A Study on Quality of Service Based Congestion Management

THESIS

Submitted in partial fulfilment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

SUMAN KASWAN

Under the Supervision of

PROF. CHANDRA SHEKHAR

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

October 2023

"*Sometimes mess is more than just the easy choice - it's the optimal choice.*"
— by Brian Christian

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis titled "**A Study on Quality of Service Based Congestion Management**" submitted by **Ms. Suman Kaswan**, ID No. **2019PHXF0434P** for the award of Ph.D. of the institute embodies original work done by her under my supervision.

Signature of the Supervisor
Name: **PROF. CHANDRA SHEKHAR**
Designation: **Professor**
Date: **October 9, 2023**

Acknowledgements

First and foremost, I thank the Almighty God for blessing me with courage, confidence, purpose, enthusiasm, and good health to reach so far and attain my goals till today, and I pray for the same in my future endeavors.

I wish to acknowledge all those who have helped and encouraged me throughout the journey of doctoral research. First of all, I owe my heartfelt gratitude and indebtedness to my esteemed supervisor, **Prof. Chandra Shekhar** for his enlightening guidance and sympathetic attitude exhibited during the entire course of this work. His insightful feedback, constant encouragement, and valuable discussions pushed me to improve my research work. Mere words would not suffice to express gratitude for the parental love received from him and his family members during this entire journey.

I am thankful to **Prof. V. Ramgopal Rao**, Vice-Chancellor, **Prof. Sudhirkumar Barai**, Director, **Prof. M. B. Srinivas**, Dean Academic (AGSR), and **Prof. Shamik Chakraborty**, Associate Dean (AGSRD), for giving me the opportunity to achieve a challenging position in respective field pertinent to my qualifications, which allowed me to use my skills to prove myself worthy. I am further thankful to them for providing me with facilities for my research work and a healthy environment.

I am grateful for the opportunity provided by the Department of Mathematics, BITS Pilani to pursue research. I humbly acknowledge **Prof. Devendra Kumar**, HoD, Department of Mathematics and Ex-Hod, **Prof. B. K. Sharma** for providing all necessary documents required for conducting this research. I would like to acknowledge all the faculty members and staff of the Mathematics Department for demonstrating genuine interest and enthusiasm in their teaching and relentless support. I am also thankful to my doctoral advisory committee (DAC) members, **Prof. R.P. Mishra** and **Prof. Shivi Agarwal**, for their suggestions and constructive inputs to enrich this study during presentations and seminars throughout my Ph.D. tenure. I extend my sincere thanks to departmental research committee (DRC) Convener, **Prof. Ashish Tiwari** and other committee members for their helpful assistance.

I shall always remain indebted to my parents **Mrs. Rammoorti Devi** and **Mr. Budha Ram Kaswan** for their care, love, and encouragement in all my endeavors. I express my deepest sense of gratitude towards my husband, **Mr. Manish Kumar Singh**, for extending every care, moral support, and affection to enable this work to become a reality. I am grateful for the continuous support of my mother-in-law, **Mrs. Leelawati Singh**, my father-in-law, **Mr. Ram Awadh Singh**, my brother, **Mr. Shishpal Kaswan**, my sister-in-law, **Mrs. Pramod Kumari**, my brothers-in-law, **Mr. Abhishek Kumar Singh**, **Mr. Kuldeep Singh**,

and **Mr. Gaurav Kumar Singh**, my sisters, **Mrs. Santosh**, and **Mrs. Pooja Singh**. I am deeply indebted to them for all the pains they took to make my dream come true. My special love goes to my niece and nephew, **Aryama, Priyansh, Aryaveer, Kanvika, Sharvil**, and **Bhavik** for the joy of my heart.

I am fortunate to have **Ms. Sonu Jakhar** as my best friend who stands with me in every good and bad situation during this journey. With heartfelt thanks, I extend special appreciation to **Mrs. Shikha Gupta** and junior-cum-friend **Mr. Vijender Yadav** for their care, support, and laughter moments during the entire tenure. I am grateful for the assistance given by my seniors, **Dr. Amit Kumar** and **Dr. Shreekant Varshney**, colleague **Mr. Mahendra Devanda**, and junior **Mr. Ankur Saurav**. I thank **Dr. Sangita Yadav** for keeping a friendly nature with utmost care and affection toward me. I wish to acknowledge all my friends, in particular, **Ashvini**, **Anshu**, **Shipra**, **Sonu**, **Poonam**, **Raveena**, **Gourav**, **Amit**, **Umesh**, **Mahendra**, **Parveen**, **Himanshu**, and **Satpal**, department fellows, and well-wishers, for their continuous help and support. I have really enjoyed their company at BITS PILANI.

I thankfully acknowledge BITS PILANI for providing me financial assistance as Institute Fellowships during my tenure at BITS Pilani as a Ph.D. research scholar.

Place: BITS Pilani
Date: October 2023

Suman Kaswan
(Department of Mathematics)

Abstract

The present thesis addresses the problem of formulation of realistic service systems using various heuristic and meta heuristic optimization techniques. The main aim of this study is to analyze the behavior of the $M/M/1$ queueing networks with various features like vacation, retrial, balking, reneging, jockeying, control policy, differentiated vacation, server breakdown, and many others.

This research work consists of nine chapters. Chapter 1 introduces about the essential service system characteristics and terminologies, random processes, an exhaustive literature survey on queueing theory, gaps in the existing research, objectives of the thesis, and methodologies used in this study.

Chapter 2 deals with the customers' impatient attributes in congestion using a queueing theoretic approach which is motivated by observing real service systems where these queueing occurrences interact. The impatience attributes taken in this model are balking, reneging, and jockeying in an $M/M/1$ queueing system with 2 servers.

Chapter 3 consists of a service system with two types of unreliable servers and the service is provided in two phases. In such tandem queues, the service is completed only when all phases of services are rendered successfully. In the present model, the server of the initial phase has a dual role as a server for the first phase and as a customer for the second phase.

Chapter 4 contains a two-phase stochastic queueing system wherein initial phase service can be either rendered offline or online and final phase service is rendered in offline mode only. In this multi-phase and multi-server tandem queue model, the arrival control policy, namely F-policy, and the balking phenomena are considered for online and offline customers, respectively.

Chapter 5 presents the optimal analysis of a F-policy $M/M/1/K$ service system with unreliable service and exponential startup time. This chapter focuses on optimal policies for the highly efficient service system since the congestion of the customers more often originates from degraded policies rather than faulty arrangements.

Chapter 6 analyzes a finite capacity service system with several realistic customer-server phenomena: customer impatience, server's partial breakdown, and threshold recovery policy. This queueing model also incorporates the concept of service pressure coefficient to model real-time strategic policy.

Chapter 7 deals with the critical issue of the single-server congestion problem with prominent customer impatience attributes and server strategic differentiated vacation. Despite their apparent practical relevance, the proposed congestion problem has yet to be studied from a service/production perspective with transient analysis.

Chapter 8 presents the notion of orbital search mechanism in Markovian retrial queueing, including multiple vacation policies, and server breakdown. Processes like arrival, service, search, repair, and vacation are all stochastic in nature. System characteristics are derived using the probability generating function (PGF) technique.

Finally, Chapter 9 summarizes the major contributions of the thesis work along with some future direction.

Contents

List of Figures

List of Tables

List of Abbreviations

CTMC	Continuous Time Markov Chain
DTMC	Discrete Time Markov Chain
MAM	Matrix-Analytic Method
FCFS	First-Come, First-Served
IID	Independent and Identically Distributed
QBD	Quasi-Birth-Death
QNM	Quasi-Newton Method
QoS	Quality of Service
MRP	Machine Repair Problem
PDF	Probability Density Function
PGF	Probability Generating Function
PSO	Particle Swarm Optimization
CS	Cuckoo Search
TLBO	Teaching Learning Based Optimization
GWO	Grey Wolf Optimizer
GOA	Grasshopper Optimization Algorithm
SGO	Social Group Optimization
AOA	Archimedes Optimization Algorithm

Dedicated To

My Husband, Mr. Manish Kumar Singh,

for his unconditional love throughout the journey

&

My Beloved Family

for making it possible to complete this journey

Chapter 1

Introduction

This chapter presents the thesis work's motivation, general introduction, and historical literature survey. The chapter discusses the basics of queueing theory and the various methodologies used throughout this study. It also includes the objectives and scope of the present work.

1.1 Motivation

Queueing theory is a mathematical study of waiting lines. In the year 1909, queueing theory was introduced by A. K. Erlang, a Danish engineer and mathematician who worked for the Copenhagen Telephone Exchange and published the first paper on what would now be called queueing theory. He modeled the number of telephone calls arriving at an exchange by a Poisson process. Queueing theory is one of the truest areas of statistics and has a wider applicability in several systems. The queueing models can effectively and efficiently utilize the resources in several systems. Waiting is a naturally occurring phenomenon in service systems, which leads to an annoying and dissatisfying situation among customers. The common reason for waiting is large service requests against constrained capacity and a shortfall in capital to provide service. Thus, customer waiting leads to the formation of a queue in front of service providers, who experience a situation called congestion. Despite congestion, it is essential to maintain the quality of service, which is the primary motivation behind this research. The anticipated explosive growth in the number of arrivals in service systems like health care, emergency services, ticket counters, communication services, public transportation, etc., which are fundamental to daily life, imposes unprecedented challenges. System design and management are prerequisites for handling such challenges. Based on the identified problem, efforts are directed in this work toward developing new service regimes and strategies for improving service quality in a congested system. The strategies include increasing the service capacities, incorporating a backup server in case of unreliable service, distributing the workload among servers smoothly, implementing a controllable arrival policy to avoid congestion, and others. The demand for service and service times are both stochastic, leading to congestion as arriving service requests might not be fulfilled immediately when service providers are busy serving other customers.

Online service systems have gained immense research interest in the past few years due to the severe congestion that service facilities will face with the rapid development of the internet and electronic devices. Online ordering is becoming a cornerstone standard defining congestion management. Such online processes are employed in various real-world contexts, including selling airline tickets, cloud resource allocation, sponsored search, real-time bidding in display advertisement, dynamic fleet management, fog computing, and real-time ride-sharing [137], [26]. However, this study primarily focuses on employing a queueing theoretic approach to represent a service mechanism.

This study aims to optimally configure the congested system while improving the Quality of Service (QoS). In this work, the customer-server interaction is mathematically modeled as a queue system with specific features. In this work, the queueing model is first developed and formulated, resulting in a closed-form expression for steady-state or transient-state performance measures of the system. Then the findings are applied in the design or reconfiguration of the service system with a balance between system design cost and service quality cost. The system design cost includes the cost of servers for providing service and system maintenance costs. In contrast, the service quality cost is the sum of the customer's accessing and waiting costs [83].

1.2 Service systems

The term service is a provision of assistance and expertise through interaction between interdependent parts like people, technologies, and providers that is externally oriented to achieve and maintain a sustainable competitive advantage [19]. The term "services" can also refer to a set of actions in which resources of different kinds (people, physical resources, commodities, and systems of service providers) are employed in contact with a customer to address a problem or meet demand. A service system is defined as a configuration of people, technologies, organizations, and shared information that is able to create and deliver value to providers, users, and other interested entities through service [129]. In the era of the internet and the fourth industrial revolution, there is a continuous upgrade in the needs of customers, and service facilities undergo an enormous transformation and adapt a new model using state-of-the-art technology with flexibility to meet customer satisfaction and enhance their competitiveness [227]. The adoption of self-monitoring, analysis, and reporting technology affects value co-creation in service systems [134]. This monitoring, however, incurs a large amount of resource costs, which include software, hardware, and human labour.

1.2.1 Quality of Service (QoS)

QoS is the description or measurement of the overall performance of a service, such as improvement in operational processes, customer satisfaction, reliable service, and many other performance outcomes. QoS is a pool of technologies that ensures the performance of crucial applications even when the system efficiency is constrained. The economic success of service systems depends on their ability to provide assured QoS. The QoS is measured on three parameters: (a) customer attribute; (b) server attribute; and (c) service attribute. The services of multimedia are an enticing trend distributed by the internet of the future, which requires diverse QoS [85].

1.2.2 Failure of Service System

The unavailability of a facility can be treated as the facility's failure to provide service. Detection of failures in a service system, or service monitoring, is crucial to the improvement of QoS and plays a vital role in distributed service-based systems where services are interdependent, and in turn, the failure of one service may cause the failure of other services [220].

It is generally unrealistic or infeasible to make sufficient resources available to construct service facilities in large numbers so that all conceivable service requests can be met immediately. Since there are a fixed number of servers due to the cost associated with each server, congestion arises in such a system due to the long waiting times and heavy flow of incoming service requests.

1.3 Congestion Management Using Queueing Theory Tools

Congestion is a natural phenomenon in several systems dealing with queues. Reducing congestion and providing QoS is a prerequisite for any service station. Catering for an enormous amount of service requests in the system significantly increases congestion and degrades the achievable QoS. Examples of congested systems include (i) the accident and emergency departments of a hospital, (ii) vehicles in traffic jams, (iii) incoming calls in call-centers, and (iv) large data traffic in computing networks. Modeling the congestion evolution triggered by user service requests for mult-applications is significant in mechanism analysis.

Congestion is a pervasive problem, and it significantly impacts the system from an economic perspective through increased waiting times. The poor management of systems, the exponential growth in service requests, unreliable servers, the absence of adequate planning, the limited use of technology, and the lack of funds for preparation and implementation of measures oriented to improve service systems have acted as catalysts in increasing congestion. Broadly, two schools of thought exist for quantifying congestion on service systems. The first considers the ratio of inflow service request volume to the capacity of the service facility as a measure to quantify congestion. The second considers waiting time in receiving service as a measure to define and characterize congestion [68]. After quantifying congestion within the system, it is imperative to categorize the congestion into distinct levels that accurately represent QoS. The development of efficient service regimes is needed to cope with the increasing service requests while still providing satisfactory QoS [1]. The QoS-based service systems with congestion can be improved by facilitating the following criteria: (a) monitoring of existing congestion levels; (b) evaluation of the effectiveness

of congestion strategies; and (c) identification and prioritization of critical segments in the system.

The basic idea of queueing model has been borrowed from the every-day experience of the queues at the checkout counters in a supermarket, calls arriving in call centers, data transfer in computer networks etc.

1.4 Characteristics of Service Systems

A service system can be evaluated quantitatively by using a mathematical characterization of the process. In general, there are eight basic aspects that provide comprehensive details about the system.

Arrival Process

The pattern of arriving customers in the service system is stochastic or random in nature; therefore, a probability distribution can be identified with the arrivals, i.e., inter-arrival times. The time dependency factor classifies the arrival process as either stationary (time-independent) or nonstationary (time-dependent). In the stationary arrival process, after a considerable time period, the system reaches an equilibrium state, and the mean arrival rate (λ) becomes constant, which can be calculated mathematically as

$$\lambda = \frac{1}{\int_0^\infty t f(t) dt}$$

where, $f(t)$ is the probability density function (PDF) of the inter-arrival time T.

Service Process

The service providers in the system follow certain service mechanisms to serve the customers. The time between successive service completions is called service times, which is not constant for all customers, so service times behaves stochastically. In this way, service times follow a probability distribution depending on their nature. Service, like arrivals, can be classified as stationary or on-stationary with respect to time. Another factor is the situation in which services depend on the number of customers queueing up, which is defined as a state-dependent service.

Number of Servers

The size of the service providers in the system classifies it into either single-server or multi-server service systems. In the case of a single server, there is one queue of customers waiting for service, whereas in other cases, there are several possible configurations, either a single queue for all service providers or a different queue for individual servers. In a multi-server service system, servers provide service at the same or different rates to the arriving customers, called homogeneous or heterogeneous servers, respectively. It is generally assumed that the servers operate independently of each other.

System Capacity

The size of the system for accommodating customers is generally limited. In virtual queueing, the size is considerable enough to consider it an infinite-capacity service system. The arrivals after the maximum capacity is reached are forced to balk the system or are termed as lost customers.

Size of Prospective Arrivals

The prospective customers are those who join a service system based on their requirements. For example, a patient seek service from a hospital, customer in need of household items joins a grocery or super market store, person going to their destination waits for transportation service and many more. Therefore, depending on the size of the population requiring a particular service, prospective customers sizes can be either finite or infinite. In general, service systems have an infinite number of prospective customers, it is unpredictable to determine who might need a particular service.

Queue Discipline

The service mechanism followed by service providers for arriving customers determines the queue discipline. There are several ways to select customers for service, like First-come, First-served (FCFS), Last-come, First-served (LCFS), service in random order (SIRO), priority, processor sharing (PS), etc. In general, FCFS queue discipline is most preferred.

Stages of Service

The service can be rendered in one go or in multiple stages depending on the arrangement of servers in system as either arranged in parallel or in series. A queue in which service is provided in phases or servers are arranged in series is termed a tandem queue. In such

queueing systems, customers may be blocked or starved, as they can leave the system after completion of all stages of service allocated in series.

Customer Behavior

The behavior of customers to join or remain in the system is unpredictable due to long waiting queues. The common impatient behaviors shown by customers include balking, reneging, and jockeying. Balking is a probabilistic impatient phenomenon of customers in which a customer may leave the system at an epoch of joining in the system due to a long waiting queue. Reneging is another unpredictable impatient phenomenon of customers; wherein a customer initially enters the system but may decide to leave the system after some time due to impatience in waiting queues. On the other hand, jockeying takes place as the difference between the queue lengths increases. Customers may decide to switch from a longer queue to an approximative shorter queue if it is perceived that the waiting time can be reduced by switching lines. However, if the jockeying takes place, the customer must join the other queue's end and leave the current queue's position. In practice, customers show their impatience level depending on their queue position and level of satisfaction when they know they are close to or far from the front of the queue.

1.5 Notation of Service Models

In service models with many stochastic processes and the involvement of several parameters as random variables, there is a need to categorize and describe these models succinctly in a mathematical short form. In 1953, a British statistician, D.G. Kendall [103], devised a shorthand notation known as Kendall notation in the form $A/B/C/X/Y/Z$, where
A: specifies the arrival process or the distribution for inter-arrival times
B: describes the service time distribution
C: number of servers
X: system capacity
Y: size of prospective arrivals
Z: the queue discipline.
Generally, only the first three symbols are used when there are no restrictions on the size of the system and prospective arrivals, and queue discipline is FCFS. In this case, the convention is to omit the corresponding symbols from the queueing system representation. For symbols A and B, the notation used to denote the exponential distribution is M which stands for the Markovian or memoryless property of the exponential. Note that the Poisson arrival

process is represented by M (i.e., exponential inter-arrival times); thus, an $M/M/1$ queueing system means a FCFS single server queueing system with a Poisson arrival process, and independent and identically distributed (IID) exponential service times.

1.6 Random Processes

1.6.1 Stochastic Process

Let $N(t)$ be a random variable signifying the state of a system at time t. Stochastic process is defined as the collection of such random variables, i.e., $\{N(t), t \in T\}$, where T indicate the domain of time t.

Discrete-Time Stochastic Process

The stochastic process with T as a discrete set, i.e., $T = \{0, 1, 2, ...\}$.

Continuous-Time Stochastic Process

The stochastic process with T as a continuous set, i.e., $T = \{t | t \geq 0\}$.

State Space

The collection of all possible values of $\{N(t); t \in T\}$ is called state space of the stochastic process.

1.6.2 Counting Process

A counting process typically represents the cumulative number of events that have occurred till time t. A stochastic process $\{N(t), t \geq 0\}$ is said to be a counting process if following holds:

1. $N(t)$ is non-negative inter-valued
2. $N(t)$ non-decreasing in time
3. for $s < t$, $N(t) - N(s)$ is the number of events that occur in time-interval $(s, t]$.

1.6.3 Poisson Process

The Poisson process is one of the most widely-used counting processes for modeling aarivals to the service system. It is usually used in scenarios where we are counting the occurrences of certain events that appear to happen at a certain rate, but completely at random (without a certain structure). The Poisson process or its extensions have been used to model in examples like:

- the requests for individual documents on a web server
- arrival of customer in a service system
- the location of users in a wireless network

Definition: The counting process $\{N(t), t \geq 0\}$ is called a Poisson process with rate $\lambda > 0$ if following conditions hold:

1. $N(0) = 0$
2. $\{N(t), t \geq 0\}$ has independent occurrences of events in disjoint time intervals, called independent increments. Mathematically,

$$\begin{aligned} \text{Prob}[N(\Delta t) = 0] &= 1 - \lambda \Delta t + o(\Delta t) \\ \text{Prob}[N(\Delta t) = 1] &= \lambda \Delta t + o(\Delta t) \\ \text{Prob}[N(\Delta t) = 2] &= o(\Delta t) \end{aligned}$$

   where, $o(\Delta t)$ is a function such that $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$

3. The number of occurrences of events in any interval of length t is Poisson distributed with parameter λt, i.e.,

$$\text{Prob}[N(t+s) - N(s) = n] = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \quad \forall s$$

Properties of Poisson Processes

1. **Superposition property:** Let $\{N_i(t), t \geq 0\}$, $i = 1, 2, ..., k$ be independent Poisson processes with corresponding rates λ_i. If $N(t) = N_1(t) + N_2(t) + ... + N_k(t)$ then $\{N(t), t > 0\}$ is also a Poisson process with rate $\lambda_1 + \lambda_2 + ... + \lambda_k$.
2. **Decomposition property:** The occurrences of events follow Poisson distribution with rate λ. Suppose each events is recorded with property p, independent of anything

else. Let $N_1(t)$ and $N_2(t)$ denote the number of events recorded and not recorded, respectively by time t. Then, the processes $\{N_1(t), t \geq 0\}$ and $\{N_2(t), t \geq 0\}$ are independent Poisson processes with rates $p\lambda$ and $(1-p)\lambda$, respectively.

3. **Exponentially distributed inter-arrival times:** When the number of occurrences of events in any interval is a Poisson random variable, the inter-arrival follows exponential distribution and conversely, when the inter-arrival times follows exponential distribution then the number of arrivals in a time interval is given by Poisson distribution and process is Poisson arrival process.

4. **Memoryless or Markovian property of inter-arrival times:** The memoryless property of a Poisson process means that if we observe the process at a certain point in time, the distribution of the time until next arrival is not affected by the fact that some time interval has passed since the last arrival. Mathematically,

$$\text{Prob}[\text{no arrival in } (0,t_0)] = e^{-\lambda t_0}$$
$$\text{Prob}[\text{arrival in } (t_0, t_0+t) | \text{no arrival in } (0,t_0)] = \frac{\int_{t_0}^{t_0+t} \lambda e^{-\lambda t}}{e^{-\lambda t_0}} = 1 - e^{-\lambda t} \tag{1.1}$$

Also, the probability of an arrival in $(0,t)$ is

$$\int_0^t \lambda e^{-\lambda t} dt = 1 - e^{-\lambda t} \tag{1.2}$$

Therefore, from Eqns. 1.1 and 1.2 the conditional distribution of inter-arrival times given that certain time has elapsed is the same as the unconditional distribution.

1.6.4 Markov Process

A stochastic process is termed as a Markov process if it satisfies Markovian property, i.e., stochastic behavior of the process in which the future is only dependent on the present state but independent of the past progress. Mathematically, it is expressed as

$$P\{X(t_n+s) \leq x | X(t_1) = x_1, X(t_2) = x_2, ..., X(t_n) = x_n\}$$
$$= P\{X(t_n+s) \leq x | X(t_n) = x_n\}, s > 0; 0 \leq t_1 < t_2 < ... < t_n$$

1.6.5 Markov Chain

If the state space S is discrete, i.e., finite or countable infinite set and whose index set is $T = \{0, 1, 2, ...\}$, the Markov process is called the Markov chain.

Continuous-Time Markov Chain

Consider a continuous-time stochastic process $\{N(t), t \geq 0\}$ with discrete states $\{0,1,2,...\}$ it is known as a continuous-time Markov Chain if the following condition is satisfied

$$P(N(t+h) = j | N(0) = i, N(x) = i_x, 0 \leq x < t) = P(N(t+h) = j | N(t) = i) \; \forall \; h \geq 0$$

Discrete-Time Markov Chain

A discrete-time stochastic process $\{N(t), t = 0,1,2,...\}$ with discrete states $\{0,1,2,...\}$ is called Markov chain if the equation

$$\begin{aligned} P(N(t+1) = j | N(0) = i_0, N(1) = i_1, N(2) = i_2, ..., N(t) = i) \\ = P(N(t+1) = j | N(t) = i) = P_{ij}(t) \end{aligned}$$

is satisfied for all possible states of i_0, i_1, i_2,...,i_{t-1}, i, j and $t \geq 0$. $P_{ij}(t)$ is known as the transition probability for the process from the state i at time t to state j at $t+1$.

1.6.6 Renewal Process

Let $\{X_t\}$ be independent identically distributed (iid) non-negative random variables, $X_t \sim F(t)$ an arbitrary distribution. Then, the counting process

$$N(t) = \max\{n | S_n = X_1 + X_2 + X_3 + ... + X_n < t\} \tag{1.3}$$

is called renewal process. The mean number of events $m(t)$ on $(0,t)$ is called the renewal function

$$E[N(t)] = m(t) \tag{1.4}$$

Renewal process generalizes the Poisson process by allowing the inter-occerance time between two successive events to be independent and identically distributed (iid) random variable having an arbitrary distribution.

1.6.7 Birth and Death Process

Mathematically, a continuous-time Markov chain $N(t)$ with state space $\Omega = \{0,1,2,...\}$ is called a birth and death process if the following axioms are satisfied

$$\text{Prob}[N(t+h)-N(t)=k|N(t)=n]=\begin{cases}\lambda_n h+o(h); & k=1,\ n\geq 0\\ \mu_n h+o(h); & k=-1,\ n\geq 1\\ 1-(\lambda_n+\mu_n)h+o(h); & k=0,\ n\geq 1\\ 0; & \text{otherwise}\end{cases}$$ where,

$\lambda_n, n=0,1,2...$ are positive constants called birth rates, $\mu_n, n=1,2,...$ are positive constants called death rates and

$$\lim_{h\to 0}\frac{o(h)}{h}=0$$

1.6.8 Chapman-Kolmogorov Equation

Using the Markov property of the process, the Chapman-Kolmogorov equation gives multi-step transition probability from state i to state j over all possible k values and is expressed by

$$P_{ij}(t+s)=\sum_{k=0}^{\infty}P_{ik}(t)P_{kj}(s)$$

This equation describes that in order to move from state i to state j in time t, $X(t)$ moves to state k in time t and then from k to j in the remaining time s.

1.6.9 Quasi-Birth and Death Process

A Markov chain with the state-space

$$\Omega=\{(n,j);\ 1\leq j\leq n_p\ \&\ n\geq 0\}$$

is known as quasi-birth and death process, where the state space is divided into different levels and phases such that the level has n_p phases for each n. In a quasi-birth and death(QBD) process, the transitions are allowed between the adjacent states only. Therefore, a QBD process can be observed as a generator matrix in following way

$$\mathbf{Q}=\begin{bmatrix}\mathbf{A}_0 & \mathbf{B}_0 & \mathbf{0} & \mathbf{0} & \cdots\\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} & \cdots\\ \mathbf{0} & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B}_2 & \cdots\\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 & \mathbf{A}_3 & \cdots\\ \vdots & \vdots & \vdots & \vdots & \ddots\end{bmatrix}$$

where, each sub-matrix $\mathbf{B}_n$ can be obtained by the transitions from n^{th} level to $(n+1)^{\text{th}}$ level for $n \geq 0$. Similarly, the sub-matrices $\mathbf{C}_n$ can be generated by balancing the transitions from n^{th} level to $(n-1)^{\text{th}}$ level for $n \geq 1$ while the diagonal sub-matrices $\mathbf{C}_n$ are encoded within the n^{th} level for $n \geq 0$.

1.7 Methodological Aspects

1.7.1 Probability Generating Function

The joint probability distribution function $P_{n,j}$ represents the long-run fraction of time that the system remains in state $(N = n, J = j)$. Let $\{P_{n,j}, n \geq 1 \& j \geq 0$ be the stationary distribution of the Markov chain $\{N(t), J(t), t \geq 0\}$. Let $\Pi_j(z), j \geq 0$ be the partial generating functions which are given as follows

$$\Pi_j(z) = \sum_{n=1}^{\infty} z^n P_{n,j}, j \geq 0;$$

1.7.2 Matrix Analytic Method

The matrix analytic method is a procedure to determine the stationary probability distribution of a Markov chain which has a reiterating structure after some point and a unbounded state space in no more than one dimension. Such models are often designated as $M/G/1$ type Markov chains because they can figure transitions in an $M/G/1$ queueing model. The method is a more intricate form of the matrix geometric method and is the classical solution technique for $M/G/1$ chains. A stochastic matrix of an $M/G/1$ type is one of the form

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \dots \\ \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 & \dots \\ \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 & \dots \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_0 & \mathbf{A}_1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where $\mathbf{B}_i$ and $\mathbf{A}_i$ are $k \times k$ matrices, if $\mathbf{Q}$ is irreducible and positive recurrent then the stationary queue-size distribution is specified by the solution to the equations

$$\mathbf{QP} = \mathbf{P} \text{ and } \mathbf{e}^T\mathbf{P} = 1 \tag{1.5}$$

where e epitomizes a vector of suitable dimension with all values equal to 1. Matching the dimensional structure of $\mathbf{Q}, \mathbf{P}$ is partitioned to $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, ...To calculate these probabilities, the column stochastic matrix G is computed such that

$$G = \sum_{i=0}^{\infty} \mathbf{G}^i \mathbf{A}_i$$

$\mathbf{G}$ is called the auxiliary matrix. Matrices are defined

$$\bar{\mathbf{A}}_{i+1} = \sum_{j=i+1}^{\infty} \mathbf{G}^{j-i-1} \mathbf{A}_j \tag{1.6}$$

$$\bar{\mathbf{B}}_i = \sum_{j=i}^{\infty} \mathbf{G}^{j-i} \mathbf{B}_j \tag{1.7}$$

then $\mathbf{P}_0$ is found by solving

$$(\mathbf{e}^T + \mathbf{e}^T (\mathbf{I} - \sum_{i=1}^{\infty} \bar{\mathbf{A}}_i)^{-1} \sum_{i=1}^{\infty} \bar{\mathbf{B}}_i) \mathbf{P}_0 = 1$$

and hence,

$$\mathbf{P}_i = (\mathbf{I} - \bar{\mathbf{A}}_1)^{-1} [\bar{\mathbf{B}}_{i+1} \mathbf{P}_0 + \sum_{j=1}^{i-1} \bar{\mathbf{A}}_{i-j+1} \mathbf{P}_j], i \geq 1$$

1.7.3 Eigenvalue and Eigenvector

Let $\mathbf{Q}$ be any square matrix. A scalar λ is referred as an eigenvalue of $\mathbf{Q}$ if there exists a non-zero (column) vector $\mathbf{P}$ such that

$$\mathbf{QP} = \lambda \mathbf{P} \tag{1.8}$$

Any vector satisfying the Eqn. 1.8 is called an eigenvector of $\mathbf{Q}$ corresponding to the eigenvalue λ.

1.7.4 Laplace Transform

Assume $f(t)$ be a real-valued function of real variable t, defined for $t > 0$. Let s be a variable that assume to be real, and consider the function $\bar{F}(s)$ defined by

$$\bar{F}(s) = \int_0^\infty e^{-st} f(t) dt \tag{1.9}$$

for all values of s for which this integral exists.The function $\bar{F}(s) = L\{f(t)\}$ by the integral is called the Laplace transform of the function $f(t)$. With the help of Laplace transform, the system of differential equations with initial conditions is transform into system of linear equations which is computationally easy to solve.

1.7.5 Quasi-Newton Method

The QN method provides the optimal results in two steps. First, we compute a search direction p^t, which indicates the direction of the input space (vector including initial values of system design parameters) at iteration t. The second step determines how far we have to move in this direction by computing a step length $\alpha^t \in \mathbb{R}_+$. Therefore, it is an optimization method that searches for optimality with a descent direction.

$$p^t = -(H^t)\nabla f(x^t) \tag{1.10}$$

We then obtain the next iterate as

$$x^{t+1} = x^t + \alpha^t p^t \tag{1.11}$$

Here, the Hessian approximation $B^t \simeq (H^t)^{-1}$ must satisfy the quasi-Newton condition called secant equation.

$$B^t(x^{t+1} - x^t) = y^t \tag{1.12}$$

where, $y^t = \nabla f(x^{t+1}) - \nabla f(x^t)$ in which $f : D \to \mathbb{R}$ is continuously differentiable function on the domain D and $\nabla f(x^t) \in \mathbb{R}^n$ denotes the gradient of f at x^t.

Further, instead of computing the actual Hessian in the quasi-Newton method, we approximate the Hessian with the help of a positive definite symmetric matrix $B^t \in \mathbb{R}^{n \times n}$, which is updated at every iteration as

$$B^{t+1} = B^t + U$$

Now, we utilize the concept of the popular BFGS-method to compute the matrix U as

$$U = \frac{y^t (y^t)^T}{(y^t)^T (s^t)} - \frac{(B^t s^t)(B^t s^t)^T}{(s^t)^T B^t s^t} \tag{1.13}$$

where, $s^t = \alpha^t p^t$ and $(y^t)^T$ represents transpose of y^t.

When the analytically computed Jacobian $J(x^t)$ is used in place of B^t, the original Newton's method is recovered. The primary difference between Newton and the QN method is that Newton method uses the exact Jacobian matrix while the QN method uses approximated results. Therefore, the QN method is more famous for feasible super-linear convergence and is not calculated the Jacobian if some of the involved functions are twice continuously differentiable and strongly non-convex or convex [231].

Algorithm 1 Pseudo code for the quasi-Newton method

1: **Initialize:** starting point x^0, B^0, and t_{max};
2: **for** $t < t_{max}$ **do**
3: solve $B^t p^t = -\nabla f(x^t)$ using Eqn. (1.10) ;
4: step size $s^t = \alpha^t p^t$ (line search along p^t);
5: update iteration $x^{t+1} = x^t + s^t$ according to Eqn. (1.11)
6: update $B^{t+1} = B^t + U$, where U is given by Eqn. (1.13)
end for
7: **Output:** B^{new} and x^{new}

1.8 Nature-Inspired Optimization Techniques

Real-world numerical optimization problems have become increasingly challenging and complicated, necessitating effective optimization techniques. The derivative-based classical optimization techniques are unsuitable for such high grades of complex problems. The quasi and metaheuristic methods are newly developed optimization techniques used for multi-variables, multi-modal, discrete-continuous complex problems. The primary purpose of the metaheuristic technique is to explore the solution space effectively and efficiently rather than only finding optimal or non-optimal solutions. Some major metaheuristic optimization techniques that are used throughout the research work are briefly explained in the next subsections.

