all servers are busy.
   The steady-state probability of vain retrial P_V

$$P_V = \frac{\sum_{j=1}^{\infty}\sum_{i+k=c} P_{i,j}^{k}}{\sum_{j=1}^{\infty}\sum_{k=0}^{c}\sum_{i=0}^{c-k} P_{i,j}^{k}} = \frac{\sum_{j=1}^{\infty}\sum_{i+k=c} P_{i,j}^{k}}{1-\mathbf{\Pi}_0\mathbf{e}}. \tag{5.26}$$

To understand how system performance measures listed above vary with N, we also perform a numerical investigation to the measures based on changing the value of N from 5 to 25, which is based on $\lambda = 5$, $\mu = 15$, $p = 0.5$, $\sigma = 10$ and $\eta = 10$. The numerical illustration is graphically presented in Figures 5.5-5.8. From Figures 5.5-5.8, it is clear that increasing the retrial rate beyond a certain point does not result in a commensurate improvement in the system performance, which is according with the result of Neuts and Rao [50].

5.5 Numerical Results

In this section, we construct the total expected cost function per unit time based on the system performance measures for the M/M/*c*/BSV retrial queue, in which the number of servers (c) is a discrete decision variable, and the service rate (μ) and the vacation rate (η) are continuous decision variables. Let us define the following cost elements:

$C_h \equiv$ holding cost per unit time per customer present in orbit,
$C_s \equiv$ cost per unit time of providing a service rate μ,
$C_v \equiv$ cost per unit time when one server is on vacation,
$C_r \equiv$ cost per unit time of providing a vacation rate η,
$C_p \equiv$ fixed cost for purchasing one server.

Based on the definition of the cost parameters, the total expected cost function per unit time can be expressed as:

$$F(c,\mu,\eta) = C_h E[Orbit] + C_s\mu + C_v E[V] + C_r\eta + C_p c, \tag{5.27}$$

where L_s and $E[V]$ are defined previously. The main objective is to find the optimal number of servers c^*, and the optimal values of service rate and vacation rate (μ^*, η^*) simultaneously which minimize the cost function $F(c,\mu,\eta)$. The analytical study of the optimization behavior of the expected cost function would have been an arduous task to undertake since the decision variables appear in an expression which is a highly nonlinear and complex and non-linear in terms of (c,μ,η). Next, we

firstly use the Quasi-Newton method to find the optimal value of continuous variable (μ,η), say (μ^*,η^*), and then use direct search method to search the optimal value of discrete variable c, say c^*. For practice situation of purchase budget, the number of servers is bounded by a positive integer $c_U \geq 1$. We want to find the joint optimal value (μ^*, η^*) for each given c in the feasible set $\{1,2,\ldots, c_U\}$. The cost minimization problem can be illustrated mathematically as

$$F(c,\mu^*,\eta^*) = \min_{(\mu,\eta) \text{ and s.t.}(5.8)} \left\{F(c,\mu,\eta)\middle|c\right\}, \quad c = 1,2,...,c_U. \tag{5.28}$$

For the problem of (5.28), we should show the convexity of $F(c,\mu,\eta)$ in (μ,η). However, this work is difficult to implement. It is noted that the derivative of the cost function F with respect to (μ,η) indicates the direction which cost function increases. It means that, the optimal value (μ^*,η^*) can be found along this opposite direction of the gradient. (see Chong and Zak [14]). That is, for a fixed c, Quasi-Newton method is employed to search (μ,η) until the approximate minimum value of $F(c,\mu,\eta)$ is achieved, say $F(c,\mu^*,\eta^*)$. To demonstrate the validness and the optimization solution, we perform some computation and analysis on the examples shown in Table 5.1 by considering the following cost parameters as

C_h=\$25/customer/unit time, C_s=\$45 /unit time,

C_v=\$120/server/unit time, C_r=\$90 /unit time, and C_p=\$120/server.

From Table 5.1, it can be seen that the minimum expected cost per unit time of 1474.377 is achieved at $(\mu^*,\eta^*)=(11.54626, 6.305710)$ by using 6 iterations, which is based on Case (i) with initial value (c,μ,η)=(1,15, 5). Based on Case (ii) with initial value $(c,\mu,\eta)=(2, 10, 10)$, the minimum expected cost per unit time of 1968.692 is achieved at $(\mu^*,\eta^*)=(12.53093, 8.696281)$ by using 6 iterations.

After obtaining the joint optimal value (μ^*,η^*) of the continuous variable (μ,η), we would use direct search method to obtain the optimal c such that the expected cost function $F(c,\mu^*,\eta^*)$ attains a minimum, say $F(c^*,\mu^*,\eta^*)$. Therefore, the cost minimization problem can be illustrated mathematically as

$$F(c^*,\mu^*,\eta^*) = \min_{1\leq c\leq c_U} \left\{F(c,\mu^*,\eta^*)\right\}. \tag{5.29}$$

The procedure to find the optimal solution is described in the following. A numerical example is shown in Table 5.2 based on (i) $(\lambda, p, \sigma)=(10, 0.8, 15)$ and (ii) $(\lambda, p, \sigma)=(15, 0.5, 20)$.