1.8.1 Archimedes Optimization Algorithm

Archimedes optimization algorithm (AOA) is devised with inspirations from Archimedes's principle, an interesting law of Physics. This principle describes the relationship between a

buoyant force and an object submerged in a fluid. It imitates that the weight of the displaced fluid is proportionate to the buoyant force exerted upward on an object partially or fully immersed in a fluid. AOA is a high-performance optimization technique in terms of convergence speed and exploration-exploitation balance, as it effectively solves complicated problems. A trade-off balance between exploration and exploitation is always crucial for metaheuristic algorithms, among other features. This feature makes AOA well suited for solving complex optimization problems with multiple locally optimal solutions. It retains a population of solutions and investigates a vast region to identify the best global solution [77]. For the detailed study of the AOA algorithm, refer the Chapter 2, Section 2.5.

1.8.2 Teaching Learning Based Optimization Algorithm

The teaching learning based optimization algorithm (TLBO) method is based on the impact of the teacher's influence on the output of students in a classroom. In this case, the result is measured in terms of grades or outcomes. A teacher is often thought of as a highly learned person who shares their expertise with learners. The quality of a teacher has an impact on the students' outcomes. A competent teacher prepares students to achieve better achievements in terms of grades or marks.

TLBO is a population-based strategy that progresses to the global answer through a population of solutions. The population is referred to as a group of learners or a class of learners in TLBO. The population of optimization algorithms is comprised of various design variables. As in other population-based optimization approaches, different design variables will be equivalent to different subjects offered to learners. The learners' outcome will be analogous to the 'fitness'. The teacher is considered the best solution so far. The TLBO procedure is split into two stages. The first part is called the 'Teacher Phase,' while the second is the 'Learner Phase.' The 'Teacher Phase' refers to learning from the teacher, whereas the 'Learner Phase' refers to learning through peer interaction. For the detailed study of the TLBO algorithm, refer the Chapter 3, Section 3.6.

1.8.3 Grasshopper Optimization Algorithm

For all living things, survival comes first. They have been changing and adapting in many ways to reach this goal. As the best and oldest optimizer on the planet, nature is an excellent place to look for inspiration. Exploration and exploitation are the two logical segments of the search process of nature-inspired algorithms. The fundamental concept in grasshopper optimization algorithm (GOA) is that larvae with limited mobility are utilized for local exploitation, adults with high mobility are used for global exploration, and the grasshopper's

location is the optimal solution to solve the optimization problem. GOA uses mathematics to simulate and replicate the natural behavior of grasshopper swarms to solve optimization problems. The GOA developed by Saremi et al.[161] in 2017. For the detailed study of the GOA algorithm, refer the Chapter 4, Section 4.6.

1.8.4 Grey Wolf Optimizer

Grey wolf optimizer (GWO) is a recently developed metaheuristic optimization technique inspired by leadership hierarchy and the hunting mechanism of grey wolves. GWO is a swarm intelligence method wherein leadership hierarchy is simulated by four categories of grey wolves. The three primary phases of hunting: (i) searching, (ii) surrounding, and (iii) attacking, the prey are employed for more extensive exploration and local search exploitation of search space. Compared to other metaheuristic techniques, the main advantage of GWO is how quickly it converges [136], [58], [67]. It is suitable for discrete and continuous domains simultaneously. The circumferential system ensures rapid and accurate convergence. For more comprehensive exploration and local exploitation of promising search space, the three main stochastic progressions of hunting: searching, encircling, and attacking the prey, are used. For the detailed study of the GWO algorithm, refer the Chapter 5, Section 5.6.

1.8.5 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is an agent-based optimization technique, which was firstly familiarized by Kennedy and Eberhart in 1995 [104] having been inspired by swarm intelligence and its movement. When birds (particles) fly in a flock (swarm) to search for food randomly, they share information about what they find among themselves and help the entire flock get the best hunt. The roaming nature of birds in the flock will inspire the exploration phase of the optimization procedure, which aims to avoid being stuck in the local region. For the detailed study of the PSO algorithm, refer the Chapter 6, Section 6.7.2.

1.8.6 Cuckoo Search

Generally, cuckoos are captivating birds, not just for their beautiful sounds but also for their aggressive reproduction method. Most of the cuckoo species lay their eggs in communal nests, yet they may throw down the eggs of others to maximize the chances of their eggs hatching. Nevertheless, some species practice obligatory brood parasitism, which involves laying their eggs in the nests of other host birds. Some cuckoo species have evolved due

to genetic variation where female parasitic cuckoos are capable of imitating the color and pattern of eggs of certain host species. The behavior lessens the likelihood of their eggs forsaking, increasing their reproductive potential. The competitiveness between cuckoos and host species forms a combat system where cuckoos' eggs can be exposed and thrown down with a probability of P_*. The cuckoo search (CS) algorithm was given by Xin-She Yang and Suash Deb [219] in the year 2009. For the detailed study of the CS algorithm, refer the Chapter 6, Section 6.7.3.

1.8.7 Social Group Optimization

The inspiration for the population-based Social group optimization (SGO) algorithm developed by Satapathy and Naik [163] in 2016, comes from the concept of the social behavior of human beings toward solving complex tasks in life. There are a number of behavioral traits that humans possess to solve their problems in life. Individuals sometimes find these problems too complex to solve alone and form groups to solve them with the influence of one another's traits. On the basis of the idea that solving a given complex problem in a group comes out to be more effective and efficient than individuals in exploiting and exploring their different traits. Also, it has been observed that living entities imitate or follow their surroundings and so human beings as well mimic the knowledge sharing concepts in solving any task by observing others who are better than them. A person's fitness value corresponds to their ability to solve a problem in SGO. Consequently, the person with the best fitness value enhances the knowledge of the entire group. For the detailed study of the SGO algorithm, refer the Chapter 8, Section 8.5.

1.9 Literature Review

1.9.1 Historical Development of Queueing Theory

The study of telegraphic problems around the turn of the 20th century with the work of Erlang [55] and Engset [54] is the genesis of the field now known as queueing theory. James R. Jackson [88] extended Erlang's model of a single queueing station to a system of networked queueing stations. The pioneering work by Naor [140] on service design with strategic customers includes consideration of an observable $M/M/1$ queueing system with homogenous customers and analyzed of revenue-maximizing price and socially optimal price. The objective of socially optimum decision-making is to minimize the total system cost, which is an aggregate function of the costs incurred by both the service facilities and the customers who use them. Typically, expenses associated with facility opening and service are seen from the

perspective of the facilities, whereas costs associated with access and congestion are viewed from the perspective of customers. It is clear that the first factor takes the system owner's cost into account, while the second factor takes service quality into account [84].

General introductory books in the area of queueing theory and stochastic processes are by Cinlar [36], Cohen [37], Gross [69], Shortle et al. [174], Kleinrock [107], whereas Takagi and Boguslavsky [180] provide a bibliography of books on queueing analysis and performance evaluation. Computational algorithmic approaches of computer and stochastic systems, in general, are treated by Neuts [144], Conway and Georganas [38] and Tijms [185, 186].

1.9.2 Queue with Customer Impatience Behavior

A seminal contribution to queueing theory was made by Conny Palm [147] by introducing the option of customer abandonment. Haight [72, 74] rediscovered the balking and reneging behavior of customers for the single server Markovian queueing problem. Kumar [114] is the first researcher who introduced the efficient notion of retention of the reneging customer. Later, many researchers (*cf.* [108], [177], [117], [23], [115]) investigated retention of the reneging customer in the service sector in economic perspective. One of the earliest articles dealing with jockeying in a two-station, the parallel system, is owing to Haight [73] who analyzed a system in which arriving customers always join the shortest queue initially.

1.9.3 Queue with Arrival Control Policy

The control of arrivals in the service systems helps in congestion reduction, which is addressed through F-policy. Gupta [71] was the first to provide steady-state analytical solutions for the F-policy $M/M/1/K$ queueing system with an exponential startup time. The methodological characteristics of the F-policy employed in the retrial queueing model, the vacation and working vacation model, the unreliable server model, and the non-Markov model were outlined by Jain et al. [93] to give a state-of-the-art of admission control F-policy. Since then, the study of controllable F-policy has been modified by many researchers, and extensive research has been carried out to enhance the use of arrival control policy in various queueing and machining systems ([223], [206], [205], [133], [217]).

1.9.4 Queue with Vacation

The strategic vacation policy variants include multi-vacation, single-vacation, working vacation, Bernoulli vacation, gated vacation, N-policy etc. Server vacation in the queueing system takes place due to several reasons, including a low workload, maintenance time, the

failure to repair, and many more. In recent years, there has been considerable research on customer impatience attributes in queueing systems with strategic server vacations/failures. Levy and Yechiali [118] came up with the idea of the vacation queueing paradigm. A thorough, excellent, and exhaustive study of vacation queueing models is found in Doshi's survey [44], as well as in several publications on vacation queueing models ([181], [184], [8]).

1.9.5 Retrial Queue

A retrial queueing system allows customers who find all servers occupied to join a virtual queue called retrial orbits and retry for service after a random length called retrial time. An analysis of queueing economics in retrial queues was first carried out by Wang and Zhang [194]. An overview of retrial queue theory in real-world call centers and cellular networks systems may be found in [153] by Tuan. References [14], [106], [57], [42], [2] provide a comprehensive survey of retrial queueing systems.

1.9.6 Queue with Server Breakdown

The literature on queue-based service systems is rich with assumptions about reliable servers, which is seldom. The service provider is subject to breakdowns randomly at any instant in practice. Most research findings on queueing-based service systems with server breakdown consider that the server terminates working completely when the breakdown occurs. Nevertheless, in practice, some real-time systems exist in which the service provider still works at a lesser rate in breakdown state, which referred to working breakdown or partial breakdown in the queueing literature (*cf.* [178], [96], [96], [119], [124]) studied the single server Markovian queue with working breakdown. A detailed survey on queueing-based service systems with the breakdown of the server is provided by Krishnamoorthy et al. [111]. Liou [122] explored the matrix method for a single server queue with customer impatience and servers' working breakdown.

The breakdown of the server leads to massive congestion or high impatience attributes among the customers, which increases the economic losses, customer dissatisfaction, etc. The breakdown of the service facility needs strategic recovery. The concept of threshold recovery policy was firstly introduced by Efrosinin and Semenova [53]. Jain and Bhagat [89] envisaged a finite capacity retrial queueing-based service system with a threshold recovery policy for unreliable servers. Yang et al. [212] formulated a cost optimization problem for a threshold-based recovery policy for repairable $M/M/1/N$ system.

1.10 Gaps in the Existing Literature

1. The measures are required to control and regulate service mechanisms to avoid server idling, which would lead to savings of resources and increase the efficiency and throughput of the system.

2. The existing work on customer impatience behavior can be further extended by considering imperfect service, working vacation, unreliable server, working breakdowns, or threshold-based control policies viz F-policy to control admission or N-policy to control the starting of service.

3. For analyzing the non-Poisson inputs and non-exponential services time distributions such as renewal processes, general distributions, matrix-exponential distributions, and heavy-tailed distributions, and discussing the bulk arrival processes and/or the bulk service processes, the effective algorithms for the performance measures are necessary and interesting and need to develop.

4. The preemptive or non-preemptive priority, service interruption, and multi-phase optional repair facilities have not been investigated much despite versatile in depicting the many real-time congestion situations encountered in industrial and day-to-day problems.

5. Retrial queue, interjecting of the customer, faffing delay, setup delay are some crucial issues that need to study to make the service system more functional.

6. Vacation queueing system with F-Policy and vacation interruption can be further extended to more threshold-based service control policies, namely N-policy, T-policy, for finite queueing models incorporating various queueing terminologies such as geometric abandonment and feedback policy, or non-Markovian models with variant vacation.

1.11 Thesis Objectives

1. To explore a new regime of quality service.

2. To develop the mathematical Markovian models for some governing service system/waiting problem associated with several realistic congestion problems.

3. To analyze the effects of customer's behavior, server's constraints, service mechanism, and architectural limitation on the queueing system.

4. To implement state-of-the-art methodology and techniques for optimal and sensitivity analysis.

5. To establish the critical design parameter(s) of the governing models.

1.12 Organization of the Thesis

Having discussed the main objective and scope of the thesis, this section provides a brief thematic overview of the chapter-wise road map.

Chapter 1 covers the introduction of service systems and ways to improve their quality, along with the other queueing characteristics.

Chapter 2 provides customers' impatient attributes in congestion using aqueueing theoretic approach. The proposed model in this chapter contemplatesthe existence of impatient customers within a classical queueing system.

Chapter 3 presents a service system consisting of two types of servers: multi-subordinateservers working in parallel and one chief server. This service strategy study is applicable to various managerial systems.

Chapter 4 provides a two-stage service system wherein arriving customers in the first stage can either join the queue and wait for their turn or directly seek service through the online app. The controllable online booking is conceptualized for online-app users.

Chapter 5 focuses on optimal policies for an efficient service system since the congestion of customers more often originates from degraded policies than faulty arrangements. This chapter presents a notion of unreliable service and an F-policy for stochastic modeling of a finite-capacity customer service system.

Chapter 6 analyzes a finite capacity service system consisting of several realistic queueing characteristics, namely, impatient customers, partial server breakdowns, and threshold-based recovery policies.

Chapter 7 presents the critical issue of the single-server congestion problem with prominent customer impatience attributes and server strategic differentiated vacation. Despite their apparent practical relevance, the proposed congestion problem has yet to be studied from a service or production perspective with transient analysis. The queue-theoretic approach is used for mathematical modeling.

Chapter 8 provides the orbital search concept in Markovian retrial queueing, including multiple vacation policies and server breakdown, which is described by an infinite number of inflow-outflow balanced equations.

Finally, **Chapter** 9 summarizes the thesis work with an emphasis on the major contributions and future recommendations.

Chapter 2

Cost Analysis of Customer's Impatience Attributes in the Service System

This chapter uses a queueing-theoretic approach to deal with the customers' impatient attributes in congestion. Upon arrival, strategic customers initially either balk or join one of the queues selectively and decide at subsequent arrival and departure epochs whether to renege or jockey in a probabilistic manner with the aim of reducing expected waiting time. We consider the simultaneous effect of customers' impatience behaviors like balking, reneging, and jockeying and reveal fascinating facts about customers' behavior in waiting queues.

2.1 Introduction

Congestion is ubiquitous. During congestion, queueing systems interplay between the customers and the service provider. The theory of queueing systems was created and developed to forecast the behavior of service systems subject to random demand from prospective customers. In today's scenario, everything needs to be served at a faster pace due to high competence in socio-techno-economic constraints and the need of the hour, so cost and time have become important factors. Customers value their time and often lose impatience when they are delayed while waiting in a queue for service. This article extensively studies queueing systems with customer impatience behavior due to their potential applications in real-life congestion problems. The examples can be observed in hospital emergency rooms making vital patient treatment decisions; inventory systems that store perishable goods; queues arising in telecommunication networks, call centers, cloud computing, wireless sensor networks, and machine repair problems.

The impatience attributes of customers are observed through the customer actions in the waiting line, who may balk, renege, or jockey. Impatience is the most prominent feature of a service system when an individual wants to experience service but needs to queue. For a long waiting time, customer abandonment, such as reneging and balking, has been a significant concern in queueing systems in view of revenue, goodwill, incurred cost, etc. Thus, customers' impatient attributes should be involved in studying the service system to model a more realistic queueing model. Various impatient characteristics of customers, which we will be studying in this chapter, are classified as balking, reneging, and jockeying.

Modeling a service system consisting of the impatient behavior of customers using a queueing theoretic approach presents more challenges. Behavioral operations explore how servers and customers act in a functioning setting characterized by a patient threshold limit. In particular, a customer decides to leave the system without completing service when waiting time has crossed the patience threshold value, adversely affecting the firm's economic goodwill. Thus, predictive measures need to be taken to minimize the overleap and retain impatient customers in the system using specific customer retention mechanisms. Shekhar et al. [170] presented the realistic retaining policy of reneged customers under Bernoulli's scheduled modified vacation for the multi-server finite capacity queueing system. There has been an emerging trend to study the queueing model from an economic viewpoint under socio-techno constraints considering customers' strategic behavior to get the maximum benefit from a service system during the last few decades.

Haight [72], [74] first introduced customers' balking and reneging behavior for the single server Markovian queueing problem. One of the earliest articles dealing with jockeying

in a two-station, the parallel system, is owing to Haight [73], who analyzed a system in which arriving customers always join the shortest queue initially. Henceforth, empirical research has been done on customers' impatient behavior in waiting. Zhao et al. [229] dealt with jockeying in the shortest queue. The evolution of literature emerged from the study on queue joining by Naor [140], which was summarized in a book by Hassin and Haviv [81]. In reality, decisions to leave the system are exaggerated by the timely information, announcements, reviews, feedback, and dynamics of operational services, such as queue length and the nature of service flows. Eminent research provides more supportive evidence for a different types of service systems such as emergency departments at a hospital (Batt and Terwiesch [20], Bolandifar et al. [21]), call-center (Zohar et al. [232]), finite capacity (Tarabia [183]), telecommunications (Zhao and Grassman [229], Xu and Zhao [208]). Jockeying can increase customers' switching time between facilities, and on the other hand, allowing jockeying will reduce customers' waiting time in their queues considerably. Balancing the switching time drawbacks, jockeying should be employed in a system whose servers have a reasonable distance to allow customers to switch their lines. Some notable contributions are owing to [5], [160], [29], [41], [166].

The literature on optimal control of parallel service stations with jockeying could be more extensive. Recently, Ravid [160] considered a two-server in parallel service facility in which the arriving customer is assigned the server according to "join the shortest queue with threshold jockeying" rule. In a generalized queueing network (G-network) wherein signal entities are assumed to arrive in the system externally according to a Poisson process in addition to the regular customers is a useful way of modeling the queueing behavior during relief distribution due to the flexibility and computational efficiency using product form results. Ozen and Krishnamurthy [146] used routing parameters and signal entities of the G-network to model the jockeying of victims' movement between relief centers during a disaster. Dehghanian and Kharoufeh [40] minimized the total expected discounted jockeying and holding costs over finite and infinite time horizons by establishing the optimal joining and jockeying policies for the strategic customer who seeks service in a parallel queueing system. Wang et al. [200] investigated the serviceability dynamics of a busy period for an $M/M/c$ multi-server queueing system with impatient customers who may balk or renege.

Non-smooth composite convex optimization has been widely used in the real world. It entails using one or more non-smooth regularizers in several state-of-the-art techniques. Traditional gradient-based techniques cannot solve such problems due to ill-conditioned optimization problems. The quasi-Newton method is gaining traction due to its effectiveness in dealing with such problems and affine invariance [187]. Quasi-Newton methods are also popular since they have local superlinear convergence and don't require the Jacobian to be

computed. However, because the quasi-Newton direction may differ from the descent direction of the norm square metric function, global convergence of quasi-Newton techniques for nonlinear equations is challenging to establish [231]. Queueing theorists also employed the quasi-Newton method to determine optimal decision parameter(s) due to computational richness for the explored constrained and non-constrained queueing models. The noteworthy contributions in the literature for the optimum analysis of queueing models using the quasi-Newton technique are given by researchers (*cf.* [202], [203], [226] and [169]). We also employ the quasi-Newton technique to determine the value of the studied model's governing parameter(s) so that incurred cost is minimum.

The primary goal of any service system is to choose decision variables that meet all criteria while having the lowest possible cost, i.e., the main goal is to comply with basic standards while also achieving economic designs. In science and engineering, metaheuristics give acceptable solutions in a reasonable time for tackling complicated problems. Nature-inspired metaheuristic approaches are now widely employed in various scientific, computing, and engineering applications since they effectively solve complex problems. Using Darwin's theory of survival of the fittest, metaheuristic algorithms have imitated the behavior of physical and biological systems in nature. This chapter uses the recently developed Archimedes optimization algorithm (AOA) to achieve the best predicted total cost with the best values of decision parameters. We compare the results to those achieved using the well-known heuristic methodology quasi-Newton method. Archimedes optimization algorithm is devised with inspiration from an interesting law of Physics known as the Archimedes principle. It mimics the buoyant force principle, which states that the buoyant force exerted upward on an item wholly or partially submerged in a fluid is proportionate to the weight of the displaced fluid [77].

Besides, to the best of our knowledge, modeling queueing systems with impatient behavior taking simultaneously balking, reneging, and jockeying has yet to be attempted in the literature. Thus, this chapter contributes to this sense.

Against this background, the remainder of this chapter is structured in the following manner. Section 2.2 discusses the developed model and defines its various states. In Subsection 2.2.1, we describe the assumptions and notations. Steady-state governing equations are derived in Subsection 2.2.2. In Section 2.3, we brief various system performance measures using mathematical expressions. Section 2.4 presents the construction of the cost function. The Archimedes optimization algorithm is elaborated thoroughly in Section 2.5. The model's numerical results and optimum analysis are presented in Section 2.6. Section 2.7 offers the conclusion of this research and future work.

2.2 Model and State Description

The queueing-theoretic approach is essential for modeling and performance evaluation of service systems. Our model considers an $M/M/2$ finite capacity queueing system with two heterogeneous servers: Server 1 and Server 2, arranged in parallel. The capacity of each of the servers is K, so the system capacity becomes $2K$.

2.2.1 Assumptions and Notations

The major assumptions of the queueing problem contemplating random impatience behavior and notations for the investigated model are reviewed as follows:

Arrival Process

- The customers arrive at the service system according to a Poisson process, with an arrival rate λ.
- If the server is idle, the arriving customer gets service immediately; otherwise, he must wait in the queue.

Service Process

- Customers are served on FCFS basis queue discipline by two heterogeneous servers in parallel.
- The service times of all customers are independent and identically distributed random variables which follow the exponential distribution with service rate μ_1 and μ_2 for the first and second server, respectively.
- Each customer does not have any priority over any other customer.

Impatience Attributes

- The arriving customer may join the shortest queue with joining probability ξ or may balk away with complementary probability $1-\xi$.
- When a customer joins one of the queues, he may leave the system without being served after a subsequent random time interval due to a long-expected waiting time. The time-to-renege follows the exponential distribution with the reneging rate ν. The

customers leave the system either by reneging or after completing service from one of the two servers.

- When the difference between the number of customers in the queue becomes greater than one, the customers from a long line may join the adjacent queue at the last position and leave their position in the current queue. The time-to-jockey follows the exponential distribution with mean jockeying rate η.
- All customers are impatient with the same threshold limit except the first customer for each server.
- The model under consideration is of finite capacity $2K$; each customer who crosses the threshold $2K$ is deemed a lost customer.

All occurrences, such as arrival, service, balking, reneging, jockeying, and loss of customers, are statistically independent.

2.2.2 Steady State Differential Equations

We use the following notations to express the distinct states at any instant t for the stochastic modeling of the examined queueing model.

$$N_1(t) \equiv \text{Number of customers in front of server 1 at time } t$$
$$N_2(t) \equiv \text{Number of customers in front of server 2 at time } t$$

Then, a continuous time Markov chain (CTMC) $(N_1(t), N_2(t); t \geq 0)$ on the state space

$$\Omega = \{(n_1, n_2) \mid n_1 = 0,1,2,...,,K-1,K;\ n_2 = 0,1,2,...,K-1,K\}$$

As $t \to \infty$, the system approaches to steady-state. The governing steady-state probabilities are defined as follows

$$P_{n_1,n_2} = \lim_{t\to\infty}\{N_1(t) = n_1, N_2(t) = n_2;\ n_1 = 0,1,2,...,K-1,K\ \&\ n_2 = 0,1,2,...,K-1,K\}$$

Now for analyzing the studied server system, we construct steady-state Chapman-Kolmogrove forward equations for the system states as follows:

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

$$-(\lambda+\mu_2)P_{0,1}+\frac{\lambda}{2}P_{0,0}+\mu_1 P_{1,1}+(\mu_2+\nu)P_{0,2}=0 \tag{2.2}$$

$$-(\lambda+\mu_2+(n_2-1)\nu+(n_2-1)\eta P_{0,n_2}+\mu_1 P_{1,n_2}+ (\mu_2+n_2\nu)P_{0,n_2+1}=0;\ 2\le n_2\le K-1 \tag{2.3}$$

$$-(\lambda+\mu_2+(K-1)\nu+(K-1)\eta)P_{0,K}+\mu_1 P_{1,K}=0 \tag{2.4}$$

$$-(\lambda+\mu_1)P_{1,0}+\frac{\lambda}{2}P_{0,0}+\mu_2 P_{1,1}+(\mu_1+\nu)P_{2,0}=0 \tag{2.5}$$

$$-(\lambda+\mu_1+(n_1-1)\nu+(n_1-1)\eta)P_{n_1,0}+\mu_2 P_{n_1,1} +(\mu_1+n_1\nu)P_{n_1+1,0}=0;\ 2\le n_1\le K-1 \tag{2.6}$$

$$-(\lambda+\mu_1+(K-1)\nu+(K-1)\eta)P_{K,0}+\mu_2 P_{K,1}=0 \tag{2.7}$$

$$-(\lambda\xi+\mu_1+\mu_2)P_{1,1}+\lambda P_{0,1}+\lambda P_{1,0} +(\mu_2+\nu)P_{1,2}+(\mu_1+\nu)P_{2,1}+\eta P_{2,0}+\eta P_{0,2}=0 \tag{2.8}$$

$$-(\lambda\xi+\mu_1+\mu_2+\nu)P_{2,1}+(\mu_1+2\nu)P_{3,1}+\frac{\lambda}{2}\xi P_{1,1} +\lambda P_{2,0}+(\mu_2+\nu)P_{2,2}+2\eta P_{3,0}=0 \tag{2.9}$$

$$-(\lambda\xi+\mu_1+(n_1-1)\nu+(n_1-2)\eta+\mu_2)P_{n_1,1}+(\mu_2+\nu)P_{n_1,2} +(\mu_1+n_1\nu)P_{n_1+1,1}+\lambda P_{n_1,0}+n_1\eta P_{n_1+1,0}=0;\ \ 3\le n_1\le K-1 \tag{2.10}$$

$$-(\lambda\xi+\mu_1+\mu_2+(K-1)\nu+(K-2)\eta)P_{K,1}+(\mu_2+\nu)P_{K,2} +\lambda P_{K,0}=0 \tag{2.11}$$

$$-(\lambda\xi+\mu_1+\mu_2+\nu)P_{1,2}+\frac{\lambda\xi}{2}P_{1,1}+\lambda P_{0,2}+(\mu_1+\nu)P_{2,2} +(\mu_2+2\nu)P_{1,3}+2\eta P_{0,3}=0 \tag{2.12}$$

$$-(\lambda\xi+\mu_1+\mu_2+(n_2-1)\eta+(n_2-2)\nu)P_{1,n_2}+\lambda P_{0,n_2}+(\mu_1+\nu)P_{2,n_2} +(\mu_2+n_2\nu)P_{1,n_2+1}+n_2\eta P_{0,n_2+1}=0;\ 3\le n_2\le K-1 \tag{2.13}$$

$$-(\lambda\xi+\mu_1+\mu_2+(K-1)\nu+(K-2)\eta)P_{1,K}+(\mu_2+\nu)P_{2,K}+\lambda P_{0,K}=0 \tag{2.14}$$

$$-(\lambda\xi+\mu_1+(n_1-1)\nu+\mu_2+(n_2-1)\nu)+(|\zeta_1-\zeta_2|-1)\eta P_{n_1,n_2} +\Lambda_1(n_1,n_2)P_{n_1-1,n_2}+\Lambda_2(n_1,n_2)P_{n_1,n_2-1}+(\mu_1+n_1\nu)P_{n_1+1,n_2} +(\mu_2+n_2\nu)P_{n_1,n_2+1}+V_1(n_1,n_2)\eta P_{n_1-1,n_2+1} +V_2(n_1,n_2)\eta P_{n_1+1,n_2-1}=0;\ 2\le n_1\le K-1,2\le n_2\le K-1 \tag{2.15}$$

where,

$$\Lambda_1(n_1,n_2)=\begin{cases}0; & n_1-1>n_2\\ \frac{\lambda}{2}\xi; & n_1-1=n_2\\ \lambda\xi\ ; & n_1-1<n_2\end{cases}$$

$$\Lambda_2(n_1,n_2)=\begin{cases}0; & n_2-1>n_1\\ \frac{\lambda}{2}\xi; & n_2-1=n_1\\ \lambda\xi; & n_2-1<n_1\end{cases}$$

$$\zeta_1(n_1,n_2)=\begin{cases}0; & n_1=0\\ n_1-1; & n_1\geq 1\end{cases}$$

$$\zeta_2(n_1,n_2)=\begin{cases}0; & n_2=0\\ n_2-1; & n_2\geq 1\end{cases}$$

$$V_1(n_1,n_2)=\begin{cases}0\ ; & n_2-n_1+2\leq 0\\ n_2-n_1+2\ ; & n_2-n_1+2>0;\ n_1,n_2>1\end{cases}$$

$$V_2(n_1,n_2)=\begin{cases}0\ ; & n_1-n_2+2\leq 0\\ n_1-n_2+2\ ; & n_1-n_2+2>0;\ n_1,n_2>1\end{cases}$$

$$-[\lambda\xi+\mu_1+(n_1-1)\nu+\mu_2+(K-1)\nu+(K-n_1-1)\eta]P_{n_1,K}+\lambda\xi P_{n_1-1,K}$$
$$+(\mu_1+n_1\nu)P_{n_1+1,K}=0;\ 2\leq n_1\leq K-2 \tag{2.16}$$

$$-[\lambda\xi+\mu_1+(K-2)\nu+\mu_2+(K-1)\nu]P_{K-1,K}+\lambda\xi P_{K-2,K}(t)+\frac{\lambda\xi}{2}P_{K-1,K-1}$$
$$+(\mu_1+(K-1)\nu)P_{K,K}=0 \tag{2.17}$$

$$-[\lambda\xi+\mu_1+(K-2)\nu+\mu_2+(K-1)\nu]P_{K-1,K}+\lambda\xi P_{K-2,K}+\frac{\lambda\xi}{2}P_{K-1,K-1}$$
$$+(\mu_1+(K-1)\nu)P_{K,K}=0 \tag{2.18}$$

$$-[\lambda\xi+\mu_1+(K-1)\nu+\mu_2+(K-2)\nu]P_{K,K-1}+\lambda\xi P_{K,K-2}+\frac{\lambda\xi}{2}P_{K-1,K-1}$$
$$+(\mu_2+(K-1)\nu)P_{K,K}=0;\ 2\leq n_2\leq K-2 \tag{2.19}$$

$$-[\mu_1+(K-1)\nu+\mu_2+(K-1)\nu]P_{K,K}+\lambda\xi P_{K,K-1}+\lambda\xi P_{K-1,K}=0 \tag{2.20}$$

Equations 2.1-2.20 can be represented in the matrix form as

$$\mathbf{\Pi Q}=\mathbf{0} \tag{2.21}$$

where $\mathbf{Q}$ denotes the coefficient matrix of order $(K+1)^2$, $\mathbf{\Pi}$ is the column vector having all the state probabilities and $\mathbf{0}$ is the null column vector of order $(K+1)^2$. Following the law of total probability, the normalizing condition for state probabilities is given as below

$$\sum_{n_1=0}^{K}\sum_{n_2=0}^{K} P_{n_1,n_2} = 1 \tag{2.22}$$

We further use the normalizing condition of probability as

$$\mathbf{\Pi e} = 1$$

where $\mathbf{e} = [1,1,...,1]^T$ is column vector of dimension $(K+1)^2$ having all entries 1. Now, we express the system of linear equations 2.21 as

$$\mathbf{\Pi A} = \mathbf{B}$$

where, $\mathbf{A}$ denotes the matrix $\mathbf{Q}$ replacing the last row with a row vector having all elements 1 and $\mathbf{B}$ represents the column vector $[0,0,...0,1]^T$ of order $(K+1)^2$.

2.3 System Performance Measures

In this chapter, we analyze the sensitivity of the impatient attributes of the customers in the service system economically. For the performance characterization of the governing queueing model, there are some standard performance indices. We also employ performance measures to delineate the modeling and methodology for finite-capacity multi-server queueing systems with balking, reneging, and jockeying. The mathematical expression of performance indices is used to exhibit the parametric analysis for the decision purpose. These performance measures are correlated and recognized as increased importance in an optimal and sensitivity analysis of the service environment.

- The expected number of the customers in the system

$$L_S = \sum_{n_1=0}^{K}\sum_{n_2=0}^{K} (n_1+n_2)P_{n_1,n_2} \tag{2.23}$$

- The expected number of the customers in the queue

$$L_Q = \sum_{n_1=1}^{K}\sum_{n_2=1}^{K} (n_1+n_2-2)P_{n_1,n_2} \tag{2.24}$$

- The throughput of the system

$$\tau_p = \sum_{n_1=1}^{K} \sum_{n_2=1}^{K} (\mu_1 + \mu_2) P_{n_1,n_2} \tag{2.25}$$

- The effective arrival rate of the customer in the system

$$\lambda_{eff} = \sum_{n_1=1}^{K-1} \lambda \xi P_{n_1,n_1} + \sum_{n_1=2}^{K} \sum_{n_2=1}^{n_1-1} \lambda \xi P_{n_1,n_2} + \sum_{n_2=2}^{K} \sum_{n_1=1}^{n_2-1} \lambda \xi P_{n_1,n_2} + \sum_{n_1=0}^{K} \lambda P_{n_1,0} + \sum_{n_2=1}^{K} \lambda P_{0,n_2} \tag{2.26}$$

- The expected waiting time of the customer in the system

$$W_S = \frac{L_s}{\lambda_{eff}} \tag{2.27}$$

- The average balking rate of the customer

$$ABR = \sum_{n_1=1}^{K-1} \sum_{n_2=1}^{K-1} \lambda(1-\xi)) P_{n_1,n_2} + \sum_{n_1=1}^{K-1} \lambda(1-\xi) P_{n_1,K} + \sum_{n_2=1}^{K-1} \lambda(1-\xi) P_{K,n_2} \tag{2.28}$$

- The average reneging rate of the customer

$$ARR = \sum_{n_1=1}^{K} \sum_{n_2=1}^{K} (n_1+n_2-2) \nu P_{n_1,n_2} + \sum_{n_1=1}^{K} (n_1-1) \nu P_{n_1,0} + \sum_{n_2=1}^{K} (n_2-1) \nu P_{0,n_2} \tag{2.29}$$

- The frequency that the system is full

$$FF = \lambda \xi (P_{K-1,K} + P_{K,K-1}) \tag{2.30}$$

2.4 Cost Analysis

The state-of-the-art chapter objective is the optimal and sensitive economic analysis of customers' impatience attributes. For that purpose, we develop a total expected cost function per unit of time as the objective function for the $M/M/2/2K$ queueing model with balking, reneging, and jockeying. The main aim of the objective function is to determine the best decision parameter(s) value to minimize the total operational cost of the system. The parameters μ_1 and μ_2 are considered as decision variables. The various cost parameters related

to different states of the Markovian model are defined as follows:
$C_h \equiv$ holding cost for each customer present in the system.
$C_b \equiv$ balking cost for each customer who balks from the system.
$C_r \equiv$ reneging cost for each customer who reneges from the system.
$C_k \equiv$ the fixed cost for the system capacity.
$C_1 \equiv$ cost for service by server 1 of each customer in the system.
$C_2 \equiv$ cost for service by server 2 of each customer in the system.
The total expected cost function is given as follows

$$TC(\mu_1,\mu_2) = C_h L_S + C_b ABR + C_r ARR + C_k K + C_1\mu_1 + C_2\mu_2 \tag{2.31}$$

Hence, the governing optimization problem is developed as

$$TC(\mu_1^*,\mu_2^*) = min\{TC(\mu_1,\mu_2)\}$$

We opt for the metaheuristic and quasi-optimization techniques discussed in the coming sections to compute the optimal value of decision variables (μ_1^*,μ_2^*) and total cost TC^*.

2.5 Archimedes Optimization Algorithm

Real-world numerical optimization problems have become increasingly challenging and complicated, necessitating effective optimization techniques. The derivative-based classical optimization techniques are unsuitable for such high grades of complex problems. The quasi and metaheuristic methods are newly developed optimization techniques used for multi-variables, multi-modal, discrete-continuous complex problems. The primary purpose of the metaheuristic technique is to explore the solution space effectively and efficiently rather than only finding optimal or non-optimal solutions. In this chapter, we use a new metaheuristic algorithm called Archimedes optimization algorithm (AOA), a population-based algorithm to compete with the state-of-the-art and recent optimization algorithm for carrying out sensitivity analysis of the model.

2.5.1 Inspiration

AOA is devised with inspiration from Archimedes's principle, an interesting law of Physics. More details about this principle are already discussed in subection 1.8.1.

2.5.2 Mathematical Model and Algorithm

AOA commences the search process with the initial set of the population (candidate solution) with random volumes, densities, and accelerations.

Step 1 - Initialization

In the initial step, the position $(x(i))$, volume $(V(i))$, and density $(D(i))$ of object i is initialized as follows:

$$
\begin{aligned}
&x(i) = x_l(i) + rand \times (x_u(i) - x_l(i));\ i = 1,2,...,N \\
&V(i) = rand \\
&D(i) = rand \\
&A(i) = x_l(i) + rand \times (x_u(i) - x_l(i)
\end{aligned}
$$

where $x_l(i)$ and $x_u(i)$ are the lower and upper bounds of the search space, respectively and rand is a D-dimensional vector randomly generates uniformly distributed number between $[0,1]$.

Step 2 - Update volumes and densities

For iteration $t+1$, the density and volume are updated as:

$$
\begin{aligned}
V^{t+1}(i) &= V^t(i) + rand \times (V^{best} - V^t(i)) \\
D^{t+1}(i) &= D^t(i) + rand \times (D^{best} - D^t(i))
\end{aligned} \tag{2.32}
$$

where V^{best} and D^{best} are the volume and density connected with the best object, found so far.

Step 3 - Transfer operator and density factor

At the start, objects collide and sometimes attempt to get an equilibrium state. In AOA, transfer operator (TF) helps to transform search from exploration to exploitation as follows:

$$
TF = e^{\left(\frac{t - t_{max}}{t_{max}}\right)} \tag{2.33}
$$

where t and t^{max} are iteration number and maximum iterations, respectively.
In a similar manner, density decreasing factor d assists AOA on global to local search.

$$d^{t+1} = e^{\left(\frac{t-t_{max}}{t_{max}}\right)} - \left(\frac{t}{t_{max}}\right) \tag{2.34}$$

here, d decreases with time, allowing it to converge in a previously discovered favorable region.

The ability of the metaheuristic optimization algorithm to "explore" and "exploit" is the most critical aspect impacting its performance. Exploration refers to a search algorithm's capacity to search multiple sections of the search space to locate a suitable optimal with a high probability. On the other hand, exploitation refers to the degree to focus a search on a promising location to fine-tune a candidate solution. When the population diverges, the algorithm is in an exploration stage; when the population condenses into a limited search region, the algorithm is in an exploitation state. A metaheuristic algorithm generally begins the search process with more exploration and minor exploitation, but as the search advances toward the finish, the ratios invert.

Case I - Exploration phase
If $TF \leq 0.5$, there is collision between objects and in this case acceleration for iteration $t+1$ is updated as

$$A^{t+1}(i) = \frac{D_{mr} + V_{mr} \times A_{mr}}{D^{t+1}(i) \times V^{t+1}(i)} \tag{2.35}$$

where D_{mr}, V_{mr}, and A_{mr} represents density, volume, and accelaration of random object.

Case II - Exploitation phase
If $TF > 0.5$, there is no collision between objects, update acceleration for iteration $t+1$ as

$$A^{t+1}(i) = \frac{D^{best} + V^{best} \times A^{best}}{D^{t+1}(i) \times V^{t+1}(i)} \tag{2.36}$$

where A^{best} is the acceleration of the best object.

Step 4 - Normalize acceleration
To calculate the percentage of change, acceleration is normalized as

$$A^{t+1}(\bar{i}) = u \times \frac{A^{t+1}(i) - min(A)}{max(A) - min(A)} + l \tag{2.37}$$

where u and l are the range of normalization and set to 0.9 and 0.1, respectively.

If object i is distant from the global optimum, its acceleration value is high, indicating that it is in the exploration phase; otherwise, it is in the exploitation phase. This assists search agents in moving towards the global best solution while also moving away from local ones. Hence, AOA achieves the balance between exploration and exploitation.

Step 5 - Update position

The updated density, volume, and acceleration determine the new position of an object according to the following cases.

The pseudo-code for iterative AOA is as follows:

Algorithm 2 Pseudo code for AOA

Input: Fix the population size N, maximum iterations t_{max}, c_1, c_2, c_3, and c_4;
Initialization: Population with random positions, densities and volumes, Evaluate initial population and select the one with the best fitness value;
while $t < t_{max}$ or convergence criterion **do**
    for each search agent, update density and volume using 2.32 **do**
        update transfer and density factors TF and d using 2.33 and 2.34;
        if $TF \leq 0.5$ **then**
            update accelaration using 2.35 and normalize accelaration using 2.37
            update position using 2.38
        else
            update accelaration using 2.36 and normalize using 2.37
            update direction flag F using 2.40
            update position using 2.39
        end if
    end for
    Calculate fitness of all search agents and select the one with the best fitness value.
end while
Output: return population with the best fitness value.

Case I: Exploration phase

$TF \leq 0.5$

$$x^{t+1}(i) = x^t(i) + c_1 \times rand \times A^{t+1}(\bar{i}) \times d \times (x^{rand} - x^t(i)) \tag{2.38}$$

where, c_1 is constant equal to 2.

Case II: Exploitation phase

$TF > 0.5$

$$x^{t+1}(i) = x^{best^t}(i) + F \times c_2 \times rand \times A^{t+1}(\bar{i}) \times d \times (T \times x^{best} - x^t(i)) \tag{2.39}$$

where, c_2 is constant equal to 6.

T is defined as $T = c_3 \times TF$, where c_3 is a constant. T increases in the range $[c_3 \times 0.3, 1]$ over time and initially subtracts a percentage from the best position.

F is the flag to change the direction of motion using

$$F = \begin{cases} -1; & \text{if } P > 0.5 \\ 1; & \text{if } P \leq 0.5 \end{cases} \tag{2.40}$$

where $P = 2 \times rand - c_4$, where c_4 is a constant

Step 6 - Evaluation

Evaluate each object using the objective function f and remember the best solution found so far.

Assign x^{best}, D^{best}, V^{best}, and A^{best}.

2.6 Numerical Insights

When the system size is large enough, performance measures evaluated using traditional methods are not efficient and cost-effective due to the complexity of the problem and the increase in its dimension. In such a case, we use the state-of-the-art method and optimization technique to solve the cost minimization problem computationally. Although an increased system size makes the problem more complicated, it helps analyze the developed model with existing real-world systems and a more realistic model. In this section, we focus on numerical and optimal insights. Specifically, we consider the impact of different parameters on system performance measures, which helps in deciding which parameter substantially impacts some performance measures whereas less impact on others.

The numerical results for different experiments conducted on MATLAB (R2020b, 64-bit, License number 925317) on computing system with configuration Intel(R) Xeon(R) CPU E3-1231 v3 @ 3.40GHz 3.40 GHz with RAM 32.0 GB for various governing parameters and costs are summarized in Figures 2.1-2.10 and Table 2.1-2.6. For figures 2.1-2.10, the default parameters are fixed as follows $K = 15$, $\lambda = 2$, $\mu_1 = 1$, $\mu_2 = 1.5$, $\xi = 0.9$, $\nu = 1$,

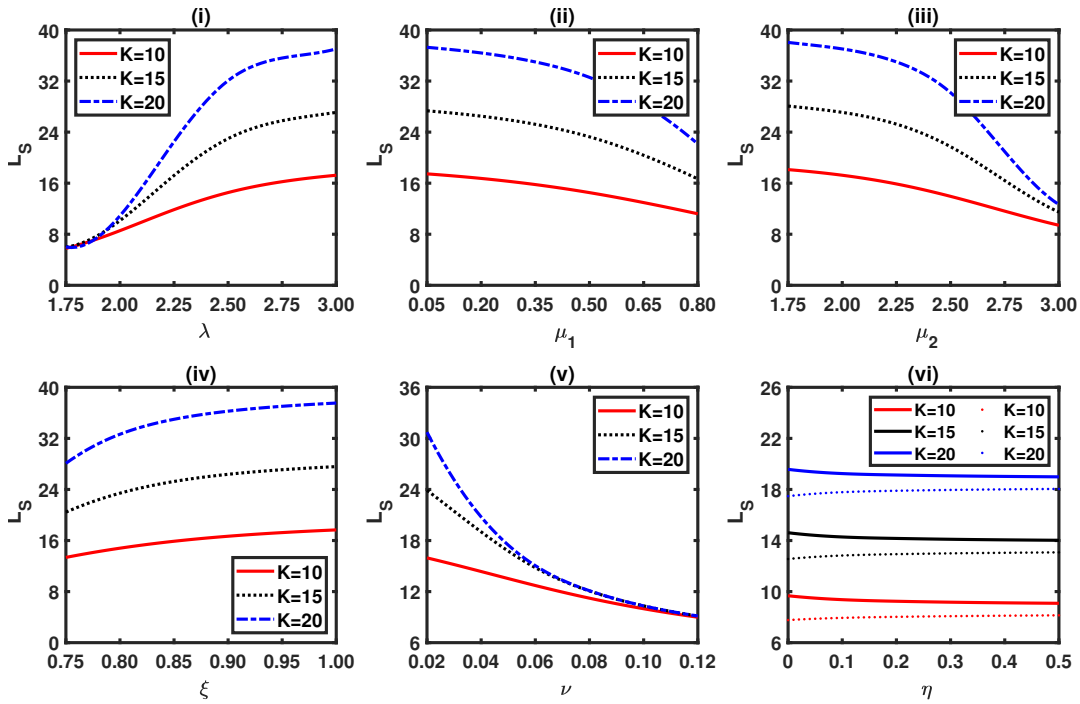


Figure 2.1: Expected number of customers in the system (L_S) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

and $\eta = 0.9$ and examine the effect of parameters K, λ, μ_1, μ_2, ξ, ν, and η on system performances as the values of one or two of these parameters vary given that others are fixed as above.

Figure 2.1 depicts the trend of the expected number of customers in the system L_S, which varies as a function of the governing parameters for capacity K. The value of L_S increases in proportion to the system capacity K as more customers are accommodated. The graph shows an increasing trend in the value of L_S for λ and ξ, and a decreasing trend for μ_1, μ_2, and ν as expected. Figure 2.1(vi) shows the value of customers in front of server 1 and server 2 for various system capacities and their variation concerning jockeying rate. The optimal service rates need to be set to minimize the customers' balking and reneging and diminish the queue length.

Figure 2.2 shows a bar graph of customers' waiting time in the system W_S as a function of the governing parameters for different values of K. The value of W_S increases proportionately to K since it allows more customers to accommodate. The graph shows similiar trend in the value of W_S for λ and ξ, μ_1, μ_2, and ν as L_S in Figure 2.1, whereas W_S shows no change to jockeying rate η. The appropriate service facilities need to be designed in view of the incurred cost to minimize the customers' balking and reneging and lessen the waiting time.

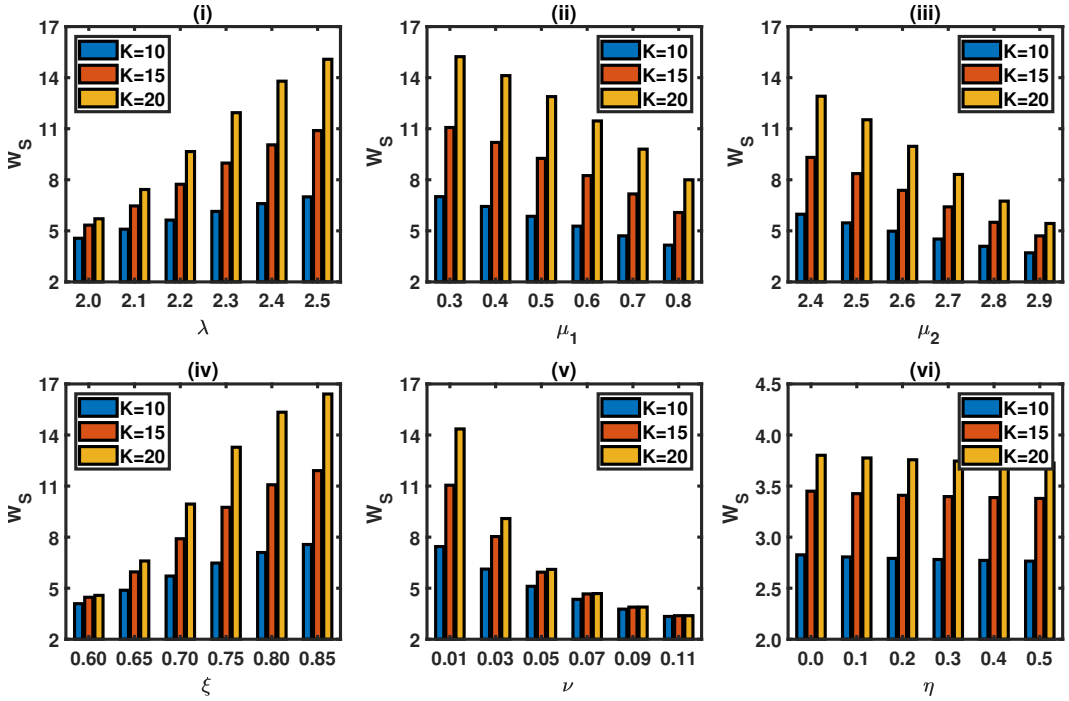


Figure 2.2: Expected waiting time of customers in the system (W_S) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

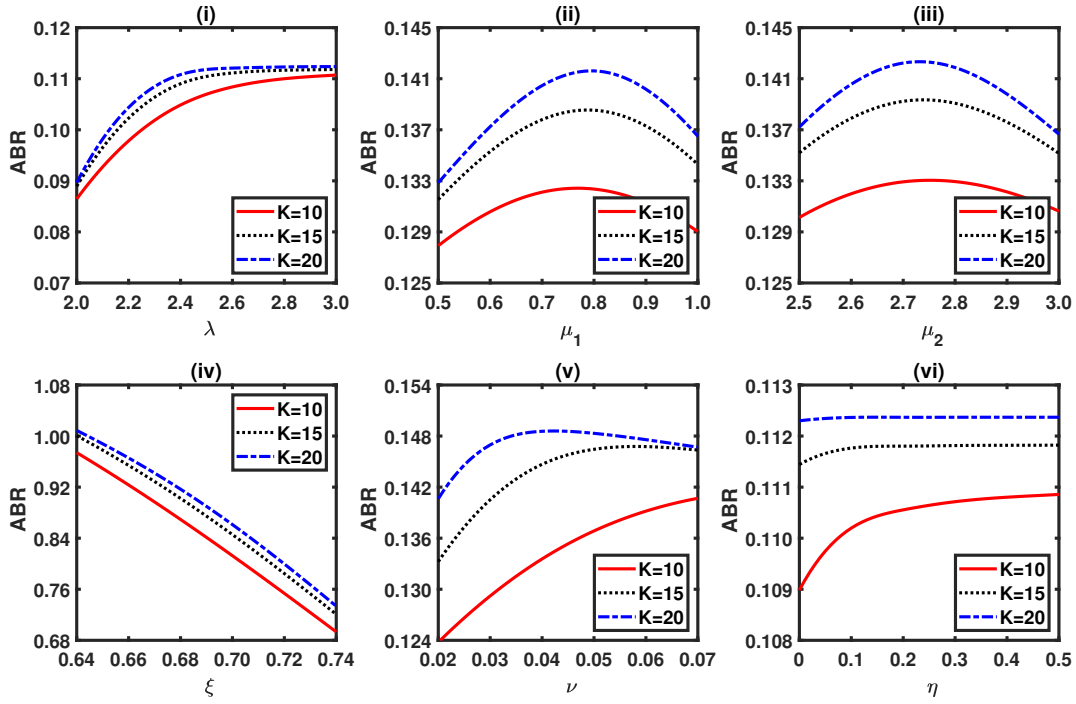


Figure 2.3: Average balking rate of customers in the system (ABR) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

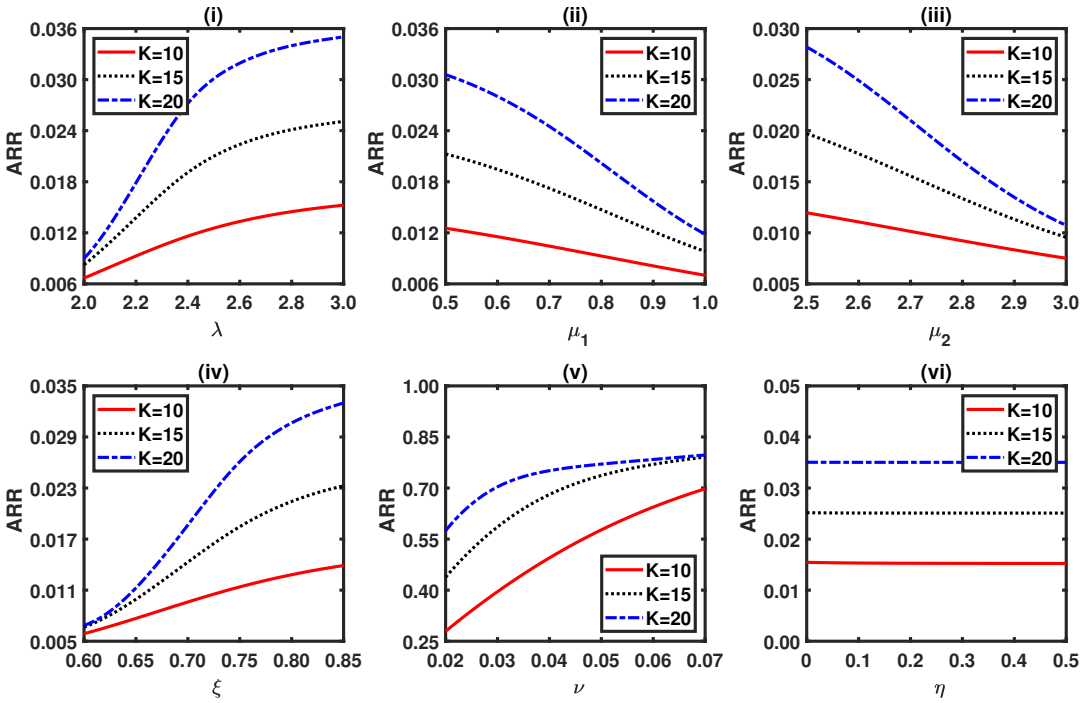


Figure 2.4: Average reneging rate of customers in the system (ARR) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

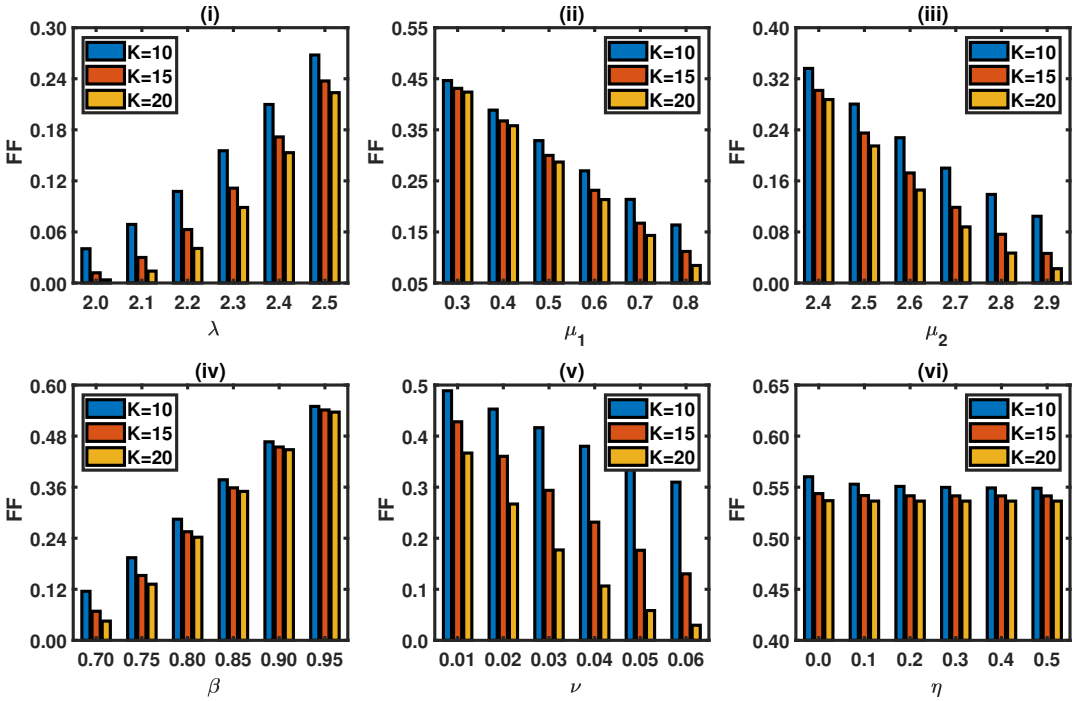


Figure 2.5: Failure frequency of the system (FF) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

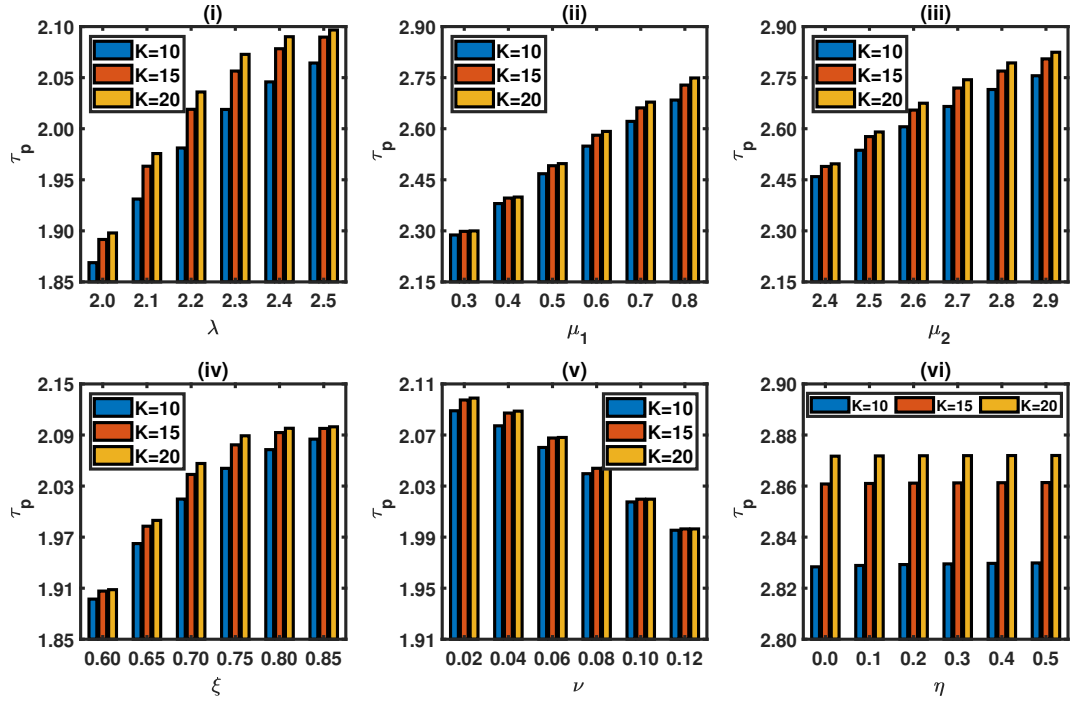


Figure 2.6: Throughput of the system (τ_p) wrt (i) λ, (ii) μ_1, (iii) μ_2, (iv) ξ, (v) ν, and (vi) η for different values of K.

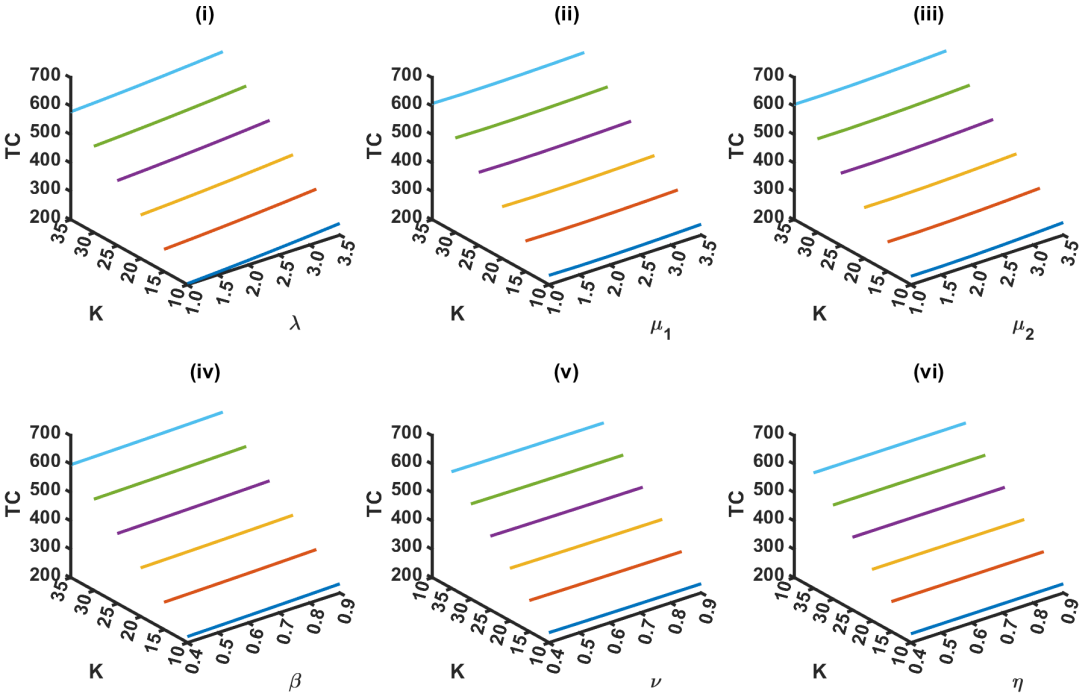


Figure 2.7: Discrete graphs for the expected total cost of the system (TC) for different parameters

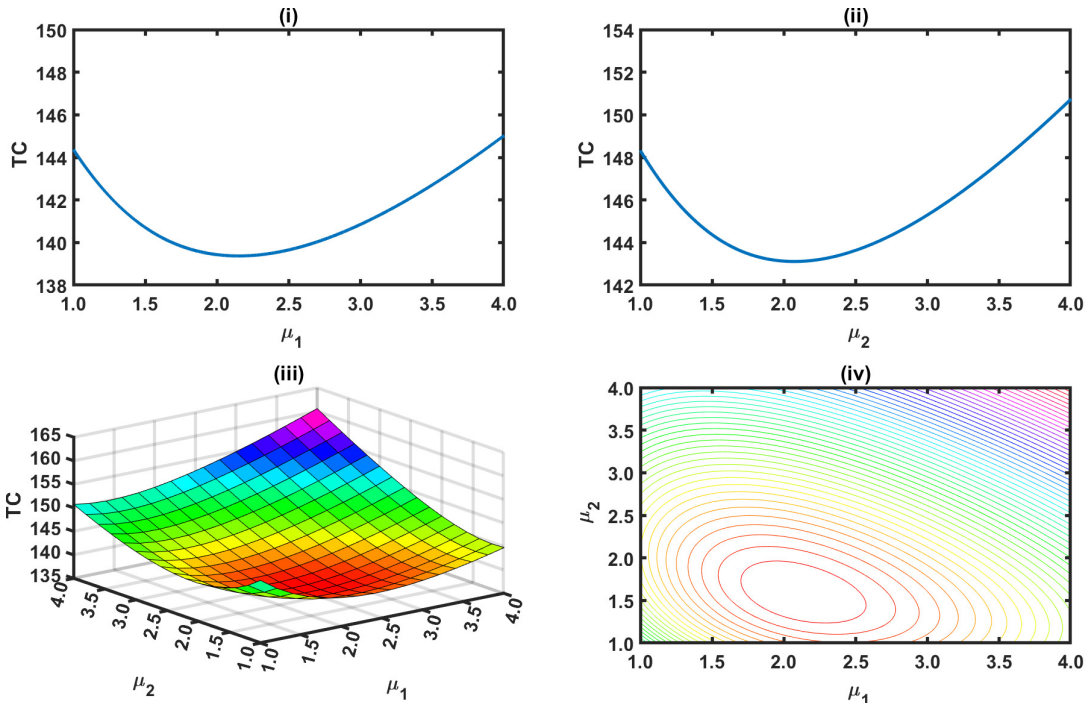


Figure 2.8: Expected total cost of the system (TC) for different parameters

The variation in the average balking rate *ABR* of a customer in relation to system parameters is depicted in Figure 2.3, which shows the value of *ABR* increases in proportion to K in all circumstances since there are more allowable customers. As the arrivals increase, there is more chance of customers balking away from the system, so *ABR* increases initially for λ, but later on, the rate of increase in *ABR* slows down with an increase in λ. The rate of balking is inversely proportional to the joining rate. So, *ABR* shows a slant decrease as ξ increases. When customers waiting in queues for service show impatience and try to leave the system, it will negatively impact arriving customers. The balking rate increases with an increase in reneging. When the system capacity is higher, the rate of increase in *ABR* decreases apparently. In case of an increase in the jockeying rate, *ABR* increases to a specific limit, become steady and does not change at all with an increase in η. It seemingly shows impatience attributes among customers' behavior and recommends prompt preventive measures.

Figure 2.4 depicts the change in a customer's average reneging rate *ARR* about system parameters, showing that the value of *ARR* grows in proportion to K since there are more customers in a high capacity service system. As the number of arrivals rises, the likelihood of customers abandoning the system increases; hence *ARR* rises with λ. When service is improved, the tendency to leave the system decreases the rate of reneging. So in subfigures (ii) and (iii) in Figure 2.4, the average rate of reneging decreases when we increase service

rates μ_1 and μ_2 respectively. Whereas an increase in the joining probability of customers leads to more congestion in the system, there will be more reneging of customers due to long waits. Therefore, *ARR* increases as ξ increases. *ARR* increases with an increasing rate of reneging ν. On the other hand, jockeying will not impact the average reneging rate.

The deviation in the failure frequency FF to system parameters is presented in Figure 2.5 for different system capacity K. Similar results for FF are perceived as for W_S above for all parameters except the fact that failure frequency is higher for smaller values of K. The apparent effects of FF also support the correct mathematical modeling and theoretical results. The bar graph in Figure 2.6 represents the variation in the system's throughput (τ_p) to system parameters for different system capacities K. As the number of customers increases due to an increase in λ and ξ, the system's throughput increases; since service is provided to more people by these servers. The throughput of the system also increases with service rate μ_1 and μ_2. The jockeying factor η does not impact the system's throughput as the customer only switches the queue and does not depart from the system. As the reneging rate ν increases, the customer leaves the system without getting service, and throughput decreases.

Besides the above-considered default value of system parameters, for figuring the change in expected total cost (TC) formulated in Eqn. 2.31, we set different unit costs value as follows $C_h = 25$, $C_b = 5$, $C_r = 15$, $C_k = 5$, $C_1 = 8$, $C_2 = 10$. We plot the variation of TC for varied rates and thresholds in Fig. 2.7. The palpable trends are noticed, which is evident in our expected total cost formulation and modeling to be correct. The illustrated results prompt an exploration of the optimal strategies for an efficient service system at minimum incurred costs.

In Figure 2.8, we provide line graphs, surface, and contour plots with respect to decision parameters μ_1 and μ_2 for default values of threshold rates and costs as assumed above. Furthermore, in order to validate the proposed model, we obtained all graphs of the expected total cost in a convex shape concerning these decision parameters. Figure 2.9 shows some selective images for the convergence of Archimedes optimization algorithm. The selection of these eight out of 30 images shows how these solutions are spread at random initially and later on converge to an optimal point in the last.