Algorithm: Direct Search Method

Step 1. Set $F^* = M$ which M is a sufficiently large number.

Step 2. For each i from 1 to c_U, set a initial trial solution (μ,η) and use Quasi-Newton method to find the optimal value (μ^*,η^*) and the cost function $F(c,\ \mu^*,\eta^*)$.

Step 3. If the Quasi-Newton method diverges, try another initial trial solution and back to step 1.

Step 4. If $F(c,\ \mu^*,\eta^*) < F^*$, set $F^* = F(c,\ \mu^*,\ \eta^*)$ and $S^*=(c,\ \mu^*,\ \eta^*)$.

It is noted that the optimal value $(c^*,\mu^*,\eta^*) = (4,\ 5.999552,\ 5.046493)$ and the corresponding minimum cost $F^* =$ 1708.284 for Case (i). For Case (ii), $(c^*,\mu^*,\eta^*) = (4,\ 8.099802,\ 5.265980)$ and $F^* = 1819.241$ are optimal. Finally, we perform a sensitivity investigation on the optimal values (c^*,μ^*,η^*). For various values of λ and p, the minimum expected cost $F(c^*,\mu^*,\eta^*)$ and the system performance measures L_s, and $E[V]$ at the optimum values (c^*,μ^*,η^*) are shown in Table 5.3. From Table 5.3, it can be seen that (i) c^* is insensitive to λ or p; (ii) μ^* increases as λ increases; and (iii) η^* increases as λ or p increases. Moreover, the minimum expected cost increases $F(c^*,\mu^*,\eta^*)$ as λ or p increases.