In Fig. 2.10, we display the plot of optimal TC^* for different iterations for multiple runs and observe the convergence to the same value for all runs. It supports our choice of Archimedes optimization algorithm for optimal analysis. Analytically, it is impossible to establish since TC is the function of system performances which are the expression of state probabilities that we get on solving the governing Chapman-Kolmogorov differential-difference equations. To obtain the optimal value of decision parameters μ_1 and μ_2, we

Figure 2.9: Convergence of iteration of Archimedes optimization algorithm

Table 2.1: Iteration of quasi-Newton method with initial guess $\mu_1 = 1, \mu_2 = 1.5$

Number of iteration	μ_1	μ_2	TC	$\frac{\partial TC}{\partial \mu_1}$	$\frac{\partial TC}{\partial \mu_2}$	**max**$\{\mid \frac{\partial TC}{\partial \mu_1} \mid, \mid \frac{\partial TC}{\partial \mu_2} \mid\}$
0	1.000000000	1.500000000	144.3688190	10.582518556	4.796131904	10.582518556
1	1.648223102	1.636721310	139.9326195	2.809790892	0.863514014	2.809790892
2	2.024469156	1.594073835	139.3675161	0.435335524	0.111215687	0.435335524
3	2.108799128	1.579919255	139.3495095	0.015566268	0.004379536	0.015566268
4	2.112010053	1.579444098	139.3494849	0.000022038	0.000006302	0.000022038
5	**2.112014604** (μ_1^*)	**1.579443440** (μ_2^*)	**139.3494849**	**0.000000001**	**0.000000003**	**0.000000003**

Table 2.2: Iteration of quasi-Newton method with initial guess $\mu_1 = 2, \mu_2 = 3$

Number of iteration	μ_1	μ_2	TC	$\frac{\partial TC}{\partial \mu_1}$	$\frac{\partial TC}{\partial \mu_2}$	**max**$\{\mid \frac{\partial TC}{\partial \mu_1} \mid, \mid \frac{\partial TC}{\partial \mu_2} \mid\}$
0	2.000000000	3.000000000	144.1628692	1.839879901	5.788102973	5.788102973
1	2.271638964	0.277295295	149.1942472	3.751134364	20.113644145	20.113644145
2	2.373778048	0.948609854	140.9075420	0.443875254	5.720689838	5.720689838
3	2.168849917	1.408897252	139.4477538	0.141912901	1.232062398	1.232062398
4	2.115511993	1.565957019	139.3500849	0.015139104	0.092484090	0.092484090
5	2.112035910	1.579354566	139.3494849	0.000108108	0.000609752	0.000609752
6	**2.112014607**(μ_1^*)	**1.579443431**(μ_2^*)	**139.3494849**	**0.000000007**	**0.000000055**	**0.000000055**

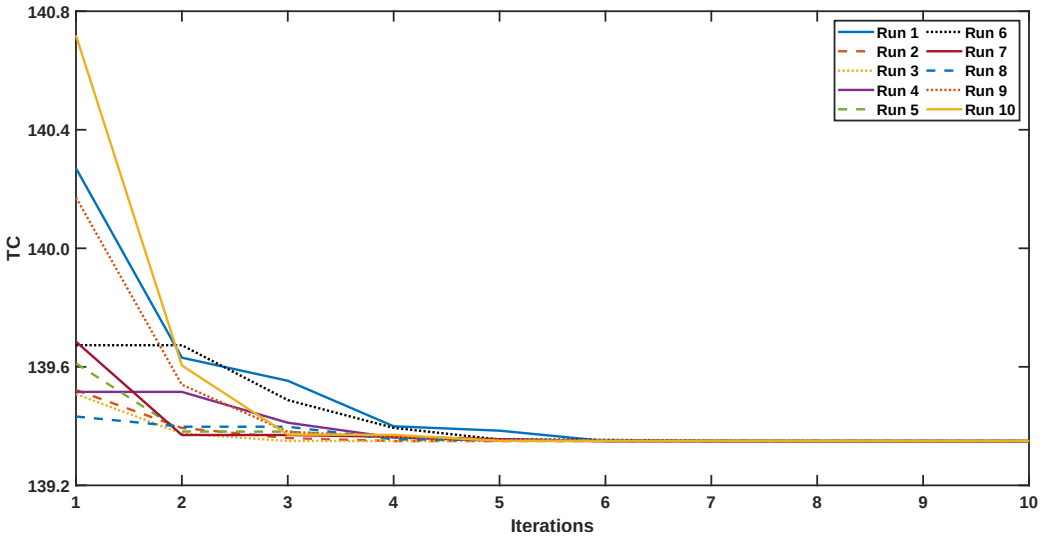


Figure 2.10: Convergence of iteration of Archimedes optimization algorithm

employ the quasi and metaheuristic optimization techniques and show that metaheuristic is very useful for the optimal analysis of complex real-time systems.

In Tables 2.1 and 2.2, as would be expected, we observe that whatever be the initial value of decision parameters μ_1 and μ_2 with tolerance 10^{-9}, the optimal value of μ_1^* and μ_2^* is achieved in a finite number of iterations through the quasi-Newton method. We compile the expected total cost corresponding to each iteration and gradient of TC with respect to μ_1 and μ_2. The last row gives the optimal value of μ_1, μ_2 and TC say μ_1^*, μ_2^* , and $TC(\mu_1^*, \mu_2^*)$ where $\max\left[\frac{\partial TC}{\partial \mu_1}, \frac{\partial TC}{\partial \mu_2}\right] < 10^{-9}$. Table 2.3 lists the four variations each of system parameters K, λ, ξ, ν, and η to find the optimal value of decision parameters μ_1, μ_2, and expected total cost as μ_1^*, μ_2^* and $TC(\mu_1^*, \mu_2^*)$ obtained via quasi-Newton method. Each of the variations gives the optimal value of total cost in 5 iterations as shown in the Table 2.3 for the initial value of decision parameters $\mu_{1_0} = 1.5$ and $\mu_{2_0} = 2$. The value of expected total cost increases with an increase in K, λ, and ξ whereas decreases as ν increases. The total cost seems insensitive to change in η and shows a slight decrease as η increases significantly. In the similiar manner,the Table 2.4 present four variations each of C_h, C_b, C_r, C_k, C_1, and C_2 to obtain the optimal values μ_1^*, μ_2^*, and $TC(\mu_1^*, \mu_2^*)$ and number of iterations required to reach these optimal values via quasi-Newton method. It can be inferred from table that when cost elements increases, the expected total cost increases as well. For the same sets of system parameters and unit costs considered in Table 2.3 and 2.4, the optimal analysis results through Archimedes optimization algorithm (AOA) have been compiled in Table 2.5 and 2.6, respectively. For the implementation of AOA, we have set the number of populations, number of iterations, and number of runs as 50, 30, and 10, respectively.

Table 2.3: Optimal expected total cost of the system $TC(\mu_1^*, \mu_2^*)$ for different parameters via Newton quasi method with $\mu_1 = 1.5, \mu_2 = 2$

K,λ,ξ,ν, η	**Number of iteration**	μ_1^*	μ_2^*	$TC(\mu_1^*, \mu_2^*)$
10,2,0.9,1,0.9	5	2.112014604	1.579443440	114.3494849
15,2,0.9,1,0.9	5	2.112014607	1.579443431	139.3494849
20,2,0.9,1,0.9	5	2.112014603	1.579443440	164.3494849
25,3,0.9,1,0.9	5	2.112014604	1.579443440	189.3494849
15,2,0.9,1,0.9	5	2.112014607	1.579443431	139.3494849
15,3,0.9,1,0.9	5	2.841282540	2.045699067	157.2286987
15,4,0.9,1,0.9	5	3.550213842	2.485950990	173.5772999
15,5,0.9,1,0.9	5	4.247474322	2.908748448	188.9565724
15,2,0.6,1,0.9	5	1.822481957	1.369516149	135.5770317
15,2,0.7,1,0.9	5	1.921309780	1.441815007	136.8568157
15,2,0.8,1,0.9	5	2.017641025	1.511649844	138.1128190
15,2,0.9,1,0.9	5	2.112014607	1.579443431	139.3494849
15,2,0.9,1,0.9	5	2.112014607	1.579443431	139.3494849
15,2,0.9,2,0.9	5	1.923198586	1.435851424	137.7867282
15,2,0.9,3,0.9	5	1.815045016	1.356754189	136.8526057
15,2,0.9,4,0.9	5	1.746419175	1.307841817	136.2349810
15,2,0.9,1,0.9	5	2.112014607	1.579443431	139.3494849
15,2,0.9,1,1.9	5	2.120662143	1.564358120	139.2005875
15,2,0.9,1,2.9	5	2.126676724	1.554264397	139.0935507
15,2,0.9,1,3.9	5	2.131036169	1.547112659	139.0127958

Table 2.4: Optimal expected total cost of the system $TC(\mu_1^*, \mu_2^*)$ for different parameters via Newton quasi method with initial guess $\mu_1 = 1.5, \mu_2 = 2$.

C_h,C_b,C_r,C_k,C_1,C_2	**Number of iteration**	μ_1^*	μ_2^*	$TC(\mu_1^*, \mu_2^*)$
15,5,15,5,8,10	5	1.726334744	1.207946750	126.688202
20,5,15,5,8,10	5	1.929439481	1.405104536	133.351003
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
30,5,15,5,8,10	5	2.278794065	1.736943788	144.841569
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
25,10,15,5,8,10	5	2.124792874	1.589587007	139.635631
25,15,15,5,8,10	5	2.137484225	1.599657312	139.919548
25,20,15,5,8,10	5	2.150089587	1.609655278	140.201273
25,5,10,5,8,10	5	2.087119948	1.558572231	139.290369
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
25,5,20,5,8,10	5	2.208228453	1.656827581	141.061106
25,5,25,5,8,10	5	2.262345088	1.700748835	141.876791
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
25,5,15,10,8,10	5	2.150089587	1.609655278	215.201274
25,5,15,15,8,10	5	2.150089588	1.609655278	290.201273
25,5,15,20,8,10	5	2.150089588	1.609655278	365.201273
25,5,15,5,6,10	6	2.713618006	1.446345318	135.375019
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
25,5,15,5,10,10	5	1.754711893	1.754711893	144.085651
25,5,15,5,12,10	4	1.456775966	1.884804721	147.284518
25,5,15,5,8,5	5	1.694899039	2.903228662	128.441137
25,5,15,5,8,10	5	2.112014607	1.579443431	139.349484
25,5,15,5,8,15	5	2.483070851	1.010359486	145.606722
25,5,15,5,8,20	5	2.737327350	0.645683093	149.683941

Table 2.5: Optimal expected total cost of the system ($TC(\mu_1^*, \mu_2^*)$) for different parameters via Archimedes optimization algorithm.

K,λ,ξ,ν,η	μ_1^*	μ_2^*	$\mathbf{TC}(\mu_1^*, \mu_2^*)$	**mean** $\frac{TC_i}{TC^*}$	**max** $\frac{TC_i}{TC^*}$	**time elapsed**
10,2,0.9,1,0.9	2.1120436227	1.5794837401	114.3494848481	1.00000000001049	1.00000000005159	675.7083195
15,2,0.9,1,0.9	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
20,2,0.9,1,0.9	2.1120465003	1.5794833982	164.3494848481	1.00000000000182	1.00000000000791	848.7520072
15,2,0.9,1,0.9	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
15,3,0.9,1,0.9	1.9213427660	1.4418557244	136.8568157391	1.00000000833455	1.00000004162598	663.5704982
15,4,0.9,1,0.9	1.9213427176	1.4418560621	136.8568157391	1.00000000244664	1.00000000879824	838.8943486
15,2,0.7,1,0.9	3.5502403832	2.4859896468	173.5772999240	1.00000000000034	1.00000000000115	855.6600125
15,2,0.8,1,0.9	2.0176746196	1.5116895493	138.1128190122	1.00000000000839	1.00000000004127	603.3146233
15,2,0.9,1,0.9	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
15,2,0.9,1,0.9	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
15,2,0.9,2,0.9	2.0176730523	1.5116882021	138.1128190122	1.00000000000304	1.00000000000796	804.7633502
15,2,0.9,3,0.9	1.8150809458	1.3567958916	136.8526056585	1.00000000000134	1.00000000000146	819.7758594
15,2,0.9,1,0.9	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
15,2,0.9,1,1.9	2.1267062995	1.5543036234	139.0935507000	1.00000000001020	1.00000000002516	678.5918819
15,2,0.9,1,2.9	2.1267064779	1.5543039162	139.0935507000	1.00000000024789	1.00000000123945	767.9797948

Table 2.6: Optimal expected total cost of the system ($TC(\mu_1^*, \mu_2^*)$) for different parameters via Archimedes optimization algorithm

C_h,C_b,C_r,C_k,C_1,C_2	μ_1^*	μ_2^*	$\mathbf{TC}(\mu_1^*, \mu_2^*)$	**mean** $\frac{TC_i}{TC^*}$	**max** $\frac{TC_i}{TC^*}$	**time elapsed**
20,5,15,5,8,10	1.9294723208	1.4051418112	133.3510028885	1.00000000001245	1.00000000006074	644.2674097
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
30,5,15,5,8,10	2.2788288165	1.7369832295	144.8415693530	1.00000000000003	1.00000000000029	901.1045426
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
25,10,15,5,8,10	2.1248215567	1.5896274422	139.6356319615	1.00000000005633	1.00000000027428	706.8352816
25,15,15,5,8,10	2.1375151372	1.5996971940	139.9195486867	1.00000000000014	1.00000000000071	814.6631040
25,5,10,5,8,10	2.0466991081	1.5264506708	138.4037638570	1.00000000002875	1.00000000014089	633.2690291
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
25,5,20,5,8,10	2.0467004622	1.5264506521	138.4037638570	1.00000000000028	1.00000000000072	746.5180005
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
25,5,15,10,8,10	2.1120478406	1.5794825424	214.3494848481	1.00000000003806	1.00000000018847	778.1173061
25,5,15,15,8,10	2.1120465003	1.5794833982	289.3494848481	1.00000000000103	1.00000000000449	854.1234979
25,5,15,5,6,10	2.6670214922	1.4192212522	134.6071886193	1.00000000000089	1.00000000000222	698.8064960
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
25,5,15,5,10,10	1.7220177365	1.7220173207	143.1634318589	1.00000000006118	1.00000000028359	628.0147493
25,5,15,5,8,10	2.1120438875	1.5794840444	139.3494848481	1.00000000000057	1.00000000000215	717.5180005
25,5,15,5,8,15	2.4391156213	0.9867825365	145.6225997350	1.00000000013322	1.00000000064550	640.6934541
25,5,15,5,8,20	2.6886243564	0.6252301692	149.5903847416	1.00000000009118	1.00000000004392	813.8234188

We consider the range [0.1, 4.1] for both μ_1 and μ_2. Table 2.5 and 2.6 summarizes the results in terms of μ_1^*, μ_2^*, and $TC(\mu_1^*, \mu_2^*)$ and be verified with results in Table 2.3 and 2.4. For almost all the sets, we have similar results. It evidences that a metaheuristic method AOA is suitable for such complex real-time problems. For the statistical validation of AOA convergent results, we compute the mean and maximum of $\frac{\min TC_i}{TC_i}$. The mean $\left[\frac{\min TC_i}{TC_i}\right]$ ranges from 1.00000000000003 to 1.00000000833455 whereas max $\left[\frac{\min TC_i}{TC_i}\right]$ ranges from 1.00000000000029 to 1.00000004162598. It shows how AoA is close to the optimal solution for multiple runs.

2.7 Conclusion

In this chapter, we have investigated a finite capacity multi-server queueing system with the impatient behavior of customers, such as balking, jockeying, and reneging, taken into account simultaneously. We have employed the matrix-analytic method to determine the steady-state probabilities and computed various system performance measures. The numerical simulation of various system performance measures has been accomplished to study the system parameters' effects. We also formulated a cost function and defined the problem of cost minimization constraint. The optimal expected cost is the state-of-the-art analysis of customers' impatience attributes in the service system. We use an efficient meta-heuristic optimization algorithm: AOA and quasi-Newton method with the aid of MATLAB software to analyze the optimal values of decision parameters μ_1 and μ_2 with the optimal stability condition and a global minimum of the cost function. Finally, several numerical experiments have been included to demonstrate and attain optimal results. The cost analysis clearly communicates the validity and profitability of the established model. Minimizing service cost, a widely sought attribute of any firm will benefit system designers and decision-makers.

Chapter 3

Economic Analysis of a Service System with Unreliable Service of Two Types of Servers

This chapter on service systems considers two types of servers: multi-subordinate servers working in parallel and one chief server. The prospective customers arrive in the system according to the Poisson process and join the single queue for initial service from any of the subordinate servers, where they provide service according to an exponentially distributed service time. The subordinate server behaves as a customer on behalf of the prospective customer for the final service in tandem with the chief server, which provides service following an exponential distribution.

3.1 Introduction

Queueing theory has seen numerous advancements and new applications in the last few decades. The operation of service, manufacturing, and supply chain management requires a thorough grasp of queueing systems. The rich and fertile theory is now used to analyze communication networks and computer systems for internet and data traffic or bandwidth management, health care systems, traffic control, data science, machining systems, and many others where "standing, waiting, and serving" takes precedence. The congestion leads to blocking and delay, which leads to loss of customer time, goodwill, and satisfaction and increases service costs. A queueing system mainly has three components, the input process, the service mechanism, and the queue discipline.

Customers and servers who serve them make up conventional queueing systems, which have been widely researched in the literature. However, this may only be the case sometimes. Servers may play the role of customers in some service systems and conversely. Perel and Yechiali [151] first initiated this type of service system. They studied a service system consisting of two connected queues in which customers of one queue act as servers of the other queue while waiting in that queue. Several applications of this model related to real-life systems were presented. To extend the applicability of the model, they further expanded the scope of the analysis to the case where the customers of both queues act as servers, and customers of each queue are the server of the other queue [152]. Sendfeld [165] investigated a broader expansion of the queueing model proposed by Perel & Yechiali [151] and append an overflow capability. The arrival and service rates for the finite first queue in this generalized two-queue network depend on the state. The first queue's customers serve the second queue's customers; hence the second queue's service rate is determined by the first queue's state. Vidhya et al. [189] explored service systems consisting of two parallel server queues with infinite buffers. Here, the first queue goes through a setup procedure for a start-up process at the beginning of the queue. Some of the queue customers provide service to the customers in the second queue by joining hands with its server. A recent study by Hanukov [75] investigated a service system in which a subordinate server approaches the chief server on the customer's behalf. To investigate this model, the author conducted an economic impact assessment to find the optimal work division policy that ensures the amount of work devoted to each phase of the service to reduce the system's total cost or increase its functionality.

Scenarios in which servers act as customers and approach the chief server on the customer's behalf are quite natural in networks comprised of nodes that can receive and provide service simultaneously. The service system shall consist of two or more levels of authority

as service providers with uncertainties in their services' success are often seen in everyday life. An example related to a service network is web development firms. In these service firms, customers want to have a website for their business model but need help understanding how to develop it. They then approach subordinates of the firm as customers can't directly approach the chief developer of the company. They initially talk to a subordinate developer who performs the first phase of the service by listening to the requirements they want on their website to run their business model. After completing this phase of service, the subordinate developer approaches the chief developer on behalf of the customer. If the chief developer is busy with some other subordinate developer, he waits in the queue for his turn. The chief developer completes the service in the end with assistance from him. After service completion, the subordinate developer returns to his service phase, hands the website to the present customer, and prepares to serve the next customer in the first phase. The customer whose service is just completed inspects his website and tries to apply it to his business model. If he finds his service unreliable, there is still some technicality involved. He again approaches a subordinate developer and asks for the correction in the website by waiting in the queue. On the other hand, if he finds the website working correctly and fulfills his requirements, he leaves the system. Another application arises in the online purchasing of goods where customer care persons simultaneously act as both server and customer. The latter example demonstrates that subordinate servers are not required to approach the chief server physically. From the above examples, the chief server is a kind of senior authority or executive. As a result, the system only has one chief server, but the number of subordinate servers is considered finite or unrestricted.

In this study, we examine a service system in which service is rendered in phases by several subordinate servers in the initial stage and a single chief server in the final step. The literature on multi-phase queues is substantial. Krishna and Lee [109] first studied the issue of queueing systems with two-phase services using batch and individual phases. Doshi [43] extended Krishna and Lee's [109] work to incorporate general batch services and general individual service times. Choudhury and Deka [33] used a Bernoulli vacation schedule to conduct steady-state and reliability assessments on an $M/G/1$ queue with two-phase service and server breakdowns. Wang et al. [193] provided performance measures for a retrial queue with a finite number of sources and two-phase service, with service times considered to be generally distributed in both phases. A number of studies (see, e.g., [127], [35], [34], [210], [22], [175], [204], [190], [116]) have recently appeared in the queueing literature in which concepts of service in phases have been discussed.

In the previous research, it was considered that a single server provides a two-phase service. Few articles have looked into scenarios including multi-server queues and a two-phase

service. Yang et al. [214] modeled a finite capacity $M/M/R/K$ queue with a two-phase service and a second optional channel. The matrix-geometric strategy was employed by Ke et al. [101] to compute the stationary probability distribution of the number of customers in a $M/M/R$ queue with a two-phase service. Ahuja et al. [7] investigated a multi-server retrial queue with a finite population, balking customers, and two-phase service.

At the epoch of all phases of service completion, customers pay attention to the service received for verification, whether reliable or unreliable. On inspection, if customers find service unreliable, he reattempts the service by joining the queue of subordinate servers. On the other hand, reliable service received customers depart from the system. In literature, studies on unreliable services are seldom available. Patterson and Korzeniowski [148] constructed the corresponding embedded Markov chain and computed the stationary queue length's probability generating function (PGF). They also provided the Laplace-Stieltjes transform of the stationary waiting time to provide sufficient conditions for positive recurrence and closed-form of stationary distribution for Markovian single server queue with a newly introduced constraint unreliable service. As an extension of [148], Patterson and Korzeniowski ([150],[149]) derived an explicit closed-form of the stationary distribution of a Poisson arrival, exponential service, and single server queue with unreliable service and working vacation. Shekhar et al. [171] conducted sensitivity and optimal analysis for the expected total cost incurred and reliability characteristics for a machine repair problem (MRP) of standbys provisioning in a Markovian environment with unreliable service and vacation interruption. Recently, Esfeh [56] formulated new analytical estimates of mean waiting time for diverse transit systems, including dial-a-ride, feeder-trunk, and single route with unreliable service.

To the best of our knowledge, there has been no study on service systems with a subordinate-chief server approach and unreliable service. In this sense, this study fills the gap and helps economically analyze such a service system and find the decisive parameters.

The rest of this chapter is organized as follows: In the next Section 3.2, the model is described with system states, notations, and steady-state equations. These equations are solved to find probabilities using the repeated substitution approach in Section 3.3. The system's performance measures are derived and computed in vector form in Section 3.4. The cost function is constructed for the model in Section 3.5. The meta-heuristic technique TLBO used in the presented model is explained in detail along with its pseudo-code in Section 3.6. We compute the steady-state probabilities numerically and plot the graphs for various performance measures to check their sensitiveness to system parameters 3.7. Sensitivity analysis is also conducted to obtain the optimal solution in Section 3.8. Finally, concluding remarks are discussed in Section 3.9.

3.2 Problem Description and Formulation

In this study, we formulated a queueing model in which the service providers are of two types: R subordinate servers and one chief server. The service is completed commutatively by subordinates and chief servers in two consecutive stages. The initial stage service is performed by one of the subordinate servers, and the chief server completes the final stage service. Any arriving customer initially joins one of the queues of subordinate servers who are arranged in parallel to serve first phase service. After completing this phase of service, the subordinate server approaches the chief server on behalf of the customer and acts as a customer in this phase. If the chief server is busy serving another subordinate server, he waits for his turn in the queue and remains occupied and doesn't perform service in this period. The chief server completes the service with the help of the subordinate server. Subordinate servers are released after the final phase service to serve the next customer waiting for the initial phase service. Upon service completion in both phases, the customer inspects whether the service is successful or not on their behalf. Customers leave the system only after successfully completing their service; otherwise, they rejoin the system and wait for service in one of the queue of subordinate servers.

Assumptions and Notations:

- The arrival of customers in the system follows a Poisson process with a rate λ.
- The inter-service time for subordinate and chief servers is exponentially distributed with rates μ and α, respectively.
- When all our subordinate services are busy, the upcoming customers have to wait in a queue and will be provided service on a FCFS basis.
- When the chief server is busy serving a subordinate server, the arriving subordinate servers will form a queue and be served on a FCFS basis.
- It is assumed that the service provided by the subordinate-chief server may be unreliable, which means that the service may be unsuccessful many times before it is successful. The rate of successful and unsuccessful services are β_1 and β_2, respectively.
- All stochastic processes and events are repeated repeatedly and statistically independent of each other.

3.2.1 Steady State Equations

The presented model is a QBD process in two dimensions with the state-space $\{N_1, N_2, J\}$, where N_1 represents the count of customers in the system, N_2 represents the count of subordinate servers acting as customers for the chief server, and J denotes the inspection state of the server after providing service to check whether service is successful or unsuccessful in a steady state. The system states' joint probability distribution function is as follows:

$P_{n_1,n_2} = \text{Prob}\{N_1 = n_1, N_2 = n_2, J = 0\};\ n_1 = 0,1,2,\ldots,K\ \&\ n_2 = 0,1,2,\ldots,R \text{ with } n_1 \geq n_2$

$Q_{n_1,n_2} = \text{Prob}\{N_1 = n_1, N_2 = n_2, J = 1\};\ n_1 = 0,1,2,\ldots,K\ \&\ n_2 = 0,1,2,\ldots,R \text{ with } n_1 \geq n_2$

The system states are arranged in the following lexicographic order:
$\{P_{0,0}, Q_{1,1}, P_{1,0}, P_{1,1}, Q_{2,1}, Q_{2,2}, ..., P_{R-1,0}, P_{R-1,1}, P_{R-1,R-1}, Q_{R,1}, Q_{R,2}, ..., Q_{R,R,}, P_{R,0}, P_{R,1}, ...$
$P_{R,R}, ..., Q_{K,1}, Q_{K,2}, ..., Q_{K,R}, P_{K,0}, P_{K,1}, ..., P_{K,R}\}$.
We have the following Chapman-Kolmogorov forward equations using the birth and death process and relating the system's states to a steady state.

$$-\lambda P_{0,0} + \beta_1 Q_{1,1} = 0 \tag{3.1}$$

$$-(\lambda + n_1\mu)P_{n_1,0} + \lambda P_{n_1-1,0} + \beta_2 Q_{n_1,1} + \beta_1 Q_{n_1+1,1} = 0;\ 1 \leq n_1 \leq R-1 \tag{3.2}$$

$$-(\lambda + R\mu)P_{n_1,0} + \lambda P_{n_1-1,0} + \beta_2 Q_{n_1,1} + \beta_1 Q_{n_1+1,1} = 0;\ R \leq n_1 \leq K-1 \tag{3.3}$$

$$-(R\mu)P_{K,0} + \lambda P_{K-1,0} + \beta_2 Q_{K,1} = 0 \tag{3.4}$$

$$-(\lambda + \alpha)P_{n_1,n_1} + \mu P_{n_1,n_1-1} + \beta_1 Q_{n_1+1,n_1+1} = 0;\ 1 \leq n_1 \leq R-1 \tag{3.5}$$

$$-(\lambda + \alpha)P_{R,R} + \mu P_{R,R-1} = 0 \tag{3.6}$$

$$-(\lambda + \alpha)P_{n_1,R} + \lambda P_{n_1-1,R} + \mu P_{n_1,R-1} = 0;\ R+1 \leq n_1 \leq K-1 \tag{3.7}$$

$$-\alpha P_{K,R} + \mu P_{K,R-1} + \lambda P_{K-1,R} = 0 \tag{3.8}$$

$$-(\lambda + (n_1 - n_2)\mu + \alpha)P_{n_1,n_2} + (n_1 - n_2 + 1)\mu P_{n_1,n_2-1} + \lambda P_{n_1-1,n_2} + \beta_2 Q_{n_1,n_2+1}$$
$$+\beta_1 Q_{n_1+1,n_2+1} = 0;\ 2 \leq n_1 \leq R-1,\ 1 \leq n_2 \leq n_1 - 1 \tag{3.9}$$

$$-(\lambda + (R - n_2)\mu + \alpha)P_{n_1,n_2} + (R - n_2 + 1)\mu P_{n_1,n_2-1} + \lambda P_{n_1-1,n_2}$$
$$+\beta_2 Q_{n_1,n_2+1} + \beta_1 Q_{n_1+1,n_2+1} = 0;\ R \leq n_1 \leq K-1,\ 1 \leq n_2 \leq R-1 \tag{3.10}$$

$$-((R - n_2)\mu + \alpha)P_{K,n_2} + (R - n_2 + 1)\mu P_{K,n_2-1} + \lambda P_{K-1,n_2}$$
$$+\beta_2 Q_{K,n_2+1} = 0;\ 1 \leq n_2 \leq R-1 \tag{3.11}$$

$$-(\beta_1 + \beta_2)Q_{n_1,n_2} + \alpha P_{n_1,n_2} = 0;\ 1 \leq n_1 \leq R,\ 1 \leq n_2 \leq n_1 \tag{3.12}$$

$$-(\beta_1 + \beta_2)Q_{n_1,n_2} + \alpha P_{n_1,n_2} = 0;\ R+1 \leq n_1 \leq K,\ 1 \leq n_2 \leq R \tag{3.13}$$

3.3 Steady-State Analysis

In order to demonstrate the steady-state probability distribution, the repeated substitution approach is used, as it is extremely difficult to calculate the closed-form expressions of the state probabilities due to multi-equation, multi-variable, and multiple-parameter queueing problems. The MAM was first familiarized by Neuts[144] utilizing the concept of embedded Markov chains in numerous realistic queue-based service systems. In this work, for applying MAM, we define the probability vectors $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{Q}}_i$ as follows:

$$\tilde{\mathbf{P}}_i = \begin{cases} [P_{i,0}, P_{i,1}, ..., P_{i,i}]; i = 0,1,2,...,R \\ [P_{i,0}, P_{i,1}, ..., P_{i,R}]; i = R+1, R+2, ..., K \end{cases} \tag{3.14}$$

$$\tilde{\mathbf{Q}}_i = \begin{cases} [Q_{i,1}, Q_{i,2}, ..., Q_{i,i}]; i = 1,2,3,...,R \\ [Q_{i,1}, Q_{i,2}, ..., Q_{i,R}]; i = R+1, R+2, ..., K \end{cases} \tag{3.15}$$

The complete probability vector of all system states is then calculated as

$$\mathbf{\Pi} = \left[\tilde{\mathbf{P}}_0, \tilde{\mathbf{Q}}_1, \tilde{\mathbf{P}}_1, \tilde{\mathbf{Q}}_2, \tilde{\mathbf{P}}_2, ..., \tilde{\mathbf{Q}}_K, \tilde{\mathbf{P}}_K\right] \tag{3.16}$$

which can be written as

$$\mathbf{\Pi} = [\mathbf{\Pi}_0, \mathbf{\Pi}_1, \mathbf{\Pi}_2, ..., \mathbf{\Pi}_K]$$

where, $\mathbf{\Pi}$ as a row vector whose $i^{th}(i = 1,2,...,K)$ element is the steady-state probability vector $[\tilde{\mathbf{Q}}_i, \tilde{\mathbf{P}}_i]$, i.e. $\mathbf{\Pi}_i = [\tilde{\mathbf{Q}}_i, \tilde{\mathbf{P}}_i]; i = 1,2,\ldots,K$ with $\mathbf{\Pi}_0 = \tilde{\mathbf{P}}_0$. The system's generator matrix, represented by $\mathbf{Q}$, can be constructed using the lexicographic order. Hence, the equivalent block-tridiagonal structure of the transition rate matrix $\mathbf{Q}$ of the continuous-time

Markov chain (CTMC) is represented as follows:

$$\mathbf{Q} = \begin{bmatrix}
\mathbf{A}_0 & \mathbf{0} & \mathbf{E}_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{B}_1 & \mathbf{C}_1 & \mathbf{F}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{D}_1 & \mathbf{A}_1 & \mathbf{0} & \mathbf{E}_1 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{B}_2 & \mathbf{C}_2 & \mathbf{F}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D}_2 & \mathbf{A}_2 & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{C}_{R-1} & \mathbf{F}_{R-1} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{D}_{R-1} & \mathbf{A}_{R-1} & \mathbf{0} & \mathbf{E}_{R-1} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{B}_R & \mathbf{C}_R & \mathbf{F}_R & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{D}_R & \mathbf{A}_R & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_R & \mathbf{0} & \mathbf{E}_R \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{B}_R & \mathbf{C}_R & \mathbf{F}_R \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{D}_R & \mathbf{A}_K
\end{bmatrix}$$

The Markov process' rate matrix $\mathbf{Q}$ is analogous to the quasi-birth and death process and the elements of the rate matrix $\mathbf{Q}$ as block submatrices are given as follows in element form as:

$$\mathbf{A}_n = \begin{bmatrix}
-\lambda - n\mu & n\mu & \dots & 0 & 0 \\
0 & -\lambda - (n-1)\mu - \alpha & \dots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \dots & -\lambda - \mu - \alpha & \mu \\
0 & 0 & \dots & 0 & -\lambda - \alpha
\end{bmatrix}_{(n+1)\times(n+1)}$$

where $0 \leq n \leq R$.

$$\mathbf{A}_k = \begin{bmatrix}
-R\mu & R\mu & 0 & \dots & 0 & 0 \\
0 & -(R-1)\mu - \alpha & (R-1)\mu & \dots & 0 & 0 \\
0 & 0 & -(R-2)\mu - \alpha & \dots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \dots & -\mu - \alpha & \mu \\
0 & 0 & 0 & \dots & 0 & -\alpha
\end{bmatrix}_{(R+1)\times(R+1)}$$

$$\mathbf{B}_n = \beta_1 \mathbf{I}_n; 1 \leq n \leq R-1$$

$$\mathbf{B}_R = \begin{bmatrix} \beta_1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \beta_1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \beta_1 & 0 & 0 \\ 0 & 0 & \dots & 0 & \beta_1 & 0 \end{bmatrix}_{R \times (R+1)}$$

$$\mathbf{C}_n = -(\beta_1 + \beta_2)\mathbf{I}_n; \ \mathbf{D}_n = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \alpha & 0 & \dots & 0 \\ 0 & \alpha & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha \end{bmatrix}_{(n+1) \times n} ; 1 \leq n \leq R$$

$$\mathbf{E}_n = \lambda \mathbf{I}_{n+1}; \ 0 \leq n \leq R$$

$$\mathbf{F}_n = \begin{bmatrix} \beta_2 & 0 & \dots & 0 & 0 \\ 0 & \beta_2 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \beta_2 & 0 \end{bmatrix}_{n \times (n+1)} ; 1 \leq n \leq R$$

Now, for the sake of computing steady-state probability vector $\mathbf{\Pi}$ from the system of equation 3.1-3.13, we have the following homogeneous system of equations with the help of rate matrix $\mathbf{Q}$ as

$$\mathbf{\Pi Q} = \mathbf{0} \tag{3.17}$$

with the initial condition $\mathbf{\Pi}_0 = P_{0,0} = 1$. The following steady-state matrix equations can represent the homogeneous governing system of the equation 3.1-3.13 in terms of pre-defined block matrices:

$$\begin{aligned}
&\tilde{\mathbf{P}}_0\mathbf{A}_0 + \tilde{\mathbf{Q}}_1\mathbf{B}_1 = \mathbf{0}\\
&\tilde{\mathbf{Q}}_n\mathbf{C}_n + \tilde{\mathbf{P}}_n\mathbf{D}_n = \mathbf{0};\ 1 \le n \le R\\
&\tilde{\mathbf{Q}}_n\mathbf{C}_R + \tilde{\mathbf{P}}_n\mathbf{D}_R = \mathbf{0};\ R+1 \le n \le K\\
&\tilde{\mathbf{P}}_n\mathbf{E}_n + \tilde{\mathbf{Q}}_{n+1}\mathbf{F}_{n+1} + \tilde{\mathbf{P}}_{n+1}\mathbf{A}_{n+1} + \tilde{\mathbf{Q}}_{n+2}\mathbf{B}_{n+2} = \mathbf{0};\ 0 \le n \le R-2\\
&\tilde{\mathbf{P}}_{R-1}\mathbf{E}_{R-1} + \tilde{\mathbf{Q}}_R\mathbf{F}_R + \tilde{\mathbf{P}}_R\mathbf{A}_R + \tilde{\mathbf{Q}}_{R+1}\mathbf{B}_R = \mathbf{0}\\
&\tilde{\mathbf{P}}_n\mathbf{E}_R + \tilde{\mathbf{Q}}_{n+1}\mathbf{F}_R + \tilde{\mathbf{P}}_{n+1}\mathbf{A}_R + \tilde{\mathbf{Q}}_{n+2}\mathbf{B}_R = \mathbf{0};\ R \le n \le K-2\\
&\tilde{\mathbf{P}}_{K-1}\mathbf{E}_R + \tilde{\mathbf{Q}}_K\mathbf{F}_R + \tilde{\mathbf{P}}_K\mathbf{A}_K = \mathbf{0}
\end{aligned}$$

We now have the result of appropriate matrix manipulation and recursive substitution as

$$\tilde{\mathbf{Q}}_1 = \tilde{\mathbf{P}}_0\mathbf{A}_0(-\mathbf{B}_1^{-1}) \tag{3.18}$$

$$\tilde{\mathbf{Q}}_n = \tilde{\mathbf{P}}_n\mathbf{D}_n(-\mathbf{C}_n^{-1}) = \tilde{\mathbf{P}}_n\mathbf{X}_n; 1 \le n \le R \tag{3.19}$$

$$\tilde{\mathbf{Q}}_n = \tilde{\mathbf{P}}_n\mathbf{D}_R(-\mathbf{C}_R^{-1}) = \tilde{\mathbf{P}}_n\mathbf{X}_n; R+1 \le n \le K \tag{3.20}$$

$$\tilde{\mathbf{P}}_n = \left[\tilde{\mathbf{P}}_{n+1}(\mathbf{X}_{n+1}\mathbf{F}_{n+1} + \mathbf{A}_{n+1}) + \tilde{\mathbf{P}}_{n+2}\mathbf{X}_{n+2}\mathbf{B}_{n+2}\right]\{-\mathbf{E}_n^{-1}\}; 0 \le n \le R-2 \tag{3.21}$$

$$\tilde{\mathbf{P}}_{R-1} = \left[\tilde{\mathbf{P}}_R(\mathbf{X}_R\mathbf{F}_R + \mathbf{A}_R) + \tilde{\mathbf{P}}_{R+1}\mathbf{X}_{R+1}\mathbf{B}_R\right]\{-\mathbf{E}_{R-1}^{-1}\} \tag{3.22}$$

$$\tilde{\mathbf{P}}_n = \left[\tilde{\mathbf{P}}_{n+1}(\mathbf{X}_{n+1}\mathbf{F}_R + \mathbf{A}_R) + \tilde{\mathbf{P}}_{n+2}\mathbf{X}_{n+2}\mathbf{B}_R\right]\{-\mathbf{E}_R^{-1}\}; R \le n \le K-2 \tag{3.23}$$

$$\tilde{\mathbf{P}}_{K-1} = \left[\tilde{\mathbf{P}}_K(\mathbf{X}_K\mathbf{F}_R + \mathbf{A}_K)\right]\{-\mathbf{E}_R^{-1}\} \tag{3.24}$$

wherein $\mathbf{X}_n$ has the closed form as follows

$$\mathbf{X}_n = \begin{cases} \mathbf{D}_n(-\mathbf{C}_n^{-1});\ 1 \le n \le R \\ \mathbf{D}_R(-\mathbf{C}_R^{-1});\ R+1 \le n \le K \end{cases}$$

Using Eqns. 3.16, 3.19 and 3.20,

$$\mathbf{\Pi} = \left[\tilde{\mathbf{P}}_0, \tilde{\mathbf{P}}_1\mathbf{X}_1, \tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2\mathbf{X}_2, \tilde{\mathbf{P}}_2, ..., \tilde{\mathbf{P}}_K\mathbf{X}_K, \tilde{\mathbf{P}}_K\right] \tag{3.25}$$

We further use the normalizing condition of probability as

$$\mathbf{\Pi e} = 1$$

where $\mathbf{e} = [1,1,...,1]^T$ is column vector of dimension $2K+1$ having all entries 1.

$$(\tilde{\mathbf{P}}_0 + \tilde{\mathbf{P}}_1\mathbf{X}_1)\mathbf{e}_1 + (\tilde{\mathbf{P}}_1 + \tilde{\mathbf{P}}_2\mathbf{X}_2)\mathbf{e}_2 + ... + (\tilde{\mathbf{P}}_{K-1} + \tilde{\mathbf{P}}_K\mathbf{X}_K)\mathbf{e}_K + \tilde{\mathbf{P}}_K\mathbf{e}_{K+1} = 1$$
$$\sum_{i=1}^{K}(\tilde{\mathbf{P}}_{i-1} + \tilde{\mathbf{P}}_i\mathbf{X}_i)\mathbf{e}_i + \tilde{\mathbf{P}}_K\mathbf{e}_{K+1} = 1 \tag{3.26}$$

where, $\mathbf{e}_i$ is column vector of dimension $(i+1)$ with all entries 1.
At this end, we have $K+1$ variables as $\tilde{\mathbf{P}}_0, \tilde{\mathbf{P}}_1, ..., \tilde{\mathbf{P}}_K$ and $K+1$ equations as 3.21, 3.22, 3.23, 3.24 and 3.26 which can be solved to obtain these variable values. Further $\tilde{\mathbf{Q}}_i$'s are obtained from $\tilde{\mathbf{P}}_i$'s by equations 3.19 and 3.20.

3.4 System Performance Measures

The acceptability of any model of the queueing problems can be best interpreted in terms of its system characteristics. Here, several indices viz. expected customers' count in the system, expected subordinate servers count in queue, waiting time , throughput, etc. are obtained in order to endorse the system's applicability. Various system indices are expressed in vector form as:

- Expected number of customers in the system

$$L_c = \sum_{n_1=0}^{K} n_1 P_{n_1,0} + \sum_{n_1=0}^{K}\sum_{n_2=1}^{R} n_1(P_{n_1,n_2} + Q_{n_1,n_2})$$
$$= \sum_{i=0}^{R} i\tilde{\mathbf{P}}_i\mathbf{c}_i + \sum_{i=R+1}^{K} i\tilde{\mathbf{P}}_i\mathbf{c}_R + \sum_{i=1}^{R} i\tilde{\mathbf{Q}}_i\mathbf{d}_i + \sum_{i=R+1}^{K} i\tilde{\mathbf{Q}}_i\mathbf{d}_R \tag{3.27}$$

  where $\mathbf{c}_i$, $\mathbf{c}_R$, $\mathbf{d}_i$, and $\mathbf{d}_R$ are column vectors of dimension $(i+1)$, $(R+1)$, i and R respectively, consisting of all entries 1.

- Expected number of subordinate servers waiting in the system

$$L_s = \sum_{n_1=1}^{R}\sum_{n_2=1}^{n_1}(n_1 - n_2)[P_{n_1,n_2} + Q_{n_1,n_2}] + \sum_{n_1=R+1}^{K}\sum_{n_2=1}^{R}(R - n_2)[P_{n_1,n_2} + Q_{n_1,n_2}]$$
$$= \sum_{i=2}^{R-1}(\tilde{\mathbf{P}}_i + \tilde{\mathbf{Q}}_i)\mathbf{s}_i + \sum_{i=R}^{K}(\tilde{\mathbf{P}}_i + \tilde{\mathbf{Q}}_i)\mathbf{s}_R \tag{3.28}$$

where $\mathbf{s}_i = [0, i-1, i-2, ..., 1, 0]^T$ and $\mathbf{s}_R = [0, R-1, R-2, ..., 1, 0]^T$ are column vectors of dimensions $(i+1)$ and $(R+1)$, respectively.

- Expected waiting time of customer

$$W_c = \frac{L_c}{\lambda_{effc}} \tag{3.29}$$

where λ_{effc} is effective arrival rate for customers given by

$$\begin{aligned}\lambda_{effc} &= \sum_{n_1=0}^{R-1}\sum_{n_2=1}^{n_1} \lambda P_{n_1,n_2} + \sum_{n_1=R}^{K-1}\sum_{n_2=1}^{R} \lambda P_{n_1,n_2} \\ &= \sum_{i=0}^{R-1} \lambda \tilde{\mathbf{P}}_i \mathbf{u}_i + \sum_{i=R}^{K-1} \lambda \tilde{\mathbf{P}} \mathbf{u}_R\end{aligned}$$

where $\mathbf{u}_i$ and $\mathbf{u}_R$ are column vector of dimensions $(i+1)$ and $(R+1)$, respectively with all entries 1.

- Expected waiting time of subordinate server

$$W_s = \frac{L_s}{\lambda_{effs}} \tag{3.30}$$

where λ_{effs} is effective arrival rate for subordinate servers given by

$$\begin{aligned}\lambda_{effs} &= \sum_{n_1=1}^{R-1}\sum_{n_2=0}^{n_1-1} (n_1 - n_2)\mu P_{n_1,n_2} + \sum_{n_1=R}^{K}\sum_{n_2=0}^{R-1} (R - n_2)\mu P_{n_1,n_2} \\ &= \mu \sum_{i=1}^{R-1} \tilde{\mathbf{P}}_i \mathbf{v}_i + \mu \sum_{i=R}^{K} \tilde{\mathbf{P}}_i \mathbf{v}_R\end{aligned}$$

where $\mathbf{v}_i = [i, i-1, i-2, ..., 1, 0]^T$ and $\mathbf{v}_R = [R, R-1, R-2, ..., 1, 0]^T$ are column vectors of dimensions $(i+1)$ and $(R+1)$, respectively.

- The probability that customer immediately after service is rendered

$$\begin{aligned}P_u &= \sum_{n_1=1}^{R}\sum_{n_2=1}^{n_1} Q_{n_1,n_2} + \sum_{n_1=R+1}^{K}\sum_{n_2=1}^{R} Q_{n_1,n_2} \\ &= \sum_{i=1}^{R} \tilde{\mathbf{Q}}_i \mathbf{w}_i + \sum_{i=R+1}^{K} \tilde{\mathbf{Q}}_i \mathbf{w}_R\end{aligned} \tag{3.31}$$

where $\mathbf{w}_i$ and $\mathbf{w}_R$ are column vector of dimensions i and R, respectively with all entries 1.

- Throughput of the successful customer in the system

$$\begin{aligned}\tau_{ps} &= \sum_{n_1=1}^{R-1}\sum_{n_2=0}^{n_1-1}(\mu+\alpha+\beta_1)P_{n_1,n_2} + \sum_{n_1=R}^{K}\sum_{n_2=0}^{R-1}(\mu+\alpha+\beta_1)P_{n_1,n_2} \\ &= (\mu+\alpha+\beta_1)\sum_{i=1}^{R}\tilde{\mathbf{P}}_i\mathbf{a}_i + (\mu+\alpha+\beta_1)\sum_{i=R+1}^{K}\tilde{\mathbf{P}}_i\mathbf{a}_R \end{aligned} \tag{3.32}$$

  where $\mathbf{a}_i = [1,1,1,...,1,0]^T$ and $\mathbf{a}_R = [1,1,1,...,1,0]^T$ are column vectors of dimensions $(i+1)$ and $(R+1)$, respectively.

- Throughput of the unsuccessful customer in the system

$$\begin{aligned}\tau_{pu} &= \sum_{n_1=1}^{R-1}\sum_{n_2=0}^{n_1-1}(\mu+\alpha+\beta_2)P_{n_1,n_2} + \sum_{n_1=R}^{K}\sum_{n_2=0}^{R-1}(\mu+\alpha+\beta_2)P_{n_1,n_2} \\ &= (\mu+\alpha+\beta_2)\sum_{i=1}^{R}\tilde{\mathbf{P}}_i\mathbf{a}_i + (\mu+\alpha+\beta_2)\sum_{i=R+1}^{K}\tilde{\mathbf{P}}_i\mathbf{a}_R \end{aligned} \tag{3.33}$$

  where $\mathbf{a}_i$ and $\mathbf{a}_R$ are given as above vectors.

- Frequency of system full

$$\begin{aligned}FF &= \sum_{n_2=0}^{R}\lambda P_{K-1,n_2} + \sum_{n_2=1}^{R}\beta_2 Q_{K,n_2} \\ &= \lambda\tilde{\mathbf{P}}_{K-1}\mathbf{f}_{R+1} + \beta_2\tilde{\mathbf{Q}}_K\mathbf{f}_R \end{aligned} \tag{3.34}$$

  where $\mathbf{f}_R$ and $\mathbf{f}_{R+1}$ are column vector of dimensions R and $(R+1)$, respectively with all entries 1.

3.5 Computation of the Cost Function

This section is dedicated to constructing a cost function using various cost components encountered in the system. Several components of the cost per unit time associated with distinct occurrences are used for this purpose. The cost components that are included in the cost function are:

$Ch_1 \equiv$ Holding cost incurred for each customer present in the system.

$Ch_2 \equiv$ Holding cost incurred for each subordinate server waiting for the approval phase service.

$C_r \equiv$ Cost associated per subordinate server.

$C_m \equiv$ Cost incurred for providing the service with rate μ by subordinate server in preparation phase.
$C_a \equiv$ Cost incurred for providing the service with rate α by chief server in the approval phase.
Now, the total cost function per unit time is constructed by combining the different cost aspects mentioned above with system performance indices such as-

$$TC(\mu,\alpha) = Ch_1L_c + Ch_2L_s + C_rR + C_m\mu + C_a\alpha \tag{3.35}$$

The governing optimization problem is developed as

$$TC^*(\mu^*,\alpha^*) = Min\{TC(\mu,\alpha)\} \tag{3.36}$$

We opt the metaheuristic optimization technique TLBO discussed in the coming section to compute the optimal value of deciding parameters (μ^*,α^*).

3.6 Teaching Learning Based Optimization Algorithm

Inspiration

The TLBO method is based on the impact of the teacher's influence on the output of students in a classroom, which is discussed in more details in subection 1.8.2.

Teacher Phase

Let M_k be the mean and T_k be the teacher at any iteration k. T_k will attempt to move mean M_k towards its own level, therefore the new mean will be T_k labelled as M_{new}. The solution is updated according to the difference between the existing and the new mean is given by

$$DM_k = r_k\,(M_{new} - T_F M_k) \tag{3.37}$$

where T_F is a teaching factor that decides the change in mean value, and r_k is a random vector uniformly distributed within $[0,1]^D$. The value of T_F can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability.
The position of each learner in the k^{th} iteration is updated by the following equation

$$X_{k,new} = X_{k,old} + DM_k \tag{3.38}$$

where $X_{k,new} = (X^1_{k,new}, ..., X^D_{k,new})$ and $X_{k,old} = (X^1_{k,old}, ..., X^D_{k,old})$ are the k^{th}learner's new and old positions, respectively. If $X_{k,new}$ is better than $X_{k,old}$, it is accepted; otherwise $X_{k,old}$ is unchanged.

Algorithm 3 Pseudo code for TLBO

Input: Initialize number of learners L, dimension D, iterations t_{max};
while $t < t_{max}$ or convergence criterion **do**
    Choose the best learner as T_k;
    Calculate the mean M_k of all learners;
    for $1 \leq i \leq L$ **do**
        $TF = \text{round}(1 + \text{rand}(0,1))$
        Update the learner according to Eq. 3.38;
        Evaluate $f(X_{k,new})$;
        if $f(X_{k,new}) < f(X_{k,old})$ **then**
            Update $X_{k,old}$ with $X_{k,new}$
        else $X_{k,old}$ unchanged
        end if
        Randomly select another learner $X_l (k \neq l)$
        if $f(X_k) \leq f(X_l)$ **then**
            Update the learner according to Eq. 3.39;
        else Update the learner according to Eq. 3.40;
            Evaluate the new learner $X_{k,new}$;
            if $f(X_{k,new}) < f(X_{k,old})$ **then**
                Update $X_{k,old}$ with $X_{k,new}$
            else $X_{k,old}$ unchanged
            end if
        end if
    end for
end while

Learner Phase

Learners acquire knowledge in two different manners: one from teacher input and the other through peer interaction. With the support of group discussions, presentations, formal communications, and other means, a learner connects with other learners at random. If another student has greater knowledge than the learner, the learner learns something new.
Learner X_k randomly selects another learner X_l $(k \neq l)$ and the learning process can be divided into two cases as:
Case I If $f(X_k) \leq f(X_l)$

$$X_{k,new} = X_{k,old} + r_k(X_k - X_l) \tag{3.39}$$

Case II If $f(X_k) > f(X_l)$

$$X_{k,new} = X_{k,old} + r_k(X_l - X_k) \tag{3.40}$$

where $f(X)$ is the objective function with $D-$dimensional variables. If $X_{k,new}$ is better than $X_{k,old}$, it is accepted [30], [159],[233].

3.7 Numerical Illustrations of the Model

To illustrate the practicability of the presented model, numerical experiments are carried out with the help of MATLAB software. The values of the default parameters are set as $K = 30$, $R = 4$, $\lambda = 2.5$, $\mu = 2$, $\alpha = 5$, $\beta_1 = 1$, and $\beta_2 = 0.5$ to satisfy the requirements of the model described in Section 3.2.

An increase in system capacity largen up the space for more accommodation of customers, which in result increases mean number of customers in the system (L_c) as depicted in Fig 3.1. The larger system size has more customers served in the first phase of service by subordinate servers, increasing to L_s in the queue of the chief server (Fig 3.2). The queue size is directly related to the waiting time; thus, W_c and W_s are also higher for larger K (Fig 3.3,3.4). The entry of customers into the system leads to the formation of the queue; therefore, increasing the arrival rate tends to increase queue size (as observed in Fig 3.1(i), Fig 3.2(i)) and waiting time (as shown in Fig 3.3(i) and Fig 3.4(i)) as well. From the common notion, it can be easily deducted that a larger queue size hampers the customer's patience due to increased waiting time in the system. The inspection for successful service (β_1) directes that customers leave the system after service completion of all phases. Also, subordinate servers get back to perform preliminary phase service to newly arriving customers and depart from the chief server queue, as clearly observed in Figs 3.2(iv) and 3.4(iv). The peculiar pattern is observed for smaller β_1 due to capacity constraints in which subordinate servers perform the preliminary phase sooner in lesser congestion but trend changes when service improves further. The inspection for unsuccessful service β_2 ultimately determines the number of customers rejoin again in the system due to service failure. Thus, the pattern followed by graphs of L_c, L_s, and W_c are similar as that of λ (see Fig 3.1(v), Fig 3.2(v), and Fig 3.3(v)). In a queueing system, service regimes are crucial in determining the satisfaction level of customers. In our model, service is provided cumulatively by subordinate servers in parallel and a chief server in the phases in tandem. Each phase's service rate is vital in deciding queue size and waiting time. An improvement in the service of subordinate servers results in an apparent reduction in queue length and waiting time (see Fig 3.1(ii, iii, iv) and Fig 3.3(ii, iii, iv)).

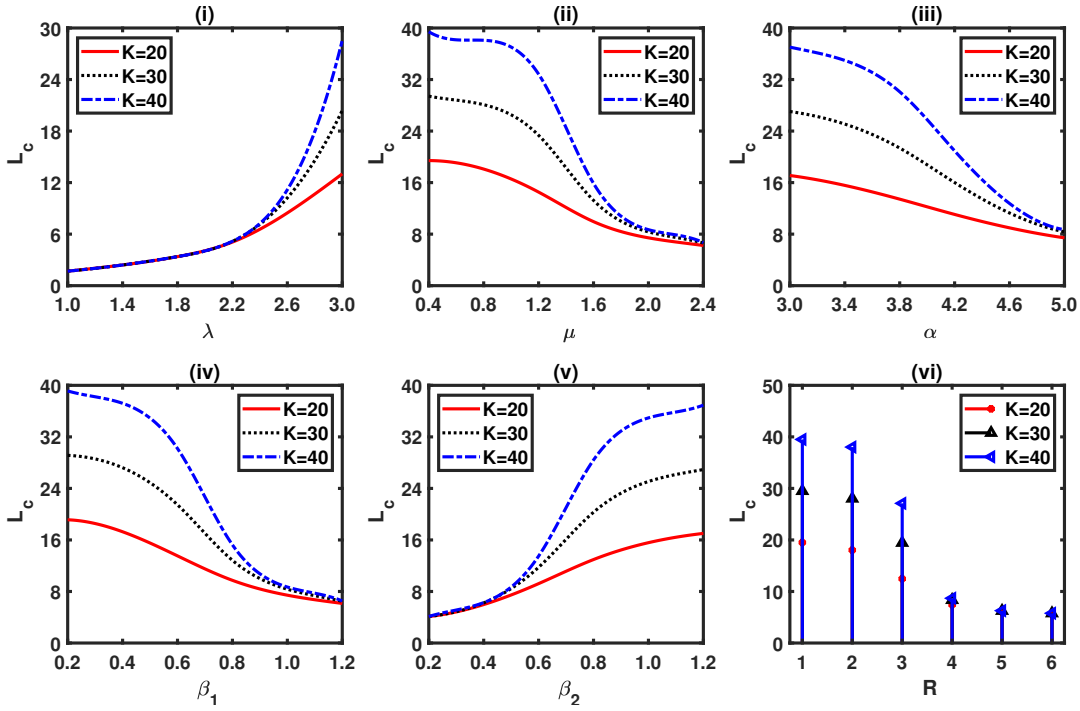


Figure 3.1: Expected number of customer in the system (L_c) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

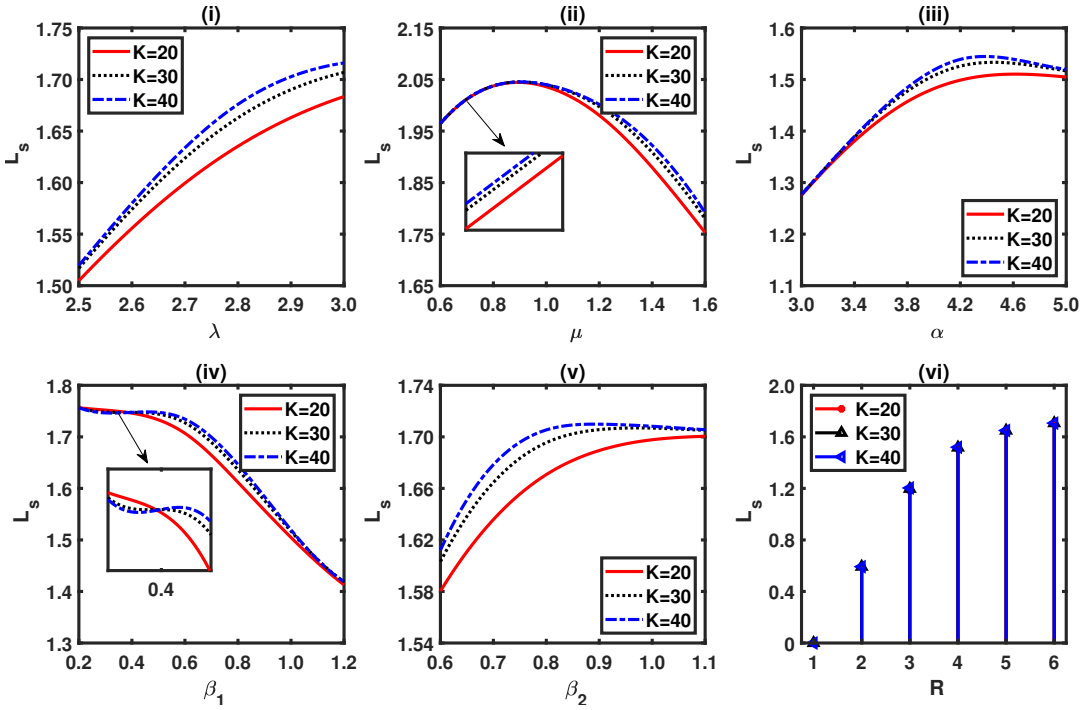


Figure 3.2: Expected number of subordinate server waiting in the system (L_s) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

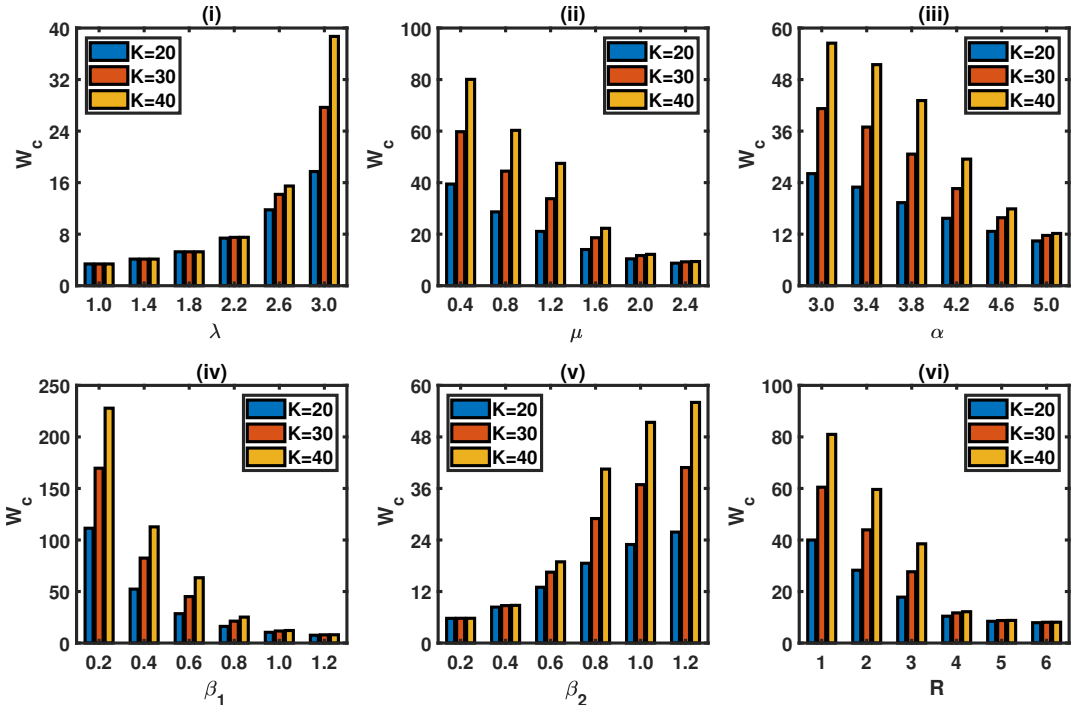


Figure 3.3: Expected waiting time of customer (W_c) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

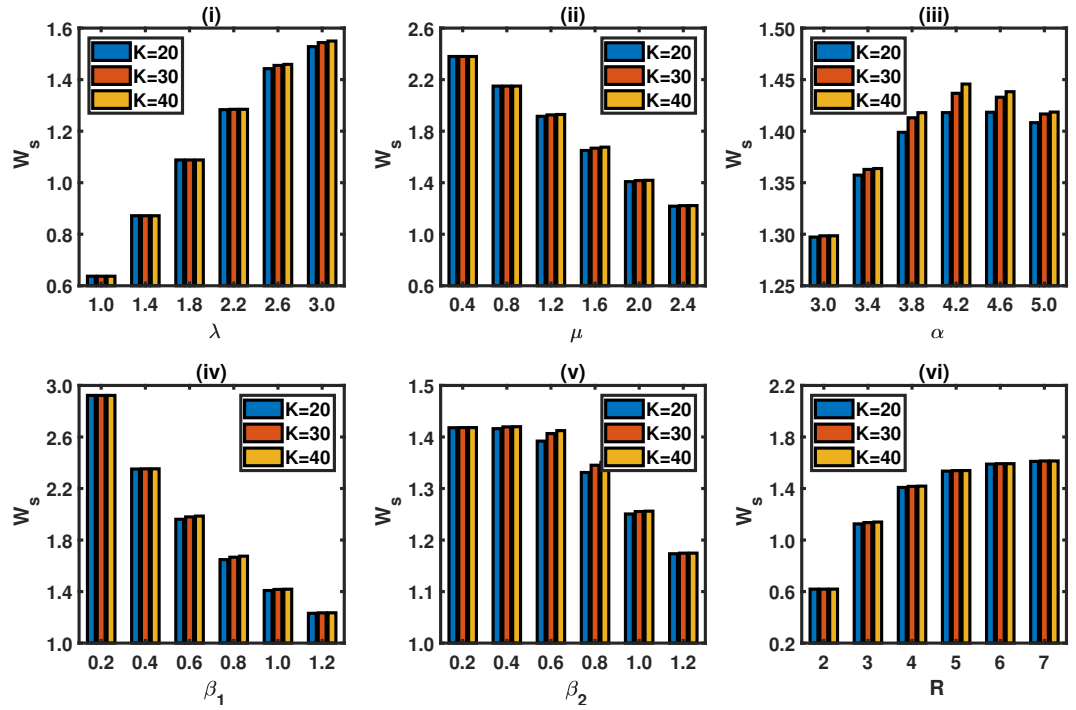


Figure 3.4: Expected waiting time of subordinate server (W_s) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

As described in the model description (Section 3.2), subordinate servers are arranged in parallel to serve customers in the preliminary phase, so each server has a different queue. An increment in the number of subordinate servers increases the number of customers served in the preliminary phase and reduces their queue length and waiting time (see Fig 3.1(vi) and Fig 3.3(vi)). From the earlier explanation, it can be easily deduced that an increased number of subordinate servers queue up for the second phase service to be provided by the chief server (see Fig 3.2(vi) and Fig 3.4(vi)).

The system's throughput measures the number of customers the service facility serves in a unit time. The incoming customers in the service system enhance the throughput count to a certain extent but gradually decrease by increasing the arrival rates beyond some limit. The trend of the number of customers beyond system capacity and upcoming customers is lost (see Fig 3.5(i)). In the service systems where service is rendered in phases, customers depart after all phases of service are completed. In the presented model, there is a single chief server, so an increased number of subordinate servers and their service rate (μ) increase congestion in the final phase, as shown in Fig 3.5(ii), (vi). On the contrary, the throughput of successful customers increases when the final phase of service improves (see Fig 3.5 (iii, iv, v)). The throughput of unsuccessful customers in the system also goes through the same procedures as customers whose service is successful. After completing the service, customers decide whether the service is reliable or unreliable. On that note, we can observe that the trend of τ_{pu} is similar to τ_{ps} for all the parameters, as shown in Fig 3.6.

The frequency of the system being full, i.e., the arriving prospective customers are lost, is determined by the flow of customers in and out. The trend of the FF graph is an easy observation of the fact that maximum customer accommodation is reached by increasing the arrival rate (see Fig 3.7 (i), (v)) and decreased by increasing the service rate (see Fig 3.7 (ii), (iii), (iv), (vi)).

3.8 Sensitivity Analysis of the Model

The computation of the total cost (TC) is done by first choosing following unit cost elements $C_{h_1} = 100$, $C_{h_2} = 300$, $C_r = 50$, $C_m = 5$, and $C_a = 30$ as default cost parameters. The result of illustrations are depicted in Fig 3.8-3.11 and Tables 3.1-3.2.

To benefit the cost analysis from an economic viewpoint, the function of cost $TC(\mu, \alpha)$ is plotted against system parameters in Fig 3.8 and Fig 3.9 as surface plot, line graph, contour plot. We observe that the graph of TC is convex for parameters μ and α (see Fig 3.8(ii), Fig 3.9(i,ii)) that prompt μ and α as decision parameters. The waiting time of customers in the system needs to be minimized to make the system cost-effective. We achieve a low-cost

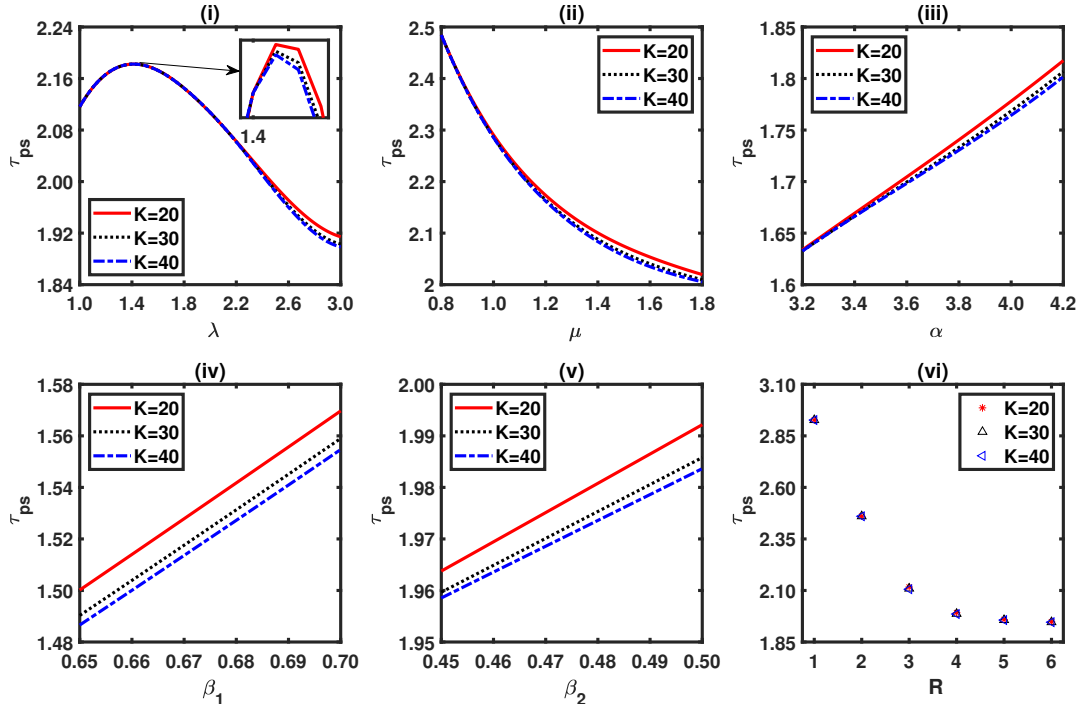


Figure 3.5: Throughput of the successful customer in the system (τ_{ps}) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

service system by providing a better service facility by increasing the number of subordinate servers. There is cost associated with each server, so if there is an increase in the number of subordinate servers after a specific optimal value, cost increases. Thus, TC is convex due to the trending nature (Fig 3.9 (iii, v, vi) and needs to balance the incurred cost. The depicted trend prompts that all decision parameters are favorable in system design and play an essential part in advancing the presented model. The results are fascinating and may be used for future upgradation in realistic queueing problems of multi-stage services with a reasonable selection of service rates. We employ the meta-heuristic technique TLBO to investigate the effect of the system's attributes on the optimal total cost $TC(\mu^*, \alpha^*)$.