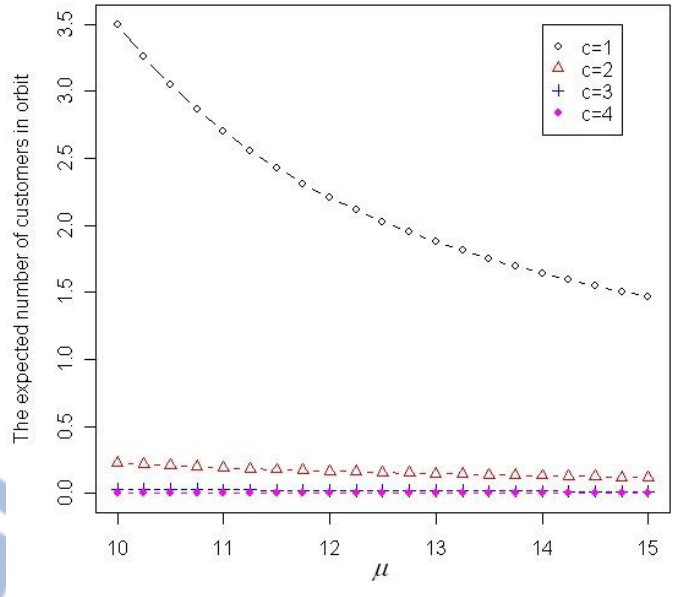

Figure 5.1. The expected number of customers in orbit versus μ

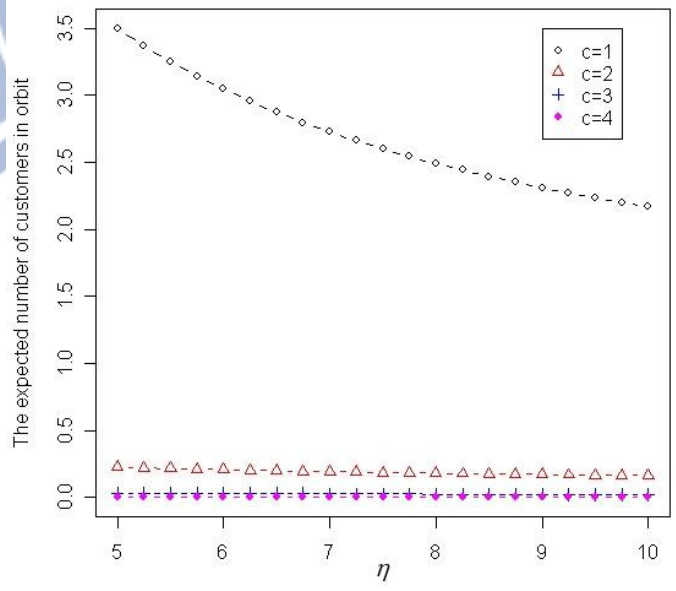

Figure 5.2. The expected number of customers in orbit versus η

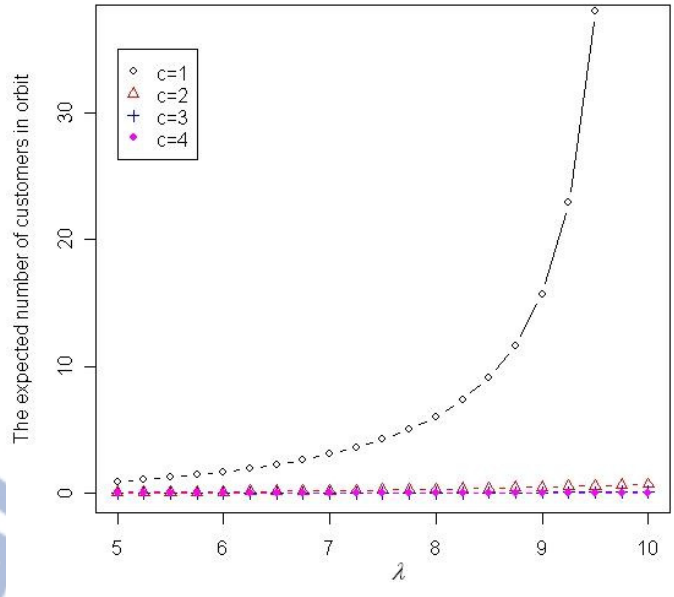

Figure 5.3. The expected number of customers in orbit versus λ

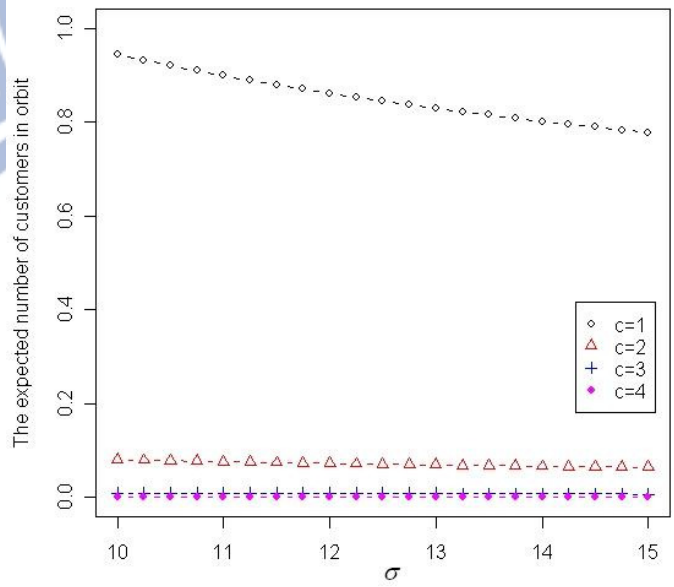

Figure 5.4. The expected number of customers in orbit versus σ

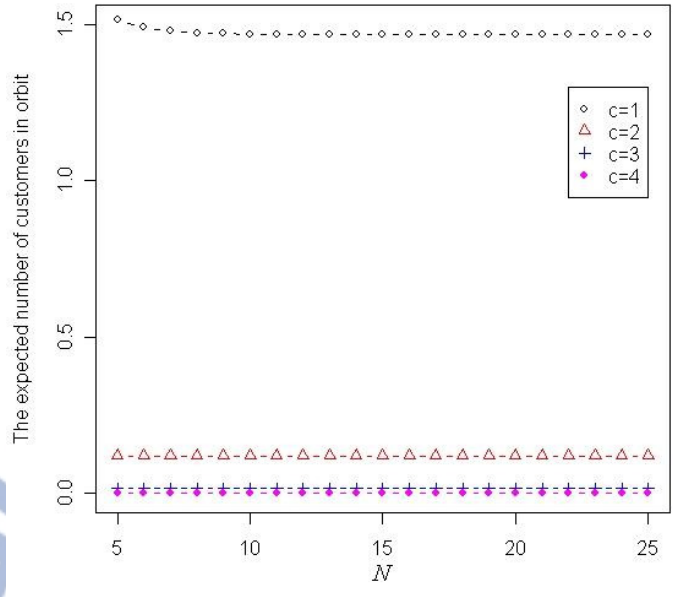

Figure 5.5. The expected number of customers in orbit versus *N*.