Many population-based metaheuristic optimization approaches, like TLBO, are iterative techniques. When the iterations proceed, all search solution points converge to the best, defined as the search point with the optimal values of the decision variables, as illustrated in Fig 3.10. The total cost ($TC(\mu^*, \alpha^*)$ value is lowest at μ^* and α^*. The convergence of multiple runs with distinct initial sets of the solution (population) is depicted in Fig 3.11. It is observed that TLBO converges to the optimal solution point after some generation from the distinct initial set of solution points.

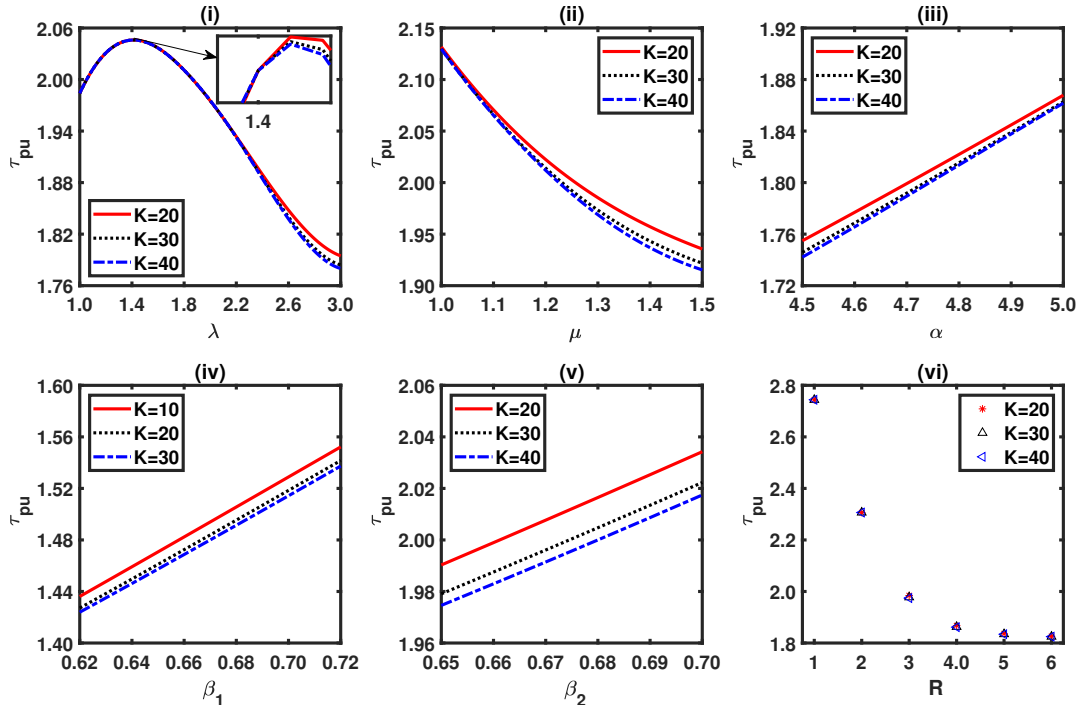


Figure 3.6: Throughput of the unsuccessful customer in the system (τ_{pu}) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

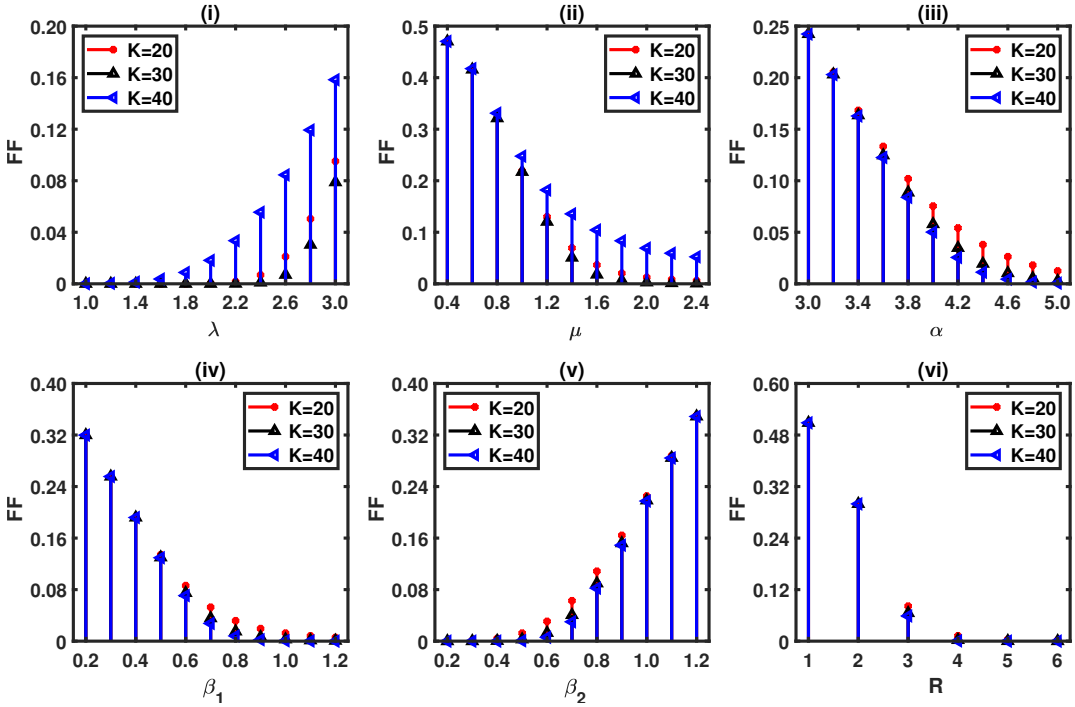


Figure 3.7: Frequency of system full (FF) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

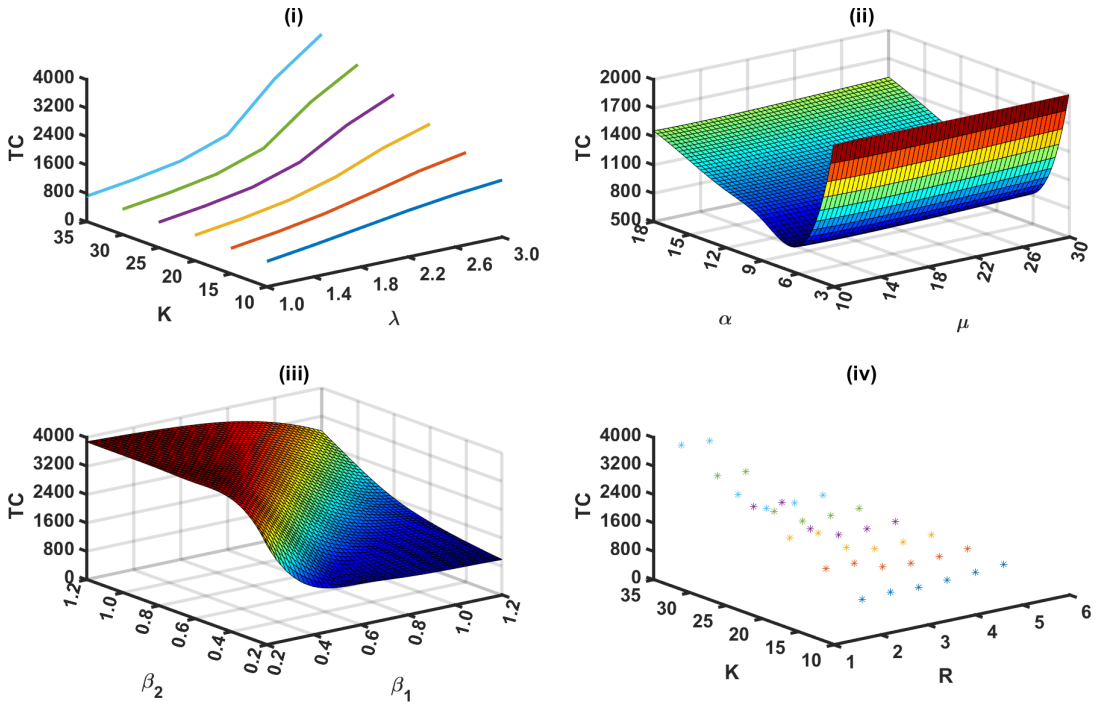


Figure 3.8: Surface plot for total cost of the system (TC) wrt combinations of (i) (K,λ) (ii) (μ,α) (iii) (β_1,β_2) (iv) (K,R).

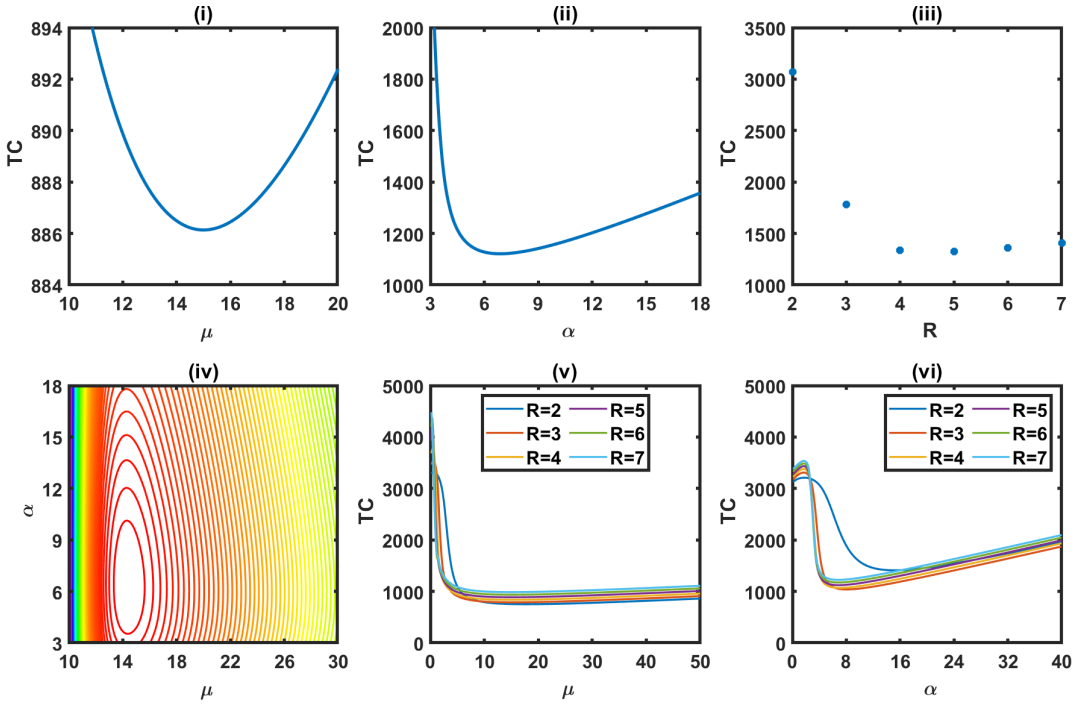


Figure 3.9: Total cost of the system (TC) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

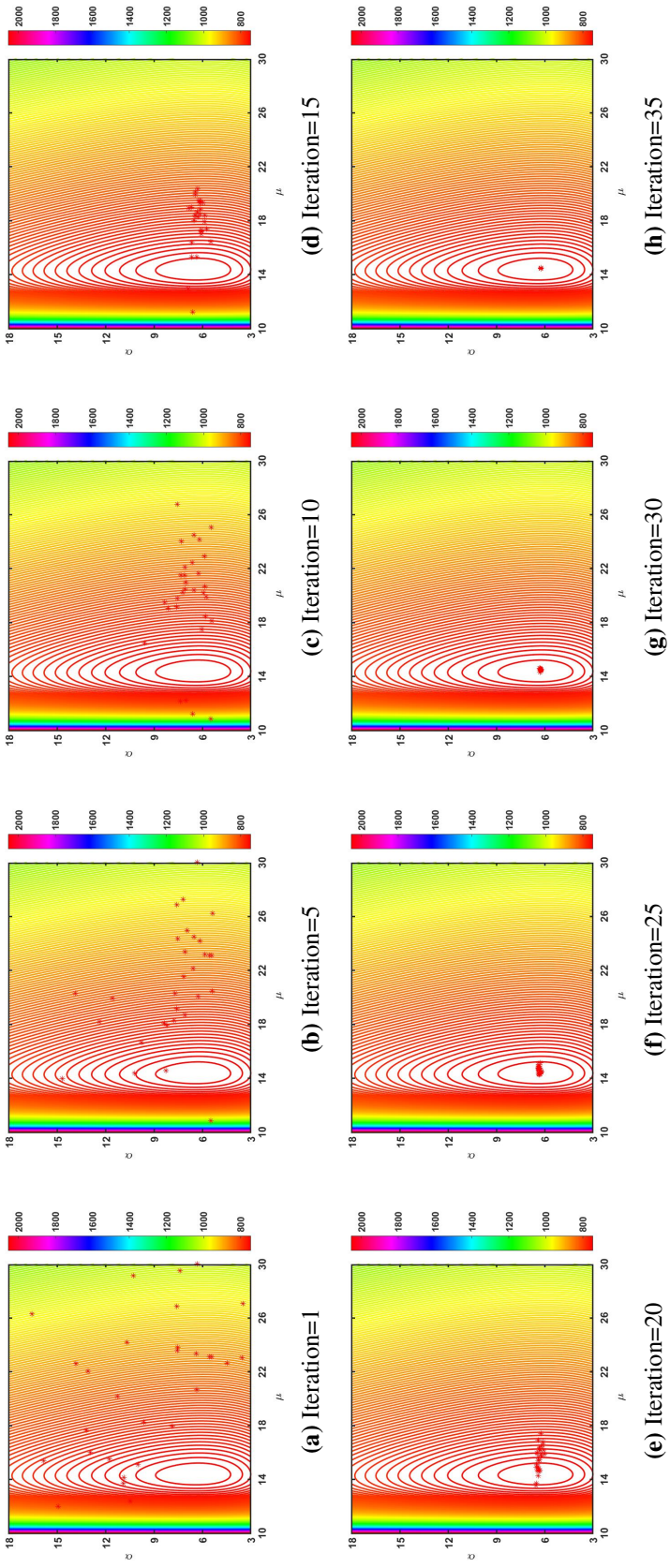


Figure 3.10: Convergence of iteration of TLBO algorithm on the contour of $TC(\mu, \alpha)$

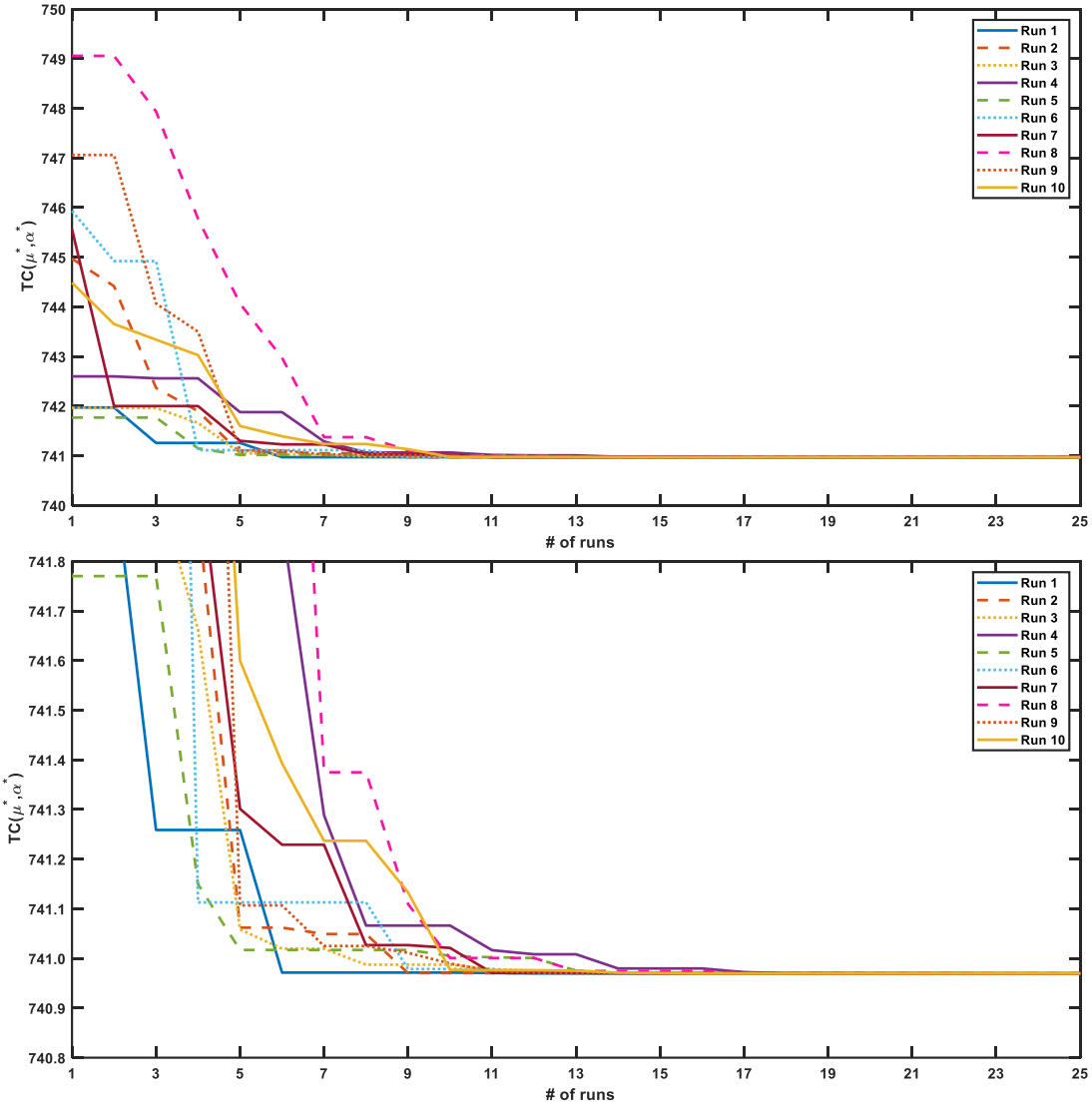


Figure 3.11: Total cost of the system (TC) wrt (i) λ, (ii) μ, (iii) α, (iv) β_1, (v) β_2, and (vi) R for different values of K.

Tables 3.1 and 3.2 comprise the optimal value of decision parameters (μ^*, α^*) for minimum cost for a different set of governing parameters and cost elements, respectively. For the demonstration purpose, we use a randomly generated population of size 50, 30 independent generations, and 10 independent numbers of run along with the lower and upper limits for μ and α as $[3, 10]$ and $[18, 30]$, respectively, and obtain optimum values of the continuous decision parameters up to the fifth decimal place in both the Tables 3.1 and 3.2. To demonstrate the resilience of the TLBO method, we employ the notion of statistical inferences, particularly the mean and maximum of the ratio of optimum $TC^*(\mu^*, \alpha^*)$ in all runs and optimal $TC^*(\mu^*, \alpha^*)$ in each run. Furthermore, we can see from both Tables 3.1 and 3.2 that the mean and maximum values of $\frac{TC}{TC^*}$ are between 1.00000000018273621 and 1.00000000098926489, where TC is the best (minimum) solution among ten independent TLBO runs and TC^* is the best (minimum) solution among ten independent TLBO runs.

In a nutshell, we infer the following points

- Appropriate service facility and system design are needed to develop a better service system.
- Optimal decision values are fixed to optimize the incurred cost.
- Precautions need to be taken in service to avoid unreliable service, delay in service, etc.

3.9 Conclusion

In this chapter, we investigated a finite capacity service system with a subordinate-chief server approach, and the service is unreliable. The service is provided in two phases, viz preparation and approval phase by subordinate and chief servers, respectively. In the approval phase, subordinate servers behave like customers and seek service from the chief server on customers' behalf. The service is finally completed by the chief server and the subordinate server's assistance. At the end of service completion, the customer decides whether the service is reliable or unreliable. Suppose a customer finds the service unreliable, s/he retries for service. We apply the repeated substitution approach to determine the probabilities of the system in the steady-state. Various system performancemeasures of the system are derived in vector form for developing their numerical simulations. The provided model is subjected to an economic analysis by finding the total cost function for decision parameters μ and α for cost minimization and utility maximization. The optimal results are computed by employing the meta-heuristic technique TLBO on the governing total cost

Table 3.1: Optimal expected total cost of the system $TC^*(\mu^*, \alpha^*)$ for different parameters via TLBO algorithm

$(K,R,\lambda,\beta_1,\beta_2)$	μ^*	α^*	$TC^*(\mu^*, \alpha^*)$	**mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**time elapsed**
25,5,2,1,0.5	14.44834	6.249860	740.9696173	2.2684614	2.3657424	59.2951031
30,5,2,1,0.5	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
35,5,2,1,0.5	14.44833	6.249865	740.9696461	7.3763543	5.7864344	131.2734026
30,4,2,1,0.5	14.43507	6.250329	690.9298334	6.7834726	2.7349624	68.9491403
30,5,2,1,0.5	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
30,6,2,1,0.5	14.44977	6.249827	790.9731531	3.7364813	5.7836494	115.9435074
30,5,1.5,1,0.5	12.24296	5.080814	664.3437405	4.8736484	9.8926489	94.4225831
30,5,2,1,0.5	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
30,5,2.5,1,0.5	16.38153	7.368267	809.9505589	2.3748514	3.8936477	92.1668645
30,5,2,0.5,0.5	17.22878	7.709962	841.4205319	5.8973626	2.7836481	93.4665737
30,5,2,1,0.5	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
30,5,2,1.5,0.5	13.2848	5.756982	699.0023806	4.8937628	2.3786482	96.6126368
30,5,2,1,0.5	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
30,5,2,1,1.0	16.76931	7.75218	823.5456298	2.8937672	4.8937467	91.9283815
30,5,2,1,1.5	18.82257	9.19389	899.9494823	5.8736415	2.8734617	92.0926648

Table 3.2: Optimal expected total cost of the system $TC^*(\mu^*, \alpha^*)$ for different parameters via TLBO algorithm

(Ch_1,Ch_2,C_r,C_m,C_a)	μ^*	α^*	$TC^*(\mu^*, \alpha^*)$	**mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**time elapsed**
50,300,50,5,30	13.55394	5.328694	643.1670084	3.8634826	6.8726342	95.5537925
100,300,50,5,30	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
150,300,50,5,30	15.35755	6.953697	826.0116315	4.8372640	5.6753887	95.9766983
100,200,50,5,30	12.61877	6.228975	722.6417680	3.8963481	5.8376421	93.9139876
100,300,50,5,30	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
100,400,50,5,30	16.06988	6.267514	757.2160720	5.8726348	7.8736429	90.5615411
100,300,40,5,30	14.44834	6.249864	690.9696453	7.8396472	8.8973642	93.2421033
100,300,50,5,30	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
100,300,60,5,30	14.44833	6.249865	790.9696453	2.8973645	1.8273621	93.1271296
100,300,50,4,30	16.15592	6.240497	725.7150890	4.8973619	2.8947523	93.1602192
100,300,50,5,30	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
100,300,50,6,30	13.18776	6.258361	754.7589896	3.7465299	2.7361428	89.9132209
100,300,50,5,20	14.34058	7.003189	675.0897841	7.7836849	8.8361921	90.4970936
100,300,50,5,30	14.44833	6.249864	740.9696453	1.8734352	3.8765843	87.8886803
100,300,50,5,40	14.52486	5.803871	801.0761157	3.7863291	8.7863421	91.5151168

optimization problem. The cost analysis clearly communicates the viability and profitability of the established model.

Chapter 4

Admission Control Policy on Online and Impatience Attributes of Offline Customers in Multi-phase Queueing Systems

This chapter considers a two-phase service system like pre-registration and verification systems, token and service systems, prepayment, and service systems, etc. The arriving customers in the first phase can either join the queue and wait for their turn or directly seek service through the online application. In the next phase, the customers, through both modes, must be present physically in the system. The controllable online booking is conceptualized for the online-application users as, after a specific threshold limit, the online application customers will not be able to book online due to capacity constraints to benefit the customers waiting physically. There is a general tendency of the waiting customers to abandon the long queue in the first phase.

4.1 Introduction

A queueing problem often develops when a customer has to wait in one to receive a service. Adjustments must be made to the arriving units, the service facilities, or both to address the problem. For many years, queueing theory has been used in various industries, including banking, aviation, restaurants, transportation, supply chains, and tourist sectors, among many others. In this chapter, we studied a finite capacity queueing system in which arriving customers have two paths to receive service. In the first stage, either by physically being in the system or through websites or mobile applications, and in the next phase of service, customers need to be physically present in the system irrespective of the path chosen in the first stage of service.

Online ordering is becoming a cornerstone standard defining congestion management. Such online processes are employed in various real-world contexts, including selling airline tickets, cloud resource allocation, sponsored search, real-time bidding in display advertisement, dynamic fleet management, fog computing, and real-time ride-sharing [137], [26]. However, in this study, our primary focus is on employing a queueing theoretic approach to represent a service mechanism. We have also employed the arrival control policy, namely the F-policy, on the customers coming to the system through the app. In this policy, the server keeps providing service to the customer until it reaches the system's maximum capacity and then stops taking further customers until the queue length decreases to a pre-specified threshold value, F. Finding the best operating strategy that results in the lowest overall cost is the goal of controllable queueing models, service control, or arrival control. Gupta [71] was the first to provide steady-state analytical solutions for the F-policy $M/M/1/K$ queueing system with an exponential startup time. To get the steady-state probability distributions of the number of customers in the F-policy $G/M/1/K$ and $M/G/1/K$ queueing systems, Wang et al. [196], [195] present a recursive approach utilizing the supplementary variable technique and treating the supplementary variable as the remaining inter-arrival time. Ke et al. [99] extended the issue of controlling arrivals for an $M/M/1/K$ queuing system with an F-policy, suggesting that specific customers may want a second service in addition to the first crucial service. The methodological characteristics of the F-policy employed in the retrial queueing model, the vacation and working vacation model, the unreliable server model, and the non-Markov model were outlined by Jain et al. [93] to give a state-of-the-art of admission control F-policy. Since then, the study of controllable F-policy has been modified by many researchers, and extensive research has been carried out to enhance the use of arrival control policy in various queueing and machining systems (see [223], [206], [205], [133], [217]). In this manner, we can prioritize that customer who has arrived in the system

in stage 1 and prevent them from getting frustrated. In this way, app customers merely have something to lose and can retry again, but the customers who have already arrived in the system hoping to be served should ideally be given service. This phenomenon is common in services like restaurants, banks, booking counters, passport or visa renewal offices, etc., as they have the dual facility of providing the initial phase of service by giving them a token number for the service they seek.

Customers' frustration is reaching a tipping point where many are unwilling to wait in line longer than the minimum amount of time. Customers' impatient attitudes make it more challenging to simulate queuing systems. The economic aspect of queueing theory in consideration of customers' decision to join the queue has received significant study focus over the last few decades. In a queuing system with the balking phenomenon, each customer joins the queue if its length is smaller than the most extended queue length they will endure, according to Haight [72]. In the model being studied, we have also employed the impatience attribute of customers, namely balking, in which customers may show a tendency to leave the system when they find congestion that is intolerable or beyond their threshold value. This impatience behavior is only for the customers planning to physically join the queue of stage one. Since there can be no balking on the app users as they have not experienced congestion or long wait due to booking for tokens through the app being present outside of the system. Economou and Kanta [49, 50, 51] have analyzed balking with various $M/M/1$ queue types in great detail. Singh [176] examines a two-server queuing model with a predetermined probability of balking with two distinct types of customers. In an $M/G/1$ queue with several vacations, Economou et al. [70] studied equilibrium and socially optimal balking approaches. In the steady-state study of queueing systems, the phenomenon of balking has been handled as one of the stochastic components. Yechiali [221] investigated a GI/M/1 queuing process with a stationary balking mechanism and a linear cost-reward structure. As a result, there has been constant discussion of the steady-state analysis of queueing systems taking balking into account. For instance, in Arizono et al. [13], Jain et al. [91], Economou et al. [47], Lumb et al. [126], and Negahban [141]the corresponding Markovian queueing systems with balking have been taken into consideration.

In this study, it is assumed that different servers provide a two-phase service. Only a few studies have looked into multi-server queues using a two-phase service. An $M/M/R/K$ queue with finite capacity and two-phase service was examined by Yang et al. [214]. A similar approach was used by Ke et al. [101] to compute the stationary probability distribution of the system's customer count for an $M/M/R$ queue with a two-phase service. A multi-server retrial queue with a finite population, balking customers, and two-phase service was taken into consideration by Ahuja et al. [7]. Yeh et al. [222] analyzed a two-phase finite

capacity $M/M/1/K$ queueing system with $< p, F >$-policy.
The novelty of the present investigation is:

- To study a two-channel queueing model that addresses arrivals via two modes.
- To address the effect of arrival control policy on app customers.
- To address the customer's impatience attributes in offline mode.
- We are determining the optimal cost of the system using an efficient and recent meta-heuristic technique called the grasshopper optimization algorithm.

The remainder of this chapter are organized as follows. In the next Section 4.2, the model is described with system assumptions, notations, and states. The model is formulated as a quasi-birth-and-death process and steady-state analysis is done by implementing the repeated substitution method to determine the stationary distribution probabilities in Section 4.3. The system's performance measures are derived and computed in Section 4.4. The cost function is constructed for the model in Section 4.5. In Section 4.6, the grasshopper optimization algorithm is discussed in detail, along with its pseudo-code. Section 4.7 is devoted to compute the steady-state probabilities numerically and plot the graphs for various performance measures to check their sensitiveness to system parameters. Finally, Section 4.8 presents concluding remarks, and future scope of this study.

4.2 Problem Description and Formulation

In this study, we aim to develop a two phase state-dependent queueing model. In the initial phase of service, there are two ways in which a customer can join, either by being physically present in the system or through online websites or mobile applications. There is no restriction on customers for choosing any route of this phase service; it totally depends on their convenience, but they can only choose one path out of two, not both. In the next consecutive phase, customers must be present in the system in any way selected in the previous service phase.

4.2.1 Basic Assumptions and Notations

1. The arrivals are identically independent of each other and follow the Poisson process for both stages of service. The inter-arrival time between two customers follows exponential distributions for both the initial and final phase of service with parameters λ_1 and λ_2, respectively.

2. The arrivals controllable F-policy is applied to the customers directly approaching the second phase of service. In this policy, the server keeps serving customers until the maximum capacity, K is reached in the system and then stops allowing app customers to apply for service until the queue size reduces to the threshold limit set to be F.

3. At the epoch of queue size reduces till F, the server requires a startup time distributed exponentially with parameter γ to start allowing fellows to the system. After that, the system usually functions until reaching its maximum capacity, at which time the above process is repeated repeatedly.

4. There is also a limit on customers' accommodation in first phase for initially joining the system for the starting service, which is k. Once this amount is reached, the next arriving customer has to leave the system and is termed as a lost customer.

5. Increased congestion in the system leads customers to leave the system without being served, and such a phenomenon is known as balking. The reasons for withdrawing from the system can be an unexpected delay, getting late for the next task, or change of mind if it takes longer than expected waiting time, etc., The impatience behavior is seen in customers who are stuck in the congestion scenario. So, we have employed this balking behavior in stage 1 customers who decide to be present in the system for service. The probabilities of a customer balking away from the system without service or joining the system are complimentary to each other as $\bar{\xi}$ and ξ, respectively. The stage 2 customers can barely think of leaving the service in between after taking and paying for it, except for some exceptions. So, we have not included balking nature of the customer in stage 2 of service.

6. The inter-time between services also follows an exponential distribution for both initial and final phases with parameters μ and β, respectively, independent of each other.

4.2.2 Practical Justification of the Model

A practical situation related to our proposed model is the cinema booking system, where tickets can be booked online by receiving a token as a seat number and joining the system directly for the movie. Another available option is queueing for tickets at the counter and receiving the token. In both cases, either booking online or offline, the final service is received by physically being present in the system. The control policy applies to online users if the situation arises in a way that enough people have arrived at the counter. In such a case, online booking is stopped for a while until the line reduces to a limit set. The preference is given to offline users for two reasons: (i) such service is in premises where

other services like food restaurants, shopping malls, etc. Hence, it increases the chances for offline customers to use these services for benefit in several ways from an economic viewpoint. (ii) to value the time and presence of offline customers, whereas online customers can rebook for another slot. The counter customers also show impatient behavior and decide to balk if the time taken to get the ticket is too long.

4.2.3 System States

In order to analyze the presented system in steady state, it is formulated as a QBD process in three-dimensional CTMC with a state space $\{N_1(t), N_2(t), J(t)\}$, where $N_1(t)$ denotes the count of customers in first stage of service, $N_2(t)$ represents the count of customers in final service stage, and $J(t)$ represents the state in which customers are allowed or not allowed at time t. Let $N_1 \equiv \lim_{t\to\infty} N_1(t)$, $N_2 \equiv \lim_{t\to\infty} N_2(t)$, and $J \equiv \lim_{t\to\infty} J(t)$. The system steady-states joint probability distribution function is as follows:

$$P_{n_1,n_2} = \text{Prob}[N_1 = n_1, N_2 = n_2, J = 0]; \quad n_1 = 0,1,2,\ldots,k \;\&\; n_2 = 0,1,2,\ldots,K-1$$
$$Q_{n_1,n_2} = \text{Prob}[N_1 = n_1, N_2 = n_2, S = 1]; \quad n_1 = 0,1,2,\ldots,k \;\&\; n_2 = 0,1,2,\ldots,K$$

Thus, P_{n_1,n_2} and Q_{n_1,n_2} represents the long-run fraction of time that the system stays in state $(N_1 = n_1, N_2 = n_2, J = 0)$ and $(N_1 = n_1, N_2 = n_2, J = 1)$, respectively.