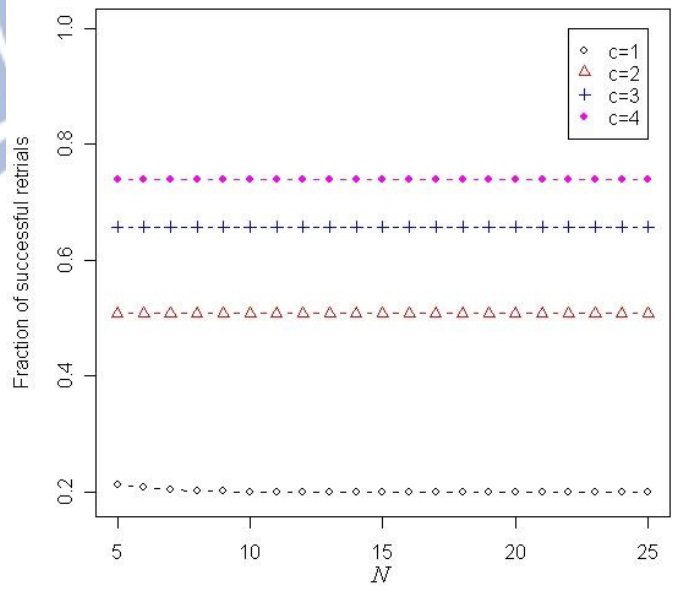

Figure 5.6. The fraction of successful retrials versus *N*

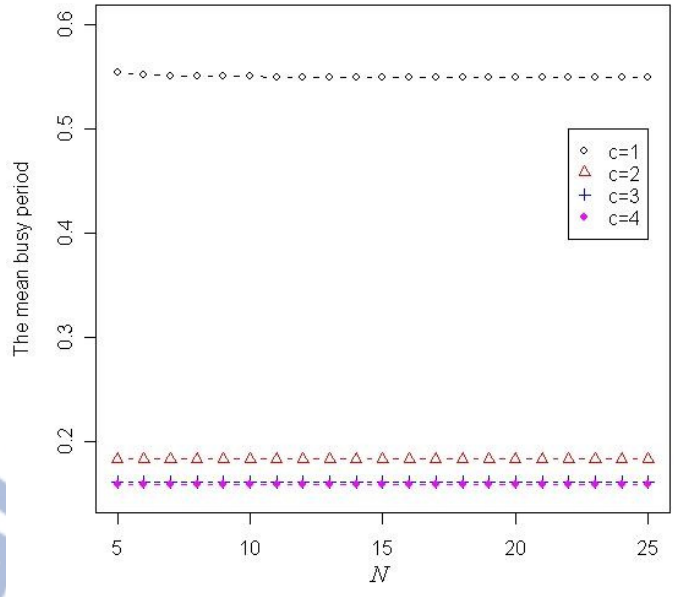

Figure 5.7. The mean busy period versus *N*.

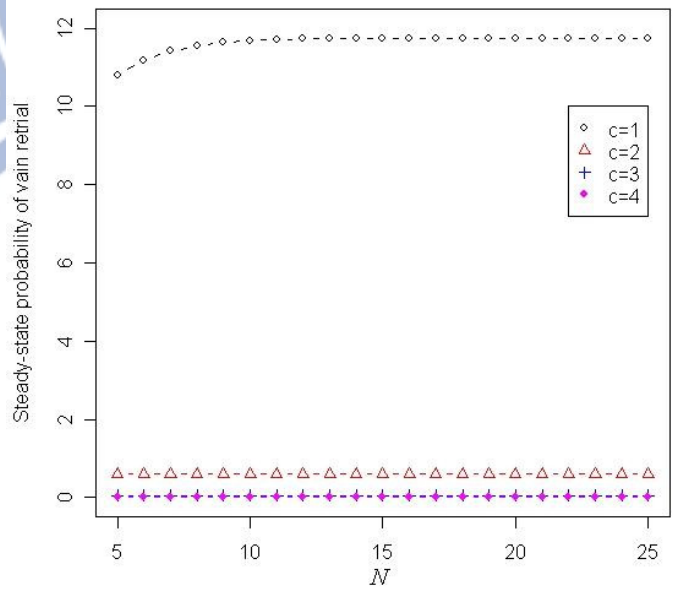

Figure 5.8. The steady-state probability of vain retrial P_V versus *N*.

Table 5.1. The illustration of the implement process of Quasi-Newton method

Case (i): $(\lambda, p, \sigma) = (5, 0.5, 10)$ with initial value $(c, \mu, \eta) = (1, 15, 5)$

Iterations	0	1	2	3	4	5	6
$F(c,\mu,\eta)$	1544.435	1517.015	1482.721	1474.921	1474.380	1474.377	1474.377
μ	15	10.74763	11.11560	11.41594	11.53441	11.54617	11.54626
η	5	5.932174	6.131345	6.263916	6.303111	6.305700	6.305710
$\frac{\partial F}{\partial \mu}$	15.31879	-78.2392	-25.8695	-5.64068	-0.43039	-0.00300	-7.8×10^{-8}
$\frac{\partial F}{\partial \eta}$	-73.2424	-133.720	-43.6031	-9.22994	-0.66640	-0.00424	-1.5×10^{-7}
E[*Orbit*]	7.177405	10.75622	8.070767	6.782341	6.418249	6.388411	6.388210
E[*V*]	0.500000	0.421422	0.407740	0.399111	0.396630	0.396467	0.396466

Case (ii): $(\lambda, p, \sigma) = (10, 0.8, 15)$ with initial value $(c, \mu, \eta) = (2, 10, 10)$

Iterations	0	1	2	3	4	5	6
$F(c,\mu,\eta)$	2037.910	1988.860	1971.630	1968.793	1968.692	1968.692	1968.692
μ	10	11.05421	11.93856	12.42039	12.52661	12.53093	12.53093
η	10	9.256253	8.869115	8.722289	8.697166	8.696282	8.696281
$\frac{\partial F}{\partial \mu}$	-98.0608	-41.9620	-13.3913	-2.29042	-0.09060	-0.00016	-7.7×10^{-9}
$\frac{\partial F}{\partial \eta}$	-35.0235	-22.3227	-9.22534	-1.86890	-0.08050	-0.00014	1.6×10^{-9}
E[*Orbit*]	9.276428	7.785777	6.717369	6.192268	6.074761	6.069724	6.069715
E[*V*]	0.799990	0.862781	0.902006	0.917190	0.919840	0.919933	0.919933