4.3 Steady-State Analysis

In order to construct the transition rate matrix $\mathbf{Q}$ of the corresponding QBD process, the system states are arranged in the following lexicographic order:
$\{P_{0,0}, P_{0,1}, P_{0,2}, ..., P_{0,K-1}, Q_{0,0}, Q_{0,1}, ..., Q_{0,K,}, P_{1,0}, P_{1,1}, ...P_{1,K-1}, Q_{1,0}, Q_{1,1}..., P_{k,0}, P_{k,1}, P_{k,2}$
$..., P_{k,K-1}, Q_{k,0}, Q_{k,1}, ..., Q_{k,K}\}$.
Hence, the equivalent block-tridiagonal structure of the transition rate matrix $\mathbf{Q}$ of the

continuous-time Markov chain (CTMC) is represented as follows:

$$\mathbf{Q} = \begin{bmatrix}
\mathbf{A}_0 & \mathbf{0} & \mathbf{C}_0 & & & & & & & & & & & \\
\mathbf{T}_0 & \mathbf{B}_0 & \mathbf{0} & \mathbf{D}_0 & & & & & & & & & & \\
\mathbf{L}_0 & \mathbf{0} & \mathbf{A}_1 & \mathbf{0} & \mathbf{C}_0 & & & & & & & & & \\
 & \mathbf{L}_1 & \mathbf{T}_0 & \mathbf{B}_1 & \mathbf{0} & \mathbf{D}_0 & & & & & & & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & & & & & & & \\
 & & & \mathbf{L}_0 & \mathbf{0} & \mathbf{A}_1 & \mathbf{0} & \mathbf{C}_0 & & & & & & \\
 & & & & \mathbf{L}_1 & \mathbf{T}_0 & \mathbf{B}_1 & \mathbf{0} & \mathbf{D}_0 & & & & & \\
 & & & & & \mathbf{L}_0 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \mathbf{C}_0 & & & & \\
 & & & & & & \mathbf{L}_1 & \mathbf{T}_0 & \mathbf{B}_2 & \mathbf{0} & \mathbf{D}_0 & & & \\
 & & & & & & & \mathbf{L}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \mathbf{C}_0 & & \\
 & & & & & & & & \mathbf{L}_3 & \mathbf{T}_0 & \mathbf{B}_2 & \mathbf{0} & \mathbf{D}_0 & \\
 & & & & & & & & & \ddots & \ddots & \ddots & \ddots & \ddots \\
 & & & & & & & & & & \mathbf{L}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\
 & & & & & & & & & & & \mathbf{L}_3 & \mathbf{T}_0 & \mathbf{B}_2 \\
 & & & & & & & & & & & & \mathbf{L}_2 & \mathbf{0} \\
 & & & & & & & & & & & & & \mathbf{L}_3
\end{bmatrix}$$

The Markov process' rate matrix $\mathbf{Q}$ is analogous to the quasi-birth and death process and the elements of the rate matrix $\mathbf{Q}$ as block sub-matrices are given as follows in element form as:

$\{\mathbf{A}_0\}_{1,1} = -(\lambda_1 + \gamma);\ \{\mathbf{A}_0\}_{i,i} = -(\lambda_1 + \beta + \gamma),\ i = 2,3,...,F$

$\{\mathbf{A}_0\}_{i,i} = -(\lambda_1 + \beta),\ i = F+1, F+2, ..., K;\ \{\mathbf{A}_0\}_{i,i+1} = \beta,\ i = 1,2,...,K-1$

$\{\mathbf{B}_0\}_{1,1} = -(\lambda_1 + \lambda_2);\ \{\mathbf{B}_0\}_{i,i} = -(\lambda_1 + \lambda_2 + \beta),\ i = 2,3,...,K$

$\{\mathbf{B}_0\}_{K+1,K+1} = -(\lambda_1 + \beta);\ \{\mathbf{B}_0\}_{i,i+1} = \beta,\ i = 1,2,...,K-1;\ \{\mathbf{B}_0\}_{i+1,i} = \lambda_2,\ i = 1,2,...,K$

$\{\mathbf{A}_1\}_{1,1} = -(\lambda_1 + \mu + \gamma);\ \{\mathbf{A}_1\}_{i,i} = -(\lambda_1 + \beta + \mu + \gamma);\ i = 2,3,...,F$

$\{\mathbf{A}_1\}_{i,i} = -(\lambda_1 + \beta + \mu),\ i = F+1, F+2, ..., K;\ \{\mathbf{A}_1\}_{i,i+1} = \beta;\ i = 1,2,...,K-1$

$\{\mathbf{B}_1\}_{1,1} = -(\lambda_1 + \lambda_2 + \mu);\ \{\mathbf{B}_1\}_{i,i} = -(\lambda_1 + \lambda_2 + \beta + \mu);\ i = 2,3,...,K-1$

$\{\mathbf{B}_1\}_{K,K} = -(\lambda_1 + \lambda_2 + \beta);\ \{\mathbf{B}_1\}_{K+1,K+1} = -(\lambda_1 + \beta)$

$\{\mathbf{B}_1\}_{i,i+1} = \beta,\ i = 1,2,...,K-1;\ \{\mathbf{B}_1\}_{i+1,i} = \lambda_2,\ i = 1,2,...,K$

$\{\mathbf{A}_2\}_{1,1} = -(\lambda_1\xi + \mu + \gamma);\ \{\mathbf{A}_2\}_{i,i} = -(\lambda_1\xi + \beta + \mu + \gamma);\ i = 2,3,...,F$

$\{\mathbf{A}_2\}_{i,i} = -(\lambda_1\xi + \beta + \mu),\ \ i = F+1, F+2, ..., K;\ \{\mathbf{A}_2\}_{i,i+1} = \beta;\ i = 1,2,...,K-1$

$$\{\mathbf{B}_2\}_{1,1} = -(\lambda_1\xi + \lambda_2 + \mu);\ \{\mathbf{B}_2\}_{i,i} = -(\lambda_1\xi + \lambda_2 + \beta + \mu);\ i = 2,3,...,K-1$$
$$\{\mathbf{B}_2\}_{K,K} = -(\lambda_1\xi + \lambda_2 + \beta);\ \{\mathbf{B}_2\}_{K+1,K+1} = -(\lambda_1\xi + \beta)$$
$$\{\mathbf{B}_2\}_{i,i+1} = \beta,\ i = 1,2,...,K-1;\ \{\mathbf{B}_2\}_{i+1,i} = \lambda_2,\ i = 1,2,...,K$$
$$\{\mathbf{A}_3\}_{1,1} = -(\mu + \gamma);\ \{\mathbf{A}_3\}_{i,i} = -(\beta + \mu + \gamma);\ i = 2,3,...,F$$
$$\{\mathbf{A}_3\}_{i,i} = -(\beta + \mu),\ i = F+1, F+2,...,K;\ \{\mathbf{A}_3\}_{i,i+1} = \beta;\ i = 1,2,...,K-1$$
$$\{\mathbf{B}_3\}_{1,1} = -(\lambda_2 + \mu);\ \{\mathbf{B}_3\}_{i,i} = -(\lambda_2 + \beta + \mu);\ i = 2,3,...,K-1$$
$$\{\mathbf{B}_3\}_{K,K} = -(\lambda_2 + \beta);\ \{\mathbf{B}_3\}_{K+1,K+1} = -\beta$$
$$\{\mathbf{B}_3\}_{i,i+1} = \beta,\ i = 1,2,...,K-1;\ \{\mathbf{B}_3\}_{i+1,i} = \lambda_2,\ i = 1,2,...,K$$
$$\{\mathbf{C}_0\}_{i+1,i} = \mu,\ i = 1,2,...,K-1;\ \{\mathbf{D}_0\}_{i+1,i} = \mu,\ i = 1,2,...,K-1$$
$$\{\mathbf{T}_0\}_{i,i} = \gamma,\ i = 1,2,...,F+1$$
$$\mathbf{L}_0 = \lambda_1\mathbf{I}_K;\ \mathbf{L}_1 = \lambda_1\mathbf{I}_{K+1};\ \mathbf{L}_2 = \lambda_1\xi\mathbf{I}_K;\ \mathbf{L}_3 = \lambda_1\xi\mathbf{I}_{K+1}$$

The matrices $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_0$, $\mathbf{L}_0$, $\mathbf{L}_2$ each of order $K \times K$, $\mathbf{B}_0$, $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{B}_3$, $\mathbf{D}_0$, $\mathbf{L}_1$, $\mathbf{L}_2$ each of order $(K+1)\times(K+1)$, and $\mathbf{T}_0$ of order $(K+1)\times K$.

The multi-equation, multi-variable, and multiple parameter queueing problem makes it highly challenging to calculate the closed-form expressions of the state probabilities; hence the repeated substitution approach is utilized to illustrate the steady-state probability distribution. The notion of embedded Markov chains was initially used in the matrix analytical approach by Neuts [143] in several realistic queue-based service systems. The probability vectors $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{Q}}_i$ are defined in this work to be used with the repeated substitution approach as follows:

$$\tilde{\mathbf{P}}_i = [P_{i,0}, P_{i,1}, P_{i,2}..., P_{i,K-1}];\quad i = 0,1,2,...,k \tag{4.1}$$

$$\tilde{\mathbf{Q}}_i = [Q_{i,0}, Q_{i,1}, ..., Q_{i,K-1}, Q_{i,K}];\quad i = 0,1,2,...,k \tag{4.2}$$

The complete probability vector of all system states is then calculated as

$$\mathbf{\Pi} = \left[\tilde{\mathbf{P}}_0, \tilde{\mathbf{Q}}_0, \tilde{\mathbf{P}}_1, \tilde{\mathbf{Q}}_1, \tilde{\mathbf{P}}_2, ..., \tilde{\mathbf{Q}}_{k-1}, \tilde{\mathbf{P}}_k, \tilde{\mathbf{Q}}_k\right] \tag{4.3}$$

Now, for the sake of solving $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{Q}}_i$ we use a subset of equations from $\mathbf{\Pi Q} = \mathbf{0}$ combined with the normalization equation $\mathbf{\Pi e} = 1$ where $\mathbf{e}$ is a column vector whose elements are all

equal to 1.
Consequently, we solve the following linear set:

$$\tilde{\mathbf{P}}_0\mathbf{A}_0 + \tilde{\mathbf{Q}}_0\mathbf{T}_0 + \tilde{\mathbf{P}}_1\mathbf{L}_0 = \mathbf{0} \tag{4.4}$$

$$\tilde{\mathbf{Q}}_0\mathbf{B}_0 + \tilde{\mathbf{Q}}_1\mathbf{L}_1 = \mathbf{0};\ 1 \leq k \leq R \tag{4.5}$$

$$\tilde{\mathbf{P}}_{i-1}\mathbf{C}_0 + \tilde{\mathbf{P}}_i\mathbf{A}_1 + \tilde{\mathbf{Q}}_i\mathbf{T}_0 + \tilde{\mathbf{P}}_{i+1}\mathbf{L}_0 = \mathbf{0};\ 1 \leq i \leq F \tag{4.6}$$

$$\tilde{\mathbf{Q}}_{i-1}\mathbf{D}_0 + \tilde{\mathbf{Q}}_i\mathbf{B}_1 + \tilde{\mathbf{Q}}_{i+1}\mathbf{L}_1 = \mathbf{0};\ 1 \leq i \leq F \tag{4.7}$$

$$\tilde{\mathbf{P}}_{i-1}\mathbf{C}_0 + \tilde{\mathbf{P}}_i\mathbf{A}_2 + \tilde{\mathbf{Q}}_i\mathbf{T}_0 + \tilde{\mathbf{P}}_{i+1}\mathbf{L}_2 = \mathbf{0};\ F+1 \leq i \leq k-1 \tag{4.8}$$

$$\tilde{\mathbf{Q}}_{i-1}\mathbf{D}_0 + \tilde{\mathbf{Q}}_i\mathbf{B}_2 + \tilde{\mathbf{Q}}_{i+1}\mathbf{L}_3 = \mathbf{0};\ F+1 \leq i \leq k-1 \tag{4.9}$$

$$\tilde{\mathbf{P}}_{k-1}\mathbf{C}_0 + \tilde{\mathbf{P}}_k\mathbf{A}_3 + \tilde{\mathbf{Q}}_k\mathbf{T}_0 = \mathbf{0} \tag{4.10}$$

$$\tilde{\mathbf{Q}}_{k-1}\mathbf{D}_0 + \tilde{\mathbf{Q}}_k\mathbf{B}_3 = \mathbf{0} \tag{4.11}$$

$$\sum_{i=0}^{k}(\tilde{\mathbf{P}}_i + \tilde{\mathbf{Q}}_i)\mathbf{e} = 1 \tag{4.12}$$

We now have the result of appropriate matrix manipulation and recursive substitution as

$$\tilde{\mathbf{P}}_1 = \left(\tilde{\mathbf{P}}_0\mathbf{A}_0 + \tilde{\mathbf{Q}}_0\mathbf{T}_0\right)\left\{-\mathbf{L}_0^{-1}\right\} \tag{4.13}$$

$$\tilde{\mathbf{Q}}_1 = \tilde{\mathbf{Q}}_0\mathbf{B}_0\left\{-\mathbf{L}_1^{-1}\right\} \tag{4.14}$$

$$\tilde{\mathbf{P}}_{i+1} = \begin{cases} \left(\tilde{\mathbf{P}}_{i-1}\mathbf{C}_0 + \tilde{\mathbf{P}}_i\mathbf{A}_1 + \tilde{\mathbf{Q}}_i\mathbf{T}_0\right)\left\{-\mathbf{L}_0^{-1}\right\}; 1 \leq i \leq F \\ \left(\tilde{\mathbf{P}}_{i-1}\mathbf{C}_0 + \tilde{\mathbf{P}}_i\mathbf{A}_2 + \tilde{\mathbf{Q}}_i\mathbf{T}_0\right)\left\{-\mathbf{L}_2^{-1}\right\}; F+1 \leq i \leq k-1 \end{cases} \tag{4.15}$$

$$\tilde{\mathbf{Q}}_{i+1} = \begin{cases} \left(\tilde{\mathbf{Q}}_{i-1}\mathbf{D}_0 + \tilde{\mathbf{Q}}_i\mathbf{B}_1\right)\left\{-\mathbf{L}_1^{-1}\right\}; 1 \leq i \leq F \\ \left(\tilde{\mathbf{Q}}_{i-1}\mathbf{D}_0 + \tilde{\mathbf{Q}}_i\mathbf{B}_2\right)\left\{-\mathbf{L}_3^{-1}\right\}; F+1 \leq i \leq k-1 \end{cases} \tag{4.16}$$

4.4 System Performance Measures

The acceptability of any model of the queueing problems can be best interpreted in terms of its system characteristics. Here, several indices viz. expected customers' count in the system, expected subordinate servers count in queue, waiting time , throughput, etc. are key performance measures of interest which are obtained in order to endorse the system's applicability. Various system indices are expressed in vector form as:

- Expected number of customers in the first stage

$$L_1 = \sum_{n_1=1}^{k} \sum_{n_2=0}^{K-1} n_1 (P_{n_1,n_2} + Q_{n_1,n_2}) + \sum_{n_1=1}^{k} n_1 Q_{n_1,K}$$
$$= \sum_{n_1=1}^{k} n_1 (\tilde{\mathbf{P}}_{n_1} \mathbf{e}_K + \tilde{\mathbf{Q}}_{n_1} \mathbf{e}_{K+1}) \tag{4.17}$$

where $\mathbf{e}_K$, and $\mathbf{e}_{K+1}$ are column vectors of dimensions K and $K+1$ respectively, consisting of all entries 1.

- Expected number of customers in the second stage

$$L_2 = \sum_{n_1=0}^{k} \sum_{n_2=1}^{K-1} n_2 (P_{n_1,n_2} + Q_{n_1,n_2}) + \sum_{n_1=0}^{k} K Q_{n_1,K}$$
$$= \sum_{n_1=0}^{k} \tilde{\mathbf{P}}_{n_1} \mathbf{u}_K + \sum_{n_1=0}^{k} \tilde{\mathbf{Q}}_{n_1} \mathbf{u}_{K+1} \tag{4.18}$$

where, $\mathbf{u}_K = [0,1,2,...,K-2,K-1]^T$ and $\mathbf{u}_{K+1} = [0,1,2,...,K-1,K]^T$ of dimensions K and $K+1$, respectively.

- Expected waiting time of customer in first stage

$$W_1 = \frac{L_1}{\lambda_{eff1}} \tag{4.19}$$

where λ_{eff1} is effective arrival rate for customers in the first stage

$$\lambda_{eff1} = \sum_{n_1=0}^{G-1} \sum_{n_2=0}^{K-1} \lambda_1 P_{n_1,n_2} + \sum_{n_1=G}^{k-1} \sum_{n_2=0}^{K-1} \lambda_1 \xi P_{n_1,n_2} + \sum_{n_1=0}^{G-1} \sum_{n_2=0}^{K} \lambda_1 Q_{n_1,n_2} + \sum_{n_1=G}^{k-1} \sum_{n_2=0}^{K} \lambda_1 \xi Q_{n_1,n_2}$$
$$= \sum_{n_1=0}^{G-1} \lambda_1 \tilde{\mathbf{P}}_{n_1} \mathbf{e}_K + \sum_{n_1=G}^{k-1} \lambda_1 \xi \tilde{\mathbf{P}}_{n_1} \mathbf{e}_K + \sum_{n_1=0}^{G-1} \lambda_1 \tilde{\mathbf{Q}}_{n_1} \mathbf{e}_{K+1} + \sum_{n_1=G}^{k-1} \lambda_1 \xi \tilde{\mathbf{Q}}_{n_1} \mathbf{e}_{K+1}$$

where $\mathbf{e}_K$ and $\mathbf{e}_{K+1}$ are defined earlier.

- Expected waiting time of customer in second queue

$$W_2 = \frac{L_2}{\lambda_{eff2}} \tag{4.20}$$

where λ_{eff2} is effective arrival rate for customers in the second stage

$$\begin{aligned}\lambda_{eff2} &= \sum_{n_1=0}^{k-1}\sum_{n_2=1}^{K-1}\mu P_{n_1,n_2} + \sum_{n_1=0}^{k}\sum_{n_2=1}^{K}\lambda_2 Q_{n_1,n_2} + \sum_{n_1=0}^{k-1}\sum_{n_2=1}^{K}\mu Q_{n_1,n_2} \\ &= \sum_{n_1=0}^{k-1}\mu\tilde{\mathbf{P}}_{n_1}\mathbf{e}_{l-1} + \sum_{n_1=0}^{k}\lambda_2\tilde{\mathbf{Q}}_{n_1}\mathbf{e}_l + \sum_{n_1=0}^{k-1}\mu\tilde{\mathbf{Q}}_{n_1}\mathbf{e}_i\end{aligned}$$

- Average balking rate

$$\begin{aligned}ABR &= \sum_{n_1=G}^{k-1}\sum_{n_2=0}^{K-1}\lambda_1\xi(P_{n_1,n_2}+Q_{n_1,n_2}) + \sum_{n_1=G}^{k-1}\lambda_1\xi Q_{n_1,K} \\ &= \lambda_1\xi\sum_{n_1=G}^{k-1}(\tilde{\mathbf{P}}_{n_1}\mathbf{e}_K + \tilde{\mathbf{Q}}_{n_1}\mathbf{e}_{K+1})\end{aligned} \tag{4.21}$$

- Throughput of the system

$$\begin{aligned}\tau_p &= \sum_{n_1=1}^{k}\sum_{n_2=0}^{K-1}(\mu+\beta)P_{n_1,n_2} + \sum_{n_1=0}^{k}\sum_{n_2=1}^{K-1}\beta P_{n_1,n_2} + \sum_{n_1=1}^{k}\sum_{n_2=0}^{K-2}(\mu+\beta)Q_{n_1,n_2} + \sum_{n_1=0}^{k}\sum_{n_2=1}^{K}\beta Q_{n_1,n_2} \\ &= \sum_{n_1=0}^{k}\beta\tilde{\mathbf{P}}_{n_1}\mathbf{e}_{l-1} + \sum_{n_1=1}^{k}(\mu+\beta)\tilde{\mathbf{P}}_{n_1}\mathbf{e}_K + \sum_{n_1=1}^{k}(\mu+\beta)\tilde{\mathbf{Q}}_{n_1}\mathbf{e}_m + \sum_{n_1=0}^{k}\beta\tilde{\mathbf{Q}}_{n_1}\mathbf{e}_l\end{aligned} \tag{4.22}$$

where, $\mathbf{e}_m = [1,1,...,1,0,0]^T$, $\mathbf{e}_{l-1} = [0,1,1,...,1,1]^T$ and $\mathbf{e}_l = [0,1,1,...,1,1]^T$ of dimensions K, $K-1$ and K, respectively.

- The probability that the server requires a startup time before starting the service

$$\begin{aligned}Ps &= \sum_{n_1=0}^{k}\sum_{n_2=0}^{F}P_{n_1,n_2} \\ &= \sum_{n_1=0}^{k}\tilde{\mathbf{P}}_{n_1}\mathbf{e}_F\end{aligned} \tag{4.23}$$

where, $\mathbf{e}_F = [1,1,...,1,0,0,...,0,0]^T$ of dimension K with first $F+1$ entries as 1 and remaining entries as 0.

- The probability that the server is blocked

$$Pb = \sum_{n_1=0}^{k} \sum_{n_2=0}^{K-1} P_{n_1,n_2}$$
$$= \sum_{n_1=0}^{k} \tilde{\mathbf{P}}_{n_1} \mathbf{e}_K \tag{4.24}$$

4.5 The Computation of Cost Function

In this section, the total cost function of two decision parameters, μ, and β, is formulated using various cost components for performance measures encountered in the system. Several components of the cost per unit time associated with distinct occurrences are used for this purpose. The cost components that are included in the cost function are:
$Ch_1 \equiv$ The holding cost incurred for each customer present in the first stage.
$Ch_2 \equiv$ The holding cost incurred for each customer waiting for the second stage service.
$C_1 \equiv$ The cost associated for providing service with rate μ.
$C_2 \equiv$ The cost associated for providing service with rate β.
$C_b \equiv$ The cost associated when server is blocked.
$C_k \equiv$ The cost associated for for system capacity K.
$C_w \equiv$ The cost associated for waiting time of customer in initial phase.
$C_s \equiv$ The cost associated for waiting time of customer in final phase.
Now, the cost function per unit time is constructed by combining the different cost aspects mentioned above with system performance indices such as-

$$TC(\mu,\beta) = Ch_1 \times L_1 + Ch_2 \times L_2 + C_1 \times \mu + C_2 \times \beta + C_b \times Pb_l + C_k \times K + C_w \times W_1 + C_s \times W_2; \tag{4.25}$$

The governing optimization problem is developed as

$$TC(\mu^*,\beta^*) = \min_{\mu,\beta}\{TC(\mu,\beta)\} \tag{4.26}$$

Embedding a meta-heuristic optimization algorithm on the total cost function is a powerful tool for cost-optimal from an economic analysis viewpoint. In this chapter, based on the decision parameters of the system, a total cost function for the proposed model is formulated as a multi-objective optimization problem considering the customer count in the initial and

final phases, service cost for both the servers, and waiting time of customers in the system. Then, a robust meta-heuristic optimization technique named Grasshopper Optimization Algorithm is employed to solve it to obtain the optimal total cost TC^* and deciding parameters (μ^*, β^*).

4.6 Grasshopper Optimization Algorithm

Inspiration

The fundamental concept in GOA is that larvae with limited mobility are utilized for local exploitation, adults with high mobility are used for global exploration, and the grasshopper's location is the optimal solution to solve the optimization problem.

Mathematical Model and Algorithm

The mathematical model for simulating the behavior of grasshopper swarms is as follows:

$$P_i = CO_i + GF_i + W_i \tag{4.27}$$

where P_i, CO_i, GF_i, and W_i denote the position, the community interaction, the gravity force, and the wind advection of the ith grasshopper, respectively. The grasshoppers are randomly distributed in the search space as search agents. So, Eq 4.27 is rewritten by considering their random behavior in the following manner:

$$P_i = r_1 SO_i + r_2 GF_i + r_3 W_i \tag{4.28}$$

where r_1, r_2, r_3 are the random numbers within $[0,1]$. The search component in GOA is calculated as

$$CO_i = \sum_{j=1, j\neq i}^{N} S(d_{ij})\hat{d_{ij}} \tag{4.29}$$

where N denotes the number of the grasshoppers in the swarm, $d_{ij} = \mid P_j - P_i \mid$ is the distance between the ith grasshopper and the jth grasshopper, $\hat{d_{ij}} = \frac{P_j - P_i}{d_{ij}}$ is the unit vector from the ith grasshopper to the jth grasshopper. A function to explain the power of social forces represented by $S(r)$ is defined as

$$S(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{4.30}$$

where f and l indicate the intensity of attraction and the attractive length scale with ranges $[0,1]$ and $[1,2]$, respectively. The function S divides space between two grasshoppers into three zones: the attraction zone, comfort zone, and repulsion zone. Distance between any two grasshoppers assumed to be between 1 and 4 since the force between two grasshoppers disappears if the distance between them is significant. The gravitational force GF_i in the Eq. 4.27 is defined by

$$GF_i = -g\hat{e}_g \tag{4.31}$$

where g is the gravitational constant and $\hat{e}_g$ presents a combination vector toward the middle of the surface. The wind force W_i in the Eq. 4.27 is defined by

$$W_i = u\hat{e}_w \tag{4.32}$$

where u is a constant drift and $\hat{e}_w$ is a combination vector toward the wind.

$$P_i = \sum_{j=1, j\neq i}^{N} S\left(\left|P_j - P_i\right|\right)\frac{P_j - P_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \tag{4.33}$$

In order to make the algorithm converge to a specific point and prevent grasshoppers from quickly reaching their comfort zone, the formula is improved to make it close to the optimal solution. The modified equation of Eq 4.33 is given by

$$P_i^d = \eta \sum_{j=1, j\neq i}^{N} \eta\frac{Ub_d - Lb_d}{2} S\left(\left|P_j^d - P_i^d\right|\right)\frac{P_j^d - P_i^d}{d_{ij}} + \hat{T}_d \tag{4.34}$$

where Ub_d, Lb_d are the higher bound and the smaller bound of the dth component of the ith grasshopper, T^d is the value in dth dimension of best agent or the optimal grasshopper T^*, the adaptive parameter η presents a decreasing coefficient to narrow the comfort zone, repulsion zone, and attraction zone. In Eq. 4.34, the gravity factor is set to 0. And assume that the wind direction is always towards a target T^d.

To balance the exploration stage and the exploitation, the parameter c is defined by

$$\eta = \eta_{\max} - t\frac{\eta_{\max} - \eta_{\min}}{T} \tag{4.35}$$

where η_{max} and η_{min} are the maximal and the minimal values of the parameter η, respectively. T represents the maximum iteration and t is the current iteration.

Algorithm 4 Pseudo code for Grasshopper Optimizer

Parameter Initialization: iteration, η_{max}, η_{min}, l, and f
Initialize the swarm of grasshoppers randomly $X_i, i = 1,2,3,...,n$
Evaluate the fitness value of each grasshopper
Select the best solution among all (best search agent)
while $t < t_{max}$ or convergence criterion **do**
    update η using Eqn. 4.35;
    for each grasshopper **do**
        Normalize distance between grasshopper in the range $[1,4]$
        Update present position of grasshopper according to Eq. 4.34;
        Fetch current grasshopper back if it goes outside limits;
    end for
    update the best solution if there's a better one
    itration= itration+1
end while
Output: Return the best optimum solution.

Table 4.1: Data set of parameters involved in presented model (Section 4.2) with sources

System Parameters	Numeric value	Source(s)
λ_1	20	[76]
λ_2	0.5	Assumed
μ	5	Assumed
β	4	Assumed
ξ	0.1	[12]
γ	0.4	[196]
F	5	Assumed
G	5	[76]
K	20	Assumed
Ch_1	25	[196]
Ch_2	30	[196]
C_1	100	[196]
C_2	150	[196]
C_b	100	[196]
C_k	200	Assumed
C_w	250	[113]
C_s	300	[113]

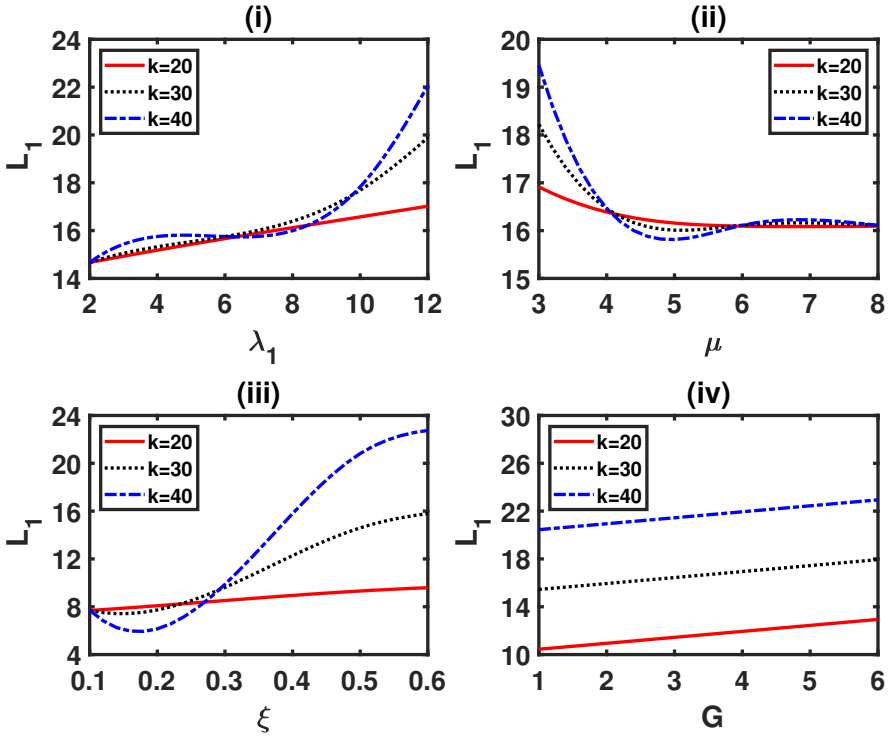


Figure 4.1: Effect of initial phase queue capacity k on queue length L_1 wrt (i) λ_1, (ii) μ, (iii) ξ, (iv) G. The parameters values are taken from Table 4.1.

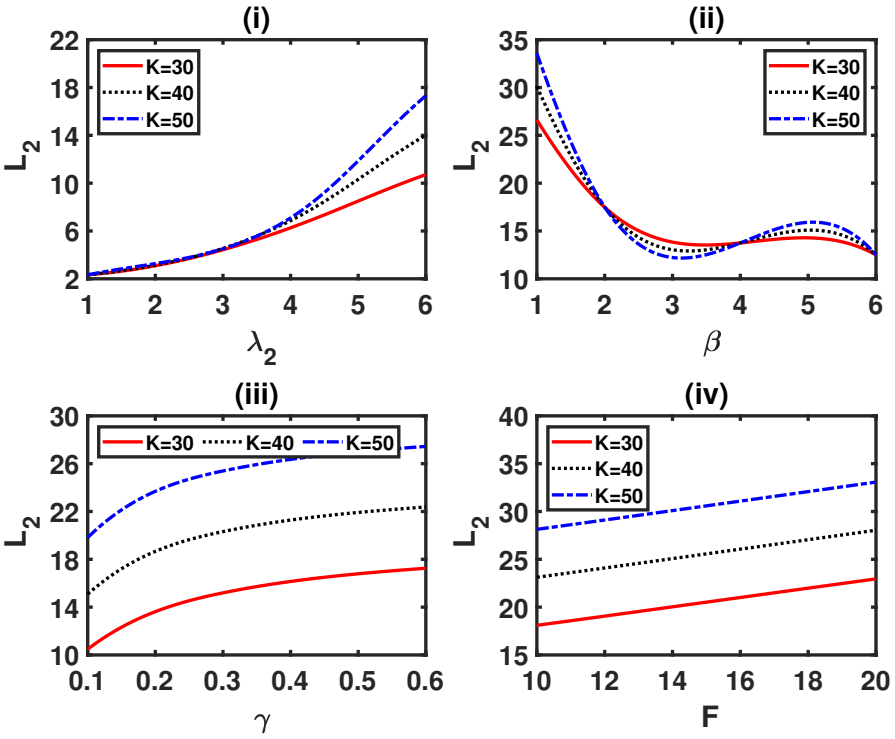


Figure 4.2: Effect of system capacity K on queue length L_2 wrt (i) λ_2, (ii) β, (iii) γ, (iv) F. The parameters values are taken from Table 4.1.

4.7 Results and Discussion

In this section, the developed model is used to simulate the performance measures of the system derived in Section 4.4 under various parameters. To illustrate the practicability of the presented model and theoretical analysis, all numerical results obtained were produced by implementing algorithms in MATLAB software (R2022b (9.9.0.1592791), 64-bit, Licence number 925317) on a system with configuration Intel(R), Xeon(R), CPU E3-1231 v3 @ 3.40 GHz with RAM 32.0 GB. The values of the default parameters are given in Table 4.1 to satisfy the requirements of the model described in Section 4.2.

Fig 4.1 shows variation in the length of the first phase queue for various determining system parameters. The mathematical expression for L_1 in Eqn. 4.17 emphasizes that accommodation constraint k for the initial phase is the determining factor of queue size for this phase. Henceforth, L_1 has experimented against system parameters for different values of k. It can be seen from Fig 4.1 that the estimated pattern follows the simulation results very closely. From Fig 4.1, we can observe that λ_1, μ, ξ and γ significantly affect L_1. Following conclusions can be induced from Fig 4.1. We can observe that λ_1, μ, ξ and γ significantly affect L_1 in fig 4.1. The following conclusions can be induced from Fig 4.1.

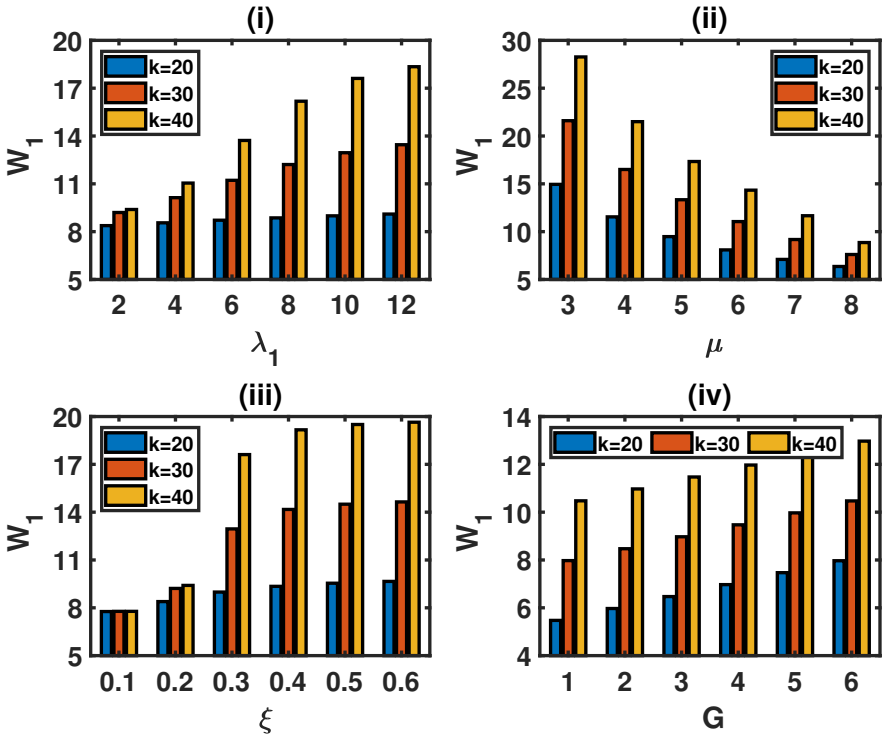


Figure 4.3: Expected waiting time of customers in the initial phase of the system (W_1) wrt (i) λ_1, (ii) μ, (iii) ξ, (iv) G for the parametric values given in Table 4.1.