Table 5.2. The optimal value (μ^*, η^*) and the corresponding minimum expected cost

Case (i) $(\lambda, p, \sigma) = (10, 0.8, 15)$

c	Initial Value	Coverage Value (μ^*, η^*)	Iteration	Cost*
1	[25, 15]	[25.13488, 16.43305]	6	3118.635
2	[10, 10]	[12.53093, 8.696281]	6	1968.692
3	[10, 5]	[8.214208, 6.210196]	6	1725.728
4	[5, 5]	[5.999552, 5.046493]	7	1708.284
5	[5, 5]	[4.652035, 4.414643]	7	1779.094

Case (ii) $(\lambda, p, \sigma) = (15, 0.5, 20)$

c	Initial Value	Coverage Value (μ^*, η^*)	Iteration	Cost*
1	[30, 20]	[33.17698, 17.35916]	6	3601.021
2	[15, 10]	[16.60255, 9.183037]	5	2210.467
3	[10, 5]	[10.97471, 6.530226]	10	1882.075
4	[6, 6]	[8.099802, 5.265980]	8	1819.241
5	[5, 5]	[6.347280, 4.561196]	7	1861.652

Table 5.3. The optimal value (c^*, μ^*, η^*) and the minimum expected cost for various value of λ and p.

(λ, p, σ)	(5, 0.2, 10)	(10, 0.2, 10)	(20, 0.2, 10)	(5, 0.8, 10)	(10, 0.8, 10)	(20, 0.8, 10)
c^*	2	3	4	4	4	5
(μ^*, η^*)	[4.965695, 2.123714]	[6.427349, 2.781059]	[9.416220, 3.974561]	[2.997995, 2.998664]	[6.062298, 5.075460]	[9.609657, 7.689420]
$F(c^*, \mu^*, \eta^*)$	901.7296	1245.806	1727.201	1325.523	1716.873	2386.602
E[*Orbit*]	2.825372	3.199280	4.199710	1.626472	3.125312	4.497047
E[*V*]	0.470873	0.719505	1.006400	1.333927	1.576212	2.080781

(λ, p, σ)	(5, 0.2, 10)	(5, 0.5, 10)	(5, 0.8, 10)	(10, 0.2, 15)	(10, 0.5, 15)	(10, 0.8, 15)
c^*	2	3	4	3	3	4
(μ^*, η^*)	[4.965695, 2.123714]	[3.774111, 2.689427]	[2.997995, 2.998664]	[6.347744, 2.767427]	[7.295827, 4.645567]	[5.999552, 5.046493]
$F(c^*, \mu^*, \eta^*)$	901.7296	1116.483	1325.523	1237.045	1511.634	1708.284
E[*Orbit*]	2.825372	2.122060	1.626472	3.024207	3.662626	2.955528
E[*V*]	0.470873	0.929566	1.333927	0.722693	1.076295	1.585259

(λ, p, σ)	(10, 0.2, 5)	(10, 0.2, 10)	(10, 0.2, 15)	(10, 0.8, 5)	(10, 0.8, 10)	(10,0.8, 15)
c^*	2	3	3	4	4	4
(μ^*, η^*)	[10.00245, 3.820378]	[6.427349, 2.781059]	[6.347744, 2.767427]	[6.232824, 5.154912]	[6.062298, 5.075460]	[5.999552, 5.046493]
$F(c^*, \mu^*, \eta^*)$	1361.503	1245.806	1237.045	1739.966	1716.873	1708.284
E[*Orbit*]	5.789514	3.199280	3.024207	3.572681	3.125312	2.955528
E[*V*]	0.5235084	0.719505	0.722693	1.551918	1.576212	1.585259

Chapter 6

Conclusions and Future Research

In this thesis, we considered M/M/*c* and M/M/*c* retrial queues with *SOS* channel, M/M/*c*/MBSV queue, and M/M/*c*/BSV retrial queueing system. For those four queueing systems, it is rather difficult to obtain the steady-state probability explicitly. Thus we employed the matrix-geometric method and recursively matrix-analytical approaches to deal with the probability distributions. The sufficient and necessary conditions for the stability of the queueing systems were derived. The closed-form or approximation procedure of the rate matrix was provided. Various system performances of those four queues were also developed. Using the system performances and cost elements, the cost functions were constructed to determine the optimal parameters setting of the queueing system such that the cost is minimized. Sensitivity analysis was conducted to investigate the effect of changes in the system parameters on the optimal values. In this chapter, we make conclusions and provide possible extensions of the present work for the further research.

6.1 Conclusions

In Chapter 2, we investigated the optimal infinite capacity M/M/*c* queue arisen from some practical situations, where arrivals may need an additional optional service (second optional channel by the server). The matrix-geometric method was employed to deal with the complex steady-state equation system. The stability condition was also developed. Some important system performance measures were derived. A sensitivity analysis was performed to discuss how the system performances can be affected by the input parameters in the investigated queueing service model. Furthermore, we also provided numerical results among the optimal number of channels, the optimal service rates, and minimal cost for the M/M/*c* queue with *SOS* channel.

In Chapter 3, the queue studied in Chapter 2 was extended into the multi-server retrial queue with *SOS* channel. The sufficient and necessary conditions for the stability of the system were discussed. A sequence approximation method was implemented to derive the rate matrix. An efficient algorithm was provided to obtain the stationary probability vectors recursively. The explicit formulae for the system performances were given. A cost model was constructed to calculate the optimal values of the number of servers and the two service rates. A sensitivity analysis of the

joint optimal values with respect to specific values of system parameters was performed.

In Chapter 4, an infinite capacity M/M/*c* system with modified Bernoulli single vacation policy (M/M/*c*/MBSV) was studied used the matrix-geometric method. The necessary and sufficient condition for the stability of the system was deduced. More important, the explicitly closed-form solution of stable condition and the rate matrix of the queue model were obtained. The convergence property of rate matrix was also proofed. We have not only obtained exactly the steady-state probability and the system performance measures using matrix analytical approach but also find the optimal number of servers, the optimal service rate and vacation rate based on the cost function we constructed. Finally, this study is not difficultly extended to the case that server takes multiple vacations when an empty queue is found upon a service completion.

In Chapter 5, we analyzed an M/M/*c* retrial queue with Bernoulli single vacation policy (M/M/*c*/BSV retrial queue). The explicit expression of the stability condition was developed. The stationary probability vectors and some system performance were obtained in matrix forms. A cost model was constructed to investigate the optimal control of the queueing system we discussed. Two efficient methods were employed to deal with the optimization problem heuristically. A sensitivity analysis of the joint optimal values with respect to specific values of system parameters was performed. Based on the analysis, the mathematical model formulates of M/M/*c*/BSV retrial queue and M/M/*c* retrial queue with *SOS* channel are consistent.

6.2 Future Research

We have used the matrix-geometric method to analyze the optimal M/M/*c* queue with *SOS* channel and M/M/*c* retrial queue with *SOS* channel. The optimization of M/M/*c*/MBSV queueing system and M/M/*c*/BSV retrial queue were also investigated. In the future, we may study the following topics:

1. Incorporating the feedback behavior of the customers into the M/M/*c* queue with *SOS* channel and M/M/*c* retrial queue with *SOS* channel.

2. Incorporating the balking and reneging behaviors of the impatient customers into the M/M/*c* queue with *SOS* channel and M/M/*c* retrial queue with *SOS* channel.

3. Incorporating the unreliable property of servers (server breakdown) into the M/M/*c*/MBSV queue.
4. Optimization of the M/M/*c* retrial queue with *J* additional options.
5. Optimization of the PH/M/*c* and MAP/M/*c* queueing system with *SOS* channel.

References

1. Aissani A. (2011). An $M^{[x]}$/G/1 energetic retrial queue with vacations and control, *IMA Journal of Management Mathematics*, 22, 13-32.
2. Al-Jararha J. and Madan K. C. (2003). An M/G/1 queue with second optional service with general service time distribution, *Information and Management Sciences*, 14, 47-56.
3. Amador J. and Artalejo J. R. (2007). On the distribution of the successful and blocked events in the M/M/*c* retrial queue: A computational approach. *Applied Mathematics and Computation*, 190, 1612-1626.
4. Artalejo J. R. (1999). Accessible bibliography on retrial queues, *Mathematical and Computer Modelling*, 30, 1-6.
5. Artalejo J. R. (2010). Accessible bibliography on retrial queues : Progress in 2000-2009, *Mathematical and Computer Modelling*, 51, 1071-1081.
6. Artalejo J. R. and Choudhury G. (2004). Steady state analysis of an M/G/1 queue with repeated attempts and two-phase service, *Quality Technology & Quantitative Management*, 1(2), 189-199.
7. Artalejo J. R. and Pozo M. (2002). Numerical calculation of the stationary distribution of the main multiserver retrial queue, *Annals of Operations Research*, 116, 41-56.
8. Artalejo J. R., Chakravarthy S. R. and Lopez-Herrero M. J. (2007). The busy period and the waiting time analysis of a MAP/M/*c* queue with finite retrial group, *Stochastic Analysis and Applications*, 25, 445-469.
9. Artalejo J. R., Economou A. and Lopez-Herrero M. J. (2007). Algorithmic analysis of the maximum queue length in a busy period for the M/M/*c* retrial queue, *INFORMS Journal on Computing*, 19(1), 121-126.
10. Artalejo J. R., Economou A. and Lopez-Herrero M. J. (2007). Algorithmic approximations for the busy period distribution of the M/M/*c* retrial queue, *European Journal of Operational Research*, 176, 1687-1702.
11. Breuer L., Dudin A. N. and Klimenok V. I. (2002). A retrial BMAP/PH/N system, *Queueing Systems*, 40, 433-457.
12. Bright L. and Taylor P. G. (1995). Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Communications in Statistics-Stochastic Models*, 11, 497-525.
13. Chakravarthy S.R. and Dudin A.N. (2002). Multiserver retrial queue with BMAP arriving and group services, *Queueing Systems*, 42, 5-31.
14. Chong E.K.P. and Zak S.H. (2001). An Introduction to Optimization, 2nd Ed., John Wiley and Sons, Inc., 2001.
15. Choudhury G. (2000). An $M^{[x]}$/G/1 queueing system with a setup period and a vacation period, *Queueing Systems*, 36, 23-38.

16. Choudhury G. (2002). A batch arrival queue with a vacation time under single vacation policy, *Computers & Ops Res.*, 29, 1941-1955.

17. Choudhury G. (2007). A two phase batch arrival retrial queueing system with Bernoulli vacation schedule, *Applied Mathematics and Computation*, 188, 1455-1466.

18. Choudhury G. (2008). Steady state analysis of an M/G/1 queue with linear retrial policy and two phase service under Bernoulli vacation schedule, *Applied Mathematical Modelling*, 32, 2480-2489.

19. Choudhury G. and Deka K. (2008). An M /G /1 retrial queueing system with two phases of service subject to the server breakdown and repair, *Performance Evaluation*, 65(10), 714-724.

20. Choudhury G. and Deka K. (2009). An $M^{[x]}$/G/1 unreliable retrial queue with two phases of service and Bernoulli admission mechanism, *Applied Mathematics and Computation*, 215, 936-949.

21. Choudhury G. and Madan K. C. (2004). A two phase batch arrival queueing system with a vacation time under Bernoulli schedule, *Applied Mathematic and Computation*, 149, 337-349.

22. Choudhury G. and Madan K. C. (2005). A two-stage batch arrival queueing system with a modified Bernoullu schedule vacation under *N*-policy, *Mathematical and Computer Modelling*, 42, 71-85.

23. Choudhury G. and Paul M. (2006). A batch arrival queue with a second optional service channel under *N*-policy, *Stochastic Analysis and Applications*, 24, 1-21.

24. Choudhury G. and Tadj L. (2009). An M/G/1 queue with two phases of service subject to the server breakdown and delayed repair, *Applied Mathematical Modelling*, 33(6), 2699-2709.

25. Choudhury G. and Tadj L. (2011). The optimal control of an $M^{[x]}$/G/1 unreliable server queue with two phases of service and Bernoulli vacation schedule, *Mathematical and Computer Modelling*, 54, 673-688.

26. Choudhury G., Ke J. C. and Tadj L. (2009). The *N*-policy for an unreliable server with delaying repair and two phases of service, *Journal of Computational and Applied Mathematics*, 231, 349-364.

27. Choudhury G., Tadj L. and Paul M. (2007). Steady state analysis of an $M^{[x]}$/G/1 queue with two phase service and Bernoulli vacation schedule under multiple vacation policy, *Applied Mathematical Modelling*, 31, 1079-1091.

28. Dimitriou I. and Langaris C. (2010). A repairable queueing model with two-phase service, start-up times and retrial customers, *Computers & Operations Research*, 37, 1181-1190.

29. Doshi B. T. (1986). Queueing systems with vacations-a survey, *Queueing Systems*, 1, 29-66.

30. Falin G. I. and Templeton J. G. C. (1997). Retrial Queues, Chapman and Hall,

London.

31. Gomez-Corral A. (2006). A bibliographical guide to the analysis of retrial queues through matrix analytic techniques, *Annals of Operations Research*, 141, 177-207.

32. Ke J. C. (2008) An $M^{[x]}/G/1$ system with startup server and *J* additional options for service, *Applied Mathematical Modelling*, 32, 443-458.

33. Ke J. C. and Chang F. M. (2009). $M^{[x]}/(G_1, G_2)/1$ retrial queue under Bernoulli vacation schedules with general repeated attempts and starting failures, *Applied Mathematical Modelling*, 33, 3186-3196.

34. Ke J. C. and Chu Y. K. (2006). A modified vacation model $M^{[x]}/G/1$ system, *Applied Stochastic Business and Industry*, 22, 1-16.

35. Ke J. C., Chang C. J. and Chang F. M. (2010). Controlling arrivals for a Markovian queueing system with a second optional service, *International Journal of Industrial Engineering-Theory Applications and Practice*, 17(1), 48-57.

36. Ke J. C., Wu C. H. and Zhang Z. G. (2010). Recent Developments in Vacation Queueing Models : A Short Survey. *International Journal of Operations Research*, 7(4), 3-8.

37. Keilson J. and Servi L. D. (1986). Oscillating random walk models for GI/G/1 vacation system with Bernoulli schedules, *Journal of Applied Probability*, 23, 790-802.

38. Kim C. S. Klimenok V., Mushko V. and Dudin A. (2010). The BMAP/PH/N retrial queueing system operating in Markovian random environment, *Computers & Operations Research*, 37, 1228-1237.

39. Kulkarni V. G. and Liang H. M. (1997). Retrial queues revisited, in Frontiers in Queueing: Models and Applications in Science and Engineering, J.H. Dshalalow, ed., CRC Press, Inc., Boca Raton, FL.

40. Langaris C. and Dimitriou I. (2010). A queueing system with n-phases of service and (n-1)-types of retrial customers. *European Journal of Operational Research*, 205, 638-649.

41. Latouche G. and Ramaswami V. (1999). Introduction to Matrix Analytic Methods in Stochastic Modeling (ASA-SIAM Series on Statistics and Applied Probability).

42. Lee H. W., Lee S. S., Park J. O. and Chae K. C. (1994). Analysis of $M^{[x]}/G/1$ queue with *N* policy and multiple vacations, *Journal of Applied Probability*, 31, 467-496.

43. Lee S. S. Lee, H. W., Yoon S. H and Chae K. C. (1995). Batch arrival queue with *N* policy and single vacation, *Computers & Operations Research.*, 22, 173-189.

44. Levy Y. and Yechiali U. (1976). An M/M/*c* queue with servers' vacations, *INFOR*, 14, 153-163.

45. Lin C. H. and Ke J. C. (2011). On the multi-server retrial queue with geometric loss and feedback: computational algorithm and parameter optimization, *International Journal of Computer Mathematics*, 88(5), 1083-1101.

46. Madan K. C., Abu-Dayyeh W. and Taiyyan F. (2003). A two server queue with Bernoulli schedules and a single vacation policy, *Applied Mathematics and Computation*, 145, 59-71.

47. Madan K.C. (2003). An M/G/1 queue with second optional service. *Queueing Systems*, 34, 37-46.

48. Medhi J. (2002). A single server Poisson input queue with a second optional channel. *Queueing Systems*, 42, 239-242.

49. Neuts M. F. (1981). Matrix Geometric Solutions in Stochastic Models: an Algorithmic Approach, The John Hopkins University Press, Baltimore, 1981.

50. Neuts M. F. and Rao B. M. (1990). Numerical investigation of a multiserver retrial model, *Queueing Systems - Theory and Applications*, 7, 169-190.

51. Sherman N. P. and Kharoufeh J. P. (2011). Optimal Bernoulli routing in an unreliable M/G/1 retrial queue, *Probability in the Engineering and Informational Sciences*, 25, 1-20.

52. Stepanov S. N. (1999). Markov models with retrials: The calculation of stationary performance measures based on the concept of truncation. *Mathematical and Computer Modelling*, 30, 207-228.

53. Tadj L., Choudhury G. and Tadj C. (2006). A quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule, *Quality Technology and Quantitative Management*, 3(2), 145-160.

54. Takagi H. (1991). Queueing analysis: A foundation of performance evaluation, Vol. I, vacation and priority systems, Part I. North-Holland, Amsterdam.

55. Taleb S. and Aissani A. (2010). Unreliable M/G/1 retrial queue: monotonicity and comparability, *Queueing Systems - Theory and Applications*, 64, 227-252.

56. Tien V. D. (2010). A new computational algorithm for retrial queues to cellular mobile systems with guard channels, *Computers & Industrial Engineering*, 59, 865-872.

57. Tien V. D. (2010). An efficient computation algorithm for a multiserver feedback retrial queue with a large queueing capacity, *Applied Mathematical Modelling*, 34, 2272-2278.

58. Tien V. D. and Ram C. (2010). An efficient method to compute the rate matrix for retrial queues with large number of servers, *Applied Mathematics Letters*, 23, 638-643.

59. Wang J. (2004) An M/G/1 queue with second optional service and server breakdowns, *Computers and Mathematics with Applications*, 47, 1713-1723.

60. Wang J. T. and Li J. H. (2010). Analysis of the $M^{[x]}$/G/1 queues with second multi-optional service and unreliable server, *Acta Mathematicae Applicatae*

Sinica, 26(3), 353-368.

61. Wang K. H., Yang D. Y. and Pearn W. L. (2010). Comparison of two randomized policy M/G/1 queues with second optional service, server breakdown and startup, *Journal of Computational and Applied Mathematics*, 234, 812-824.
62. Wu J., Liu Z. and Peng Y. (2009). On the BMAP/G/1 G-queues with second optional service and multiple vacations, *Applied Mathematical Modelling*, 33, 4314-4325.
63. Wu J., Liu Z. and Yang G. (2011). Analysis of the finite source MAP/PH/N retrial G-queue operating in a random environment, *Applied Mathematical Modelling*, 35, 1184-1193.
64. Yang D. Y., Wang K. H. and Kuo Y. T. (2011). Economic application in a finite capacity multi-channel queue with second optional channel, *Applied Mathematics and Computation*, 217, 7412-7419.
65. Yang T. and Templeton J. G. C. (1987). A survey on retrial queues, *Queueing Systems - Theory and Applications*, 2, 201-233.
66. Yang W. S. and Dug H. M. (2011). Approximation of M/M/*c* retrial queue with PH-retrial times, *European Journal of Operational Research*, 213, 205-209.