(a) Arrival factor: Fig 4.1(i) depicts that as the arrival rate λ_1 increases, L_1 increases with a decreasing rate due to balking factor, but ultimately it increases on a further rise in λ_1 values. (b) Service factor: An increment in the service rate of the phase 1 server decreases the queue length to a certain value, but the pattern reverses on further increase in its service. The reason for such is that balking reduces with queue length. Further service improvement outperforms the balking factor and increases L_1, as shown in Fig 4.1(ii). (c) Impatience attribute: In Fig 4.1(iii), we illustrate the impact of abandonments on queue length using joining probability ξ. As expected, the customers count increases with joining probability, implying that balking discourages customers from joining the system. It exemplifies the idea that the probability of a customer loss increases with increasing consumer impatience. (d) Fig 4.1(iv) demonstrates that L_1 increases significantly when the initialization of balking phenomena is increased.

To examine the effect of λ_2, β, γ and F on L_2, we take the same default parameters values given in Table 4.1. The trend of the L_2 graph in Fig 4.2 (i, ii) can be expected by noting that the explanation is more or less the same as that of Fig 4.1 (i, ii) for their respective arrival and service rates. The state-dependent admissible control policy in phase 2 enhances the queue size in case of an increase in set-up time and threshold limit F. (see in Fig 4.2 (iii, iv)) From the Little's formula for waiting times in Eqns 4.19 and 4.20, W_1 and W_2 are in direct proportionate to L_1 and L_2, respectively. This can be easily seen in the pattern of W_1 and W_2 graphs (Fig 4.3, 4.4) for parameters λ_1, μ, ξ, and G following similar to that of L_1 (see Fig 4.1), whereas for parameters λ_2, β, γ, and F following similar to that of L_2(see Fig 4.2). The present numerical results are in good agreement with the practical implications and existing results for classical queueing models.

Throughput measures the rate of successful services per unit of time by all servers in the system. Thus, the performance of the system ultimately depends on the throughput. In order to boost the service quality and output of the system, the model is designed in such a way as to maximize the throughput. As shown in Fig 4.5(i, ii), it turns out that service rates resulting in throughput(τ_p) of the system increase accordingly. The control of arrivals and reduction in set-up time improves the efficiency of the server, as illustrated in Fig 4.5(iii, iv). Further, we investigate the effect of control policy parameters γ and F on probabilities of server blockage (P_b) and server requiring startup time before service to restart (P_s). The numerical results of P_s and P_b for various system capacities are depicted in Fig 4.6. The observations are as follows: (i) As the startup rate γ increases, the time taken for the server to resume the service is reduced, as shown in Fig 4.6(i). (ii) When the threshold limit F of the control policy is on the higher side, the server requires sufficient time to respond to a restart of service (see Fig 4.6(ii)). (iii) The probability of the server being blocked is reduced

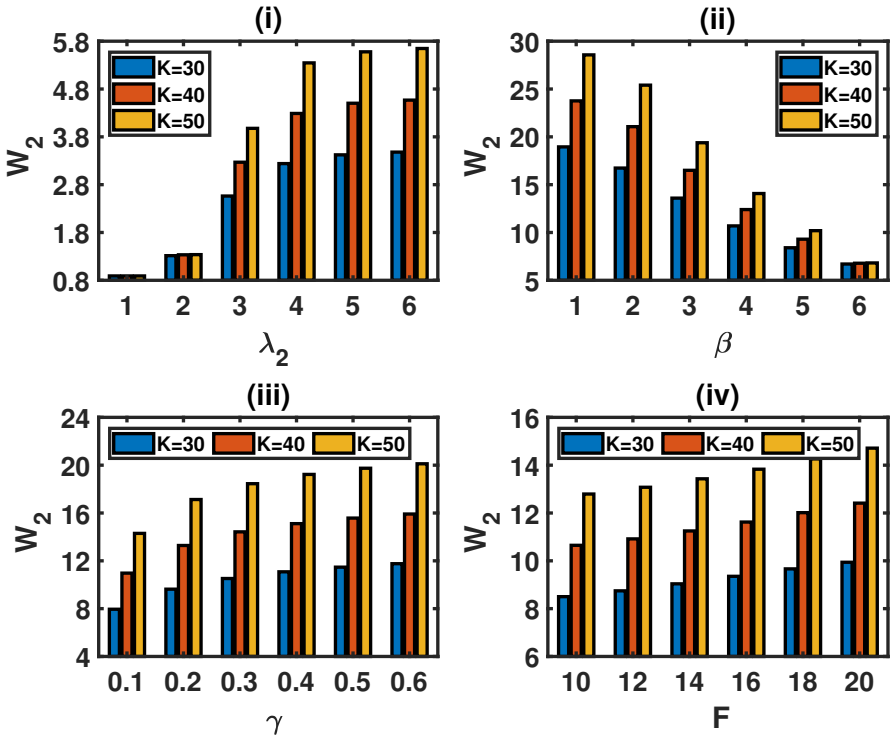


Figure 4.4: Expected waiting time of customers in the final phase of the system (W_2) wrt (i) λ_2, (ii) β, (iii) γ, (iv) F for the parametric values given in Table 4.1.

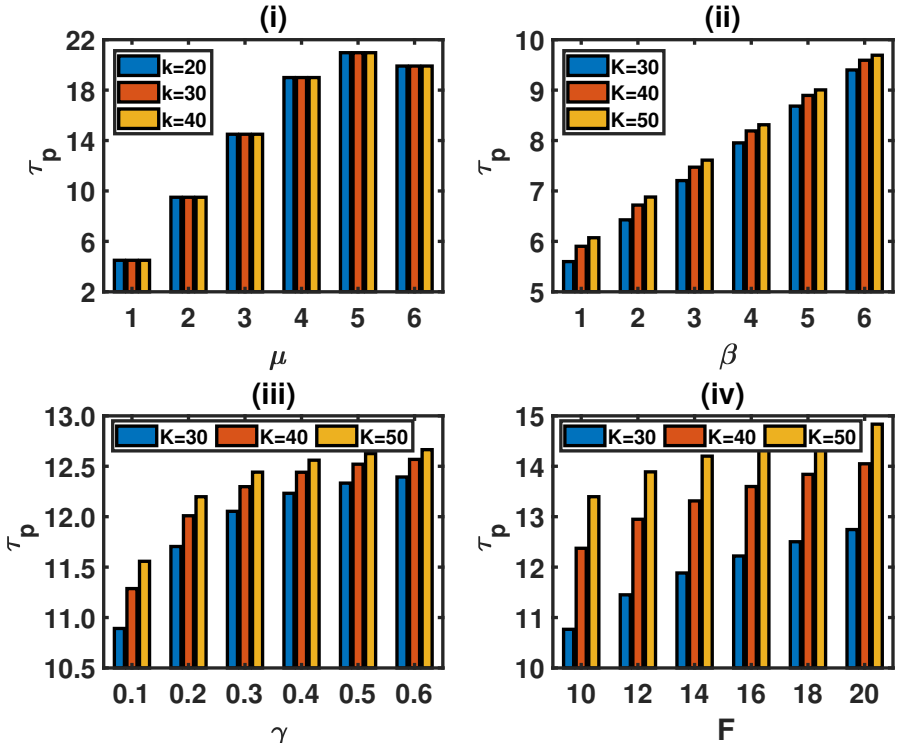


Figure 4.5: Throughput (τ_p) of the system wrt (i) μ and k, (ii) β and K, (iii) γ and K, (iv) F and K. The default set of paramters values are given in Table 4.1.

for servers that are quick in resuming the service (Fig 4.6 (iii), (iv)).

A numerical study of system performance measures allows us to obtain further insights into the effect of the decision parameters of the system on the total cost function. The deciding parameters are μ and β, which are the service rates of servers that are flexible and can be decided in the system. The problem arises in setting these service rates in order to minimize the cost incurred. In that direction, the total cost function is evaluated against μ and β in Fig 4.7(i, ii). This figure points out that the total cost function is convex with respect to both parameters. The convexity nature of TC for μ and β guarantees that TC is convex with the simultaneous effect of μ and β. The total cost is then evaluated numerically to obtain a convex surface plot and closed contour, as depicted in Fig 4.7(iii, iv).

4.7.1 Sensitivity Analysis

The primary goal of this chapter is to find the ideal point (μ^*, β^*) which gives rise to the optimal cost TC^*. This is accomplished by applying the GOA technique on the total cost objective function over some range of μ and β values obtained from the contour plot (Fig 4.7(iv)). The MATLAB software was utilized for coding GOA and carrying out the

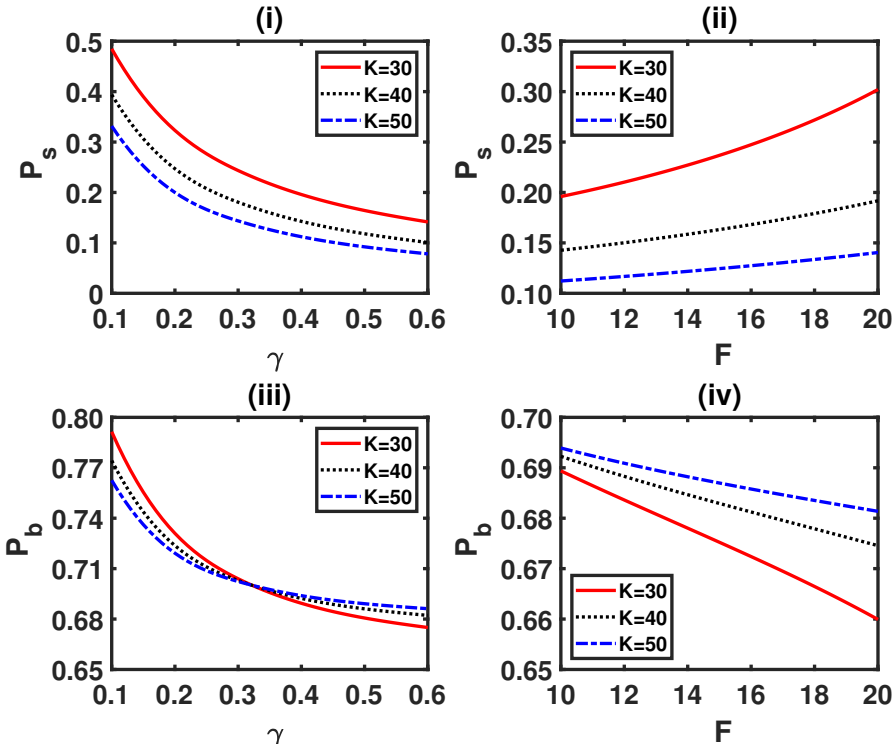


Figure 4.6: Variations in the values of probabilities P_s and P_b by varying system size K wrt (i, iii) γ, and (ii, iv) F for parametres values taken from Table 4.1.

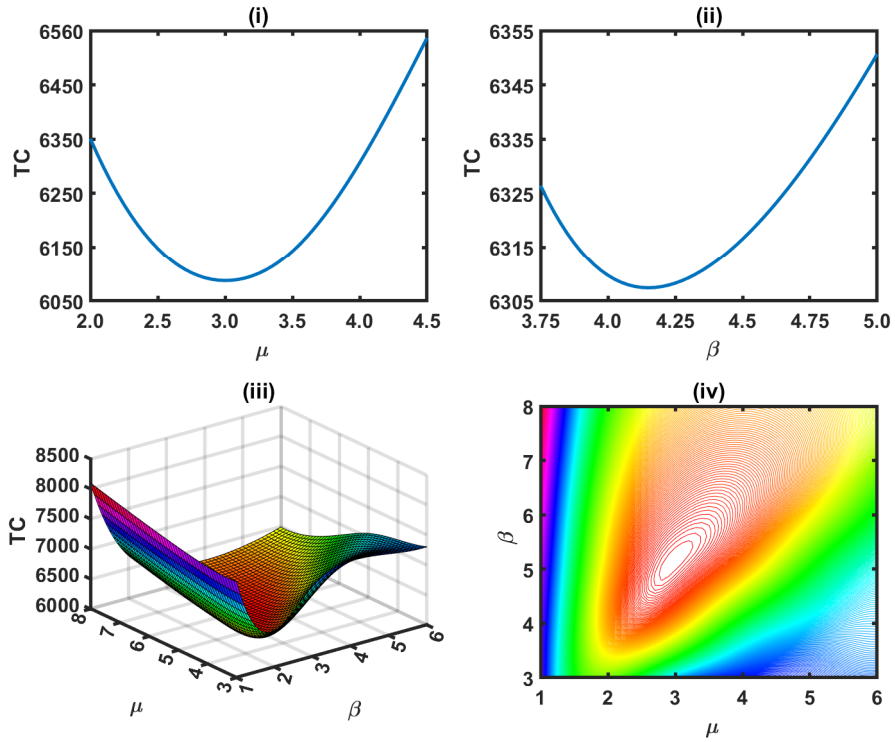


Figure 4.7: Plot of total cost (TC) wrt system's decision parameters (i) μ, (ii) β as well their cumulative effect in (iii) surface plot and (iv) contour plot of TC for parameters values given in Table 4.1.

optimization. For the algorithm's initialization, 30 populations are allotted randomly in the contour space. As the iterations go by, all try to converge to a single point inside the innermost contour which gives the optimal values for deciding parameters and total cost, as illustrated in Fig 4.8.

A total of 30 iterations are evaluated, and observed that the optimal point is reached after around 10 iterations only. Multiple runs of GOA are experimented on the total cost function to avoid misconceptions in the obtained optimal cost. The run graphs of these iterations for 10 runs are shown in Fig 4.9 for different TC ranges.

The results presented in Tables 4.2-4.3 show the effect of system parameters and cost elements on optimal total cost (TC^*) using the GOA algorithm and aid in supporting the estimation of the Mean and Maximum value of $\frac{TC_i}{TC^*}$. For instance, in the effect of system capacity in Table 4.2, it should be noted that, for increasing K, TC^* is increasing more rapidly while the optimal service rates converge to the same value $\mu^* = 3.123262$ and $\beta^* = 5.257844$ for large K. Optimal total cost TC^* is obtained for the parameter values set in Table 4.1, which give $TC^* = 6084.369292$ with $\text{Mean}\left(\frac{TC_i}{TC^*}\right) = 1.000000000156704$ and $\text{Max}\left(\frac{TC_i}{TC^*}\right) = 1.000000000162291$.

4.8 Conclusions and Managerial Insights

This chapter studies a queueing system combining two phases of service with arrival control policy. To tackle such queueing problem and strive towards an unclogged system, which contributes to economic savings through maximum utilization of service providers and queue management of system, desperate and proactive measures are deemed necessary. To this end, this chapter focuses on explicitly measuring the total cost of the model by considering some of the most determining performance measures of the system. The numerical simulation results in this chapter provide important insights into the complex interactions between the parameters and the critical performance measures of the system. The findings confirmed that the balking strategies of customers negatively impact the system's throughput, and the admission control policy, i.e., the F-policy, helps service providers reduce the congestion level. After all, the results of this study indicate the optimal service rates of servers to be kept in order to obtain the optimal cost for the multi-objective total cost optimization problem. In this regard, present study develops a queuing system to improve the management of service facilities. However, further work is required to establish this approach for the model with multi-server in each phase and determine the optimal number of servers. A natural extension of the presented model is to consider discount policy for the app user in order to promote online ordering that reduces congestion.

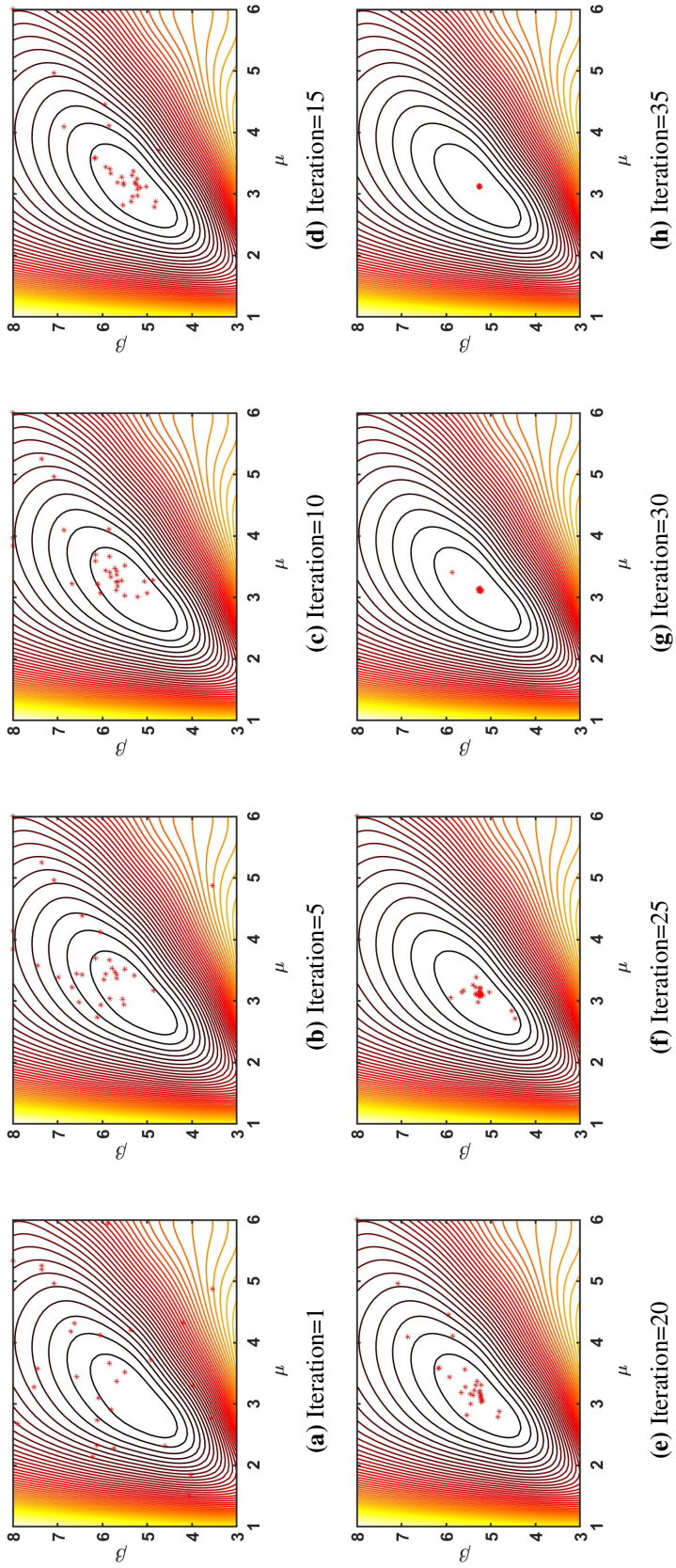


Figure 4.8: Several generations of GOA algorithm on the contour of $TC(\mu, \lambda_2)$

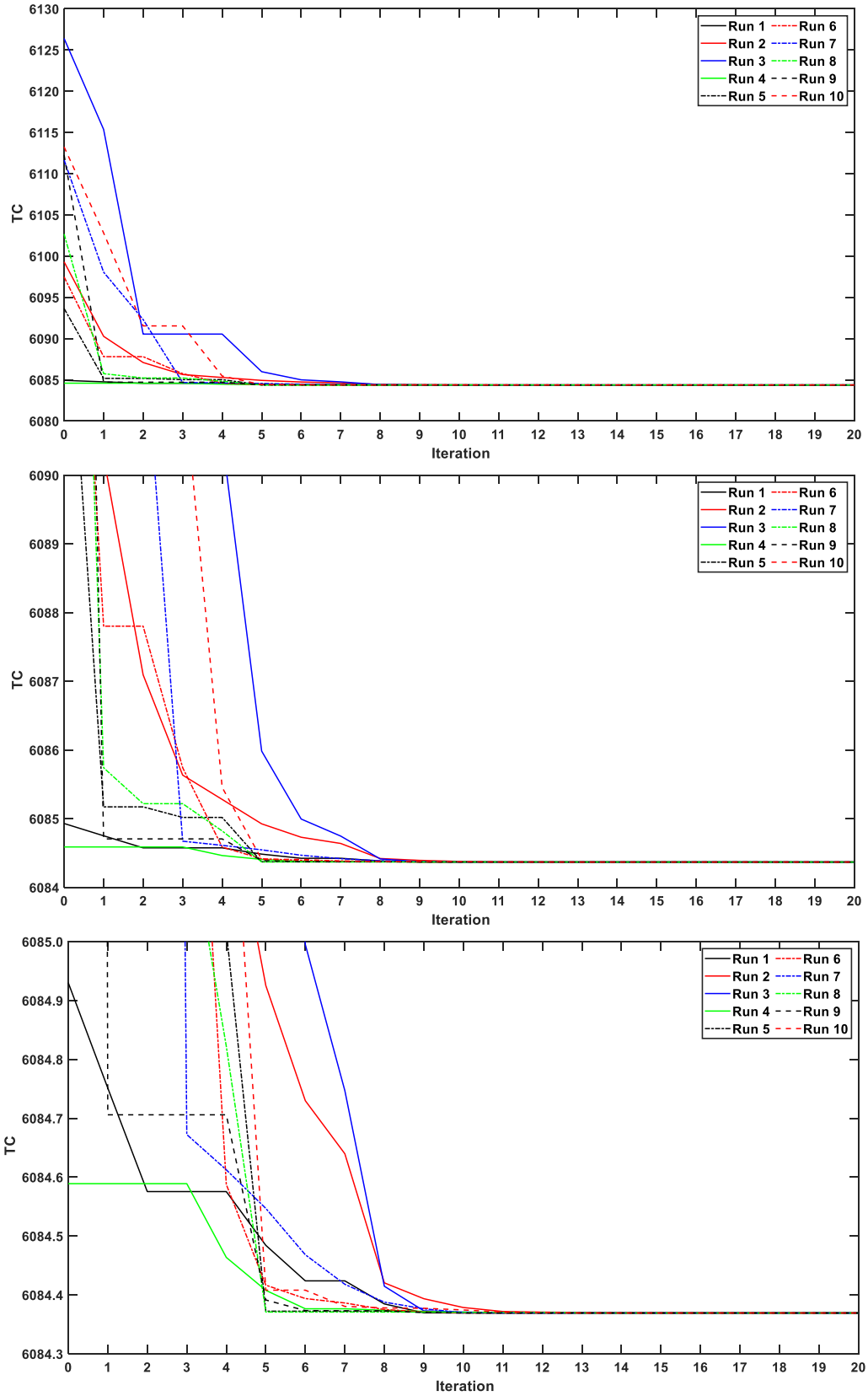


Figure 4.9: Convergence of iteration of GOA algorithm

Table 4.2: Optimal expected total cost of the system $TC^*(\mu^*,\beta^*)$ for different parameters via GOA algorithm. The default values of remaining system parameters are taken from 4.1.

$(K,F,G,k,\lambda_1,\xi,\lambda_2,\gamma)$	μ^*	β^*	$TC^*(\mu^*,\beta^*)$	**Mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**time elapsed**
10,5,5,10,20,0.1,0.5,0.4	3.27475	4.919208	4049.315062	5.576434	8.396480	532.724728
20,5,5,10,20,0.1,0.5,0.4	3.123262	5.257844	6084.369292	1.567040	1.622190	1181.584156
30,5,5,10,20,0.1,0.5,0.4	3.114569	5.278628	8085.768220	2.409447	3.407604	1876.184638
20,10,5,10,20,0.1,0.5,0.4	3.123205	5.258381	6084.389235	3.508576	4.223792	1186.328755
20,15,5,10,20,0.1,0.5,0.4	3.122838	5.259155	6084.436506	1.000000	1.000000	1225.374542
20,5,2,10,20,0.1,0.5,0.4	3.05478	5.132347	5785.812788	2.547497	7.386649	1222.679569
20,5,8,10,20,0.1,0.5,0.4	3.239739	5.467215	6359.089347	1.285653	1.653966	1130.408152
20,5,5,6,20,0.1,0.5,0.4	2.666153	5.06301	6038.839159	2.446765	5.438578	728.4653731
20,5,5,14,20,0.1,0.5,0.4	3.326657	5.450165	6108.136694	4.354760	6.546346	1664.120088
20,5,5,10,10,0.1,0.5,0.4	2.707924	4.809985	5942.063189	2.346590	5.296456	1188.744576
20,5,5,10,15,0.1,0.5,0.4	2.908567	5.025226	6016.059272	1.032759	2.486577	1189.063014
20,5,5,10,20,0.2,0.5,0.4	3.679508	5.927624	6308.007266	4.987636	6.902754	1161.990858
20,5,5,10,20,0.3,0.5,0.4	3.783528	6.215483	6462.737692	1.348765	3.846587	1165.61477
20,5,5,10,20,0.1,1.0,0.4	3.10463	5.735925	6156.548671	4.832965	9.765388	1167.027826
20,5,5,10,20,0.1,1.5,0.4	3.094492	6.205521	6229.606051	1.984655	5.896643	1163.551455
20,5,5,10,20,0.1,0.5,0.2	3.123168	5.257891	6084.381997	2.925769	4.376155	1167.489105
20,5,5,10,20,0.1,0.5,0.6	3.123273	5.257831	6084.365911	3.735476	6.530000	1181.433326

Table 4.3: Optimal expected total cost of the system $TC^*(\mu^*,\beta^*)$ for different parameters via GOA algorithm. The default values of remaining system parameters are taken from 4.1.

$(Ch_1,Ch_2,C_1,C_2,C_b,C_k,C_w,C_s)$	μ^*	β^*	$TC^*(\mu^*,\beta^*)$	**Mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**time elapsed**
15,30,100,150,100,200,250,300	3.087495	5.222036	6022.550106	2.847655	4.438643	1183.470546
25,30,100,150,100,200,250,300	3.123262	5.257844	6084.369292	1.567040	1.622190	1181.584156
35,30,100,150,100,200,250,300	3.158414	5.293089	6145.921485	1.367546	3.4368375	1181.845136
25,20,100,150,100,200,250,300	3.151388	5.204087	6061.712494	2.537956	5.7628543	1189.910272
25,40,100,150,100,200,250,300	3.097279	5.306904	6105.84409	2.4765926	9.783594	1205.385227
25,30,50,150,100,200,250,300	3.39560	5.530342	5921.744208	2.476539	7.344364	1170.395108
25,30,150,150,100,200,250,300	2.915952	5.050332	6235.140118	1.4876209	2.489792	1169.612776
25,30,100,100,100,200,250,300	3.400735	5.93600	5805.922097	1.589473	4.654838	1188.207507
25,30,100,200,100,200,250,300	2.918224	4.813091	6335.486636	9.376584	2.548762	1230.773628
25,30,100,150,50,200,250,300	3.123538	5.257211	6084.324469	1.000452	2.000946	1177.251662
25,30,100,150,150,200,250,300	3.122947	5.258423	6084.413959	3.874656	6.867460	1172.320863
25,30,100,150,100,150,250,300	3.123341	5.257931	5084.369292	3.857698	5.376854	1173.121553
25,30,100,150,100,250,250,300	3.123246	5.257828	7084.369291	1.004866	2.746598	1163.203630
25,30,100,150,100,200,200,300	2.90642	5.040325	5980.656770	1.007436	5.724658	1171.315191
25,30,100,150,100,200,300,300	3.318333	5.453763	6179.043817	3.094765	9.357650	1180.148793
25,30,100,150,100,200,250,200	3.072125	4.968389	5989.781999	1.374185	4.984365	1177.052863
25,30,100,150,100,200,250,400	3.169607	5.508111	6170.283532	1.004372	8.264876	1164.161962

Chapter 5

Quasi and Metaheuristic Optimization Approach for Service System with Strategic Policy and Unreliable Service

Demands for cost-efficient and just-in-time service systems have increased rapidly due to the nature of present-day competition. We focus on optimal policies for the highly efficient service system since the congestion of the customers more often originates from degraded policies rather than faulty arrangements. The quasi and metaheuristic optimization techniques have been widely used to establish cost-optimal service policies to diminish the congestion of customers, which arises mainly due to the phenomena of unplanned policies or caused by inadequate facilities. This chapter presents a notion of unreliable service and F-policy for stochastic modeling of a finite capacity customer service system.

5.1 Introduction

Demands for efficient service have increased vastly under the present-day just-in-time and competitive requirements. The study of state-of-the-art service systems is needed because the consequences of involved congestion often cause enormous economic and reputational losses. The queueing theory is a mathematical study of such congestion in a systematic manner.

Conventionally, many queueing systems were studied, assuming the service would never fail. It is a primary consideration to believe that service has been rendered successfully to the customer at the end of service time completion. Although these assumptions simplify the problem analysis, this may only sometimes be true in real-time queueing/service systems. In many real-life scenarios, a service rendered may be unsuccessful and need to be verified. Such a queueing system can be modeled as for crowded railway stations where announcements are made for the arrival and departure of trains, but passengers might not hear announcements due to heavy disturbance at the platform. Other applications include remotely rendered services that might not reach customers due to technical faults. At the checkpoint, sometimes, there may need to verify service quality or perfection before leaving the service facility, like bills, details, date, etc. In these examples, though service is rendered but does not reach the customer, such service is called an unreliable service. A queueing system with unreliable service is a more realistic representation of the service systems; thus, assuming unreliable service is more reasonable.

In literature, studies on unreliable services are seldom available (cf. [148], [149]). Sensitivity and optimal analysis were done by Shekhar et al. [171] for the expected total cost incurred and reliability characteristics for a machine repair problem of standby provisioning in a Markovian environment with unreliable service and vacation interruption. Esfeh [56] developed new mean waiting time formulations for diverse transit systems, including dial-a-ride service, feeder-trunk service, and single route with unreliable service.

In our day-to-day life, the formation of queues can be observed everywhere, either physically such as in shopping mall counters, vehicles in traffic jams, polling booth centers, railway reservation windows, or virtually like industry 4.0, cloud computing, the Internet of Things, and numerous other places. Queue formation eases the understanding of service patterns at a time but ultimately causes a delay in the service, a significant issue for customers and service facilitators. Despite challenges in the smooth functioning of the service system where long queues are formed, the arrivals should be controlled by implementing the strategic admission control policy. In the present study, we consider a controllable arrival queueing system with unreliable service under the F-policy, which is more practical when

dealing with real-time congestion problems. These issues represent a research gap in the literature.

Analytical solutions for computing steady-state queue-size distribution for the F-policy Markovian $M/M/1/K$ queueing system with an exponential startup time were first proposed by Gupta [71]. According to the F-policy, the customers are allowed to enter the system for service until the number of customers outreaches the system's capacity K, and no more customers are admitted to join the queue until the number of customers ceases to rest up to a predefined level F. The server takes a random startup time before allowing customers to enter the system. F-policy deals with the case of controlling arrivals in a queueing system to avoid an overload, long delay, or congestion situation at a slight loss of revenue. Wang et al. addressed the optimal control of the F-policy for $G/M/1/K$ and $M/G/1/K$ queueing systems in [196] and [195], respectively. Yang et al. [217] analyzed an $M/M/2/K$ queueing system with F-policy and heterogeneous servers. Recently, Rani et al. [158] modeled Markovian queueing with reboot, recovery, and server vacationing under the F-policy to explore the performance of fault-tolerant systems. Wu et al. [207], [205] considered an F-policy queue with alternating service rates and formulated a bi-objective model using expected costs and waiting times along with the trade-off between operating costs and service quality.

Yang et al. [213] obtained the stationary distribution of the system size using the supplementary variable technique recursively for randomized control of arrivals in a finite-capacity single-server $GI/M/1$ system with starting failures. Shekhar et al. [169] investigated a randomized arrival control policy for impatient prospective customers in the finite queueing system with working vacation interruption. Jain et al. [93] presented a state-of-the-art literature survey on the F-policy for limited capacity and finite population state-dependent Markovian and non-Markovian queueing models. Jain et al. [92] used the remaining retrial time as the supplementary variable to frame the governing equations for the finite capacity state-dependent queueing model with F-policy and general retrial attempts and obtained the solution using the Laplace-Stieltjes transform and recursive method. Yang et al. [209] used the OptQuest tool in ARENA, a simulation software, for extensive computational experiments to find the optimal threshold F that minimizes the expected cost per unit time.

The primary purpose of this study is to preview the practical implementation of optimal cost practice of service systems. Due to the prevailing global economic crises, optimal cost measures have become the prioritized topic in the current strategic management practices. Optimal cost measures address questions about efficient and effective leadership and are explored in detail using different optimization techniques (cf. [216], [11]).

This manuscript focuses on determining the value of the governing parameter(s) of studied models so that, the incurred expected cost is minimum. Ford et al. [59] generalize the standard secant equations by considering a path defined by a polynomial and a gradient vector approximation with a polynomial interpolant. Kao et al. [97] introduced a modified version of the quasi-Newton method, where parameters are determined from some numerical experimentation. A quasi-Newton method is an advanced tool, and many updates have enriched it in recent years (cf. [226], [203], [78]). Due to computational richness, queueing theorists also used the quasi-Newton method for determining optimal decision parameter(s) for the studied constrained and non-constrained queueing models. Wang et al. [199] used the direct search method and the quasi-Newton method to find the global minimum (F^*, μ^*, γ^*) for the control policy of a removable and unreliable server for finite capacity single server Markovian queueing system where the removable server operates an F-policy. Some more significant contributions for optimal analysis of queueing models employing the quasi-Newton approach in the literature exist (cf. [215], [101]). The main disadvantage of the direct-search and quasi-Newton methods is their strict discrete and continuous domain.

To our knowledge, a finite capacity $M/M/1$ queueing model under F-policy with unreliable service has never been discussed in the literature or explored economically. The research gap motivates us to develop more practical queueing models considering unreliable service and controllable arrival processes. The purpose of this investigation is to accomplish three objectives. The first is to present the mathematical model of a state-of-the-art finite controllable arrival single server Markovian queueing system with unreliable service. The second is to offer applications of the efficient GWO metaheuristics technique for optimizing congestion problems. The third is to present extensive numerical results with an exhaustive parametric investigation for decision-makers. This queueing system has potential applications in wireless communication networks, vehicular traffic flow, voice, and data networks. This model can be extended in the future by incorporating linguistic uncertainty and discouragement in arrivals, as discussed in [98], [154].

The remaining chapter is coordinated in the following manner: Section 5.2 outlines the model description of the studied unreliable service mechanism and admission control F-policy for finite capacity single-server queueing system. In Section 5.3, we represent the studied queueing system in the closed-form block matrices and delineate the solution algorithm to obtain the steady-state probabilities in vector form in Subsection 5.3.1. Section 5.4 highlights various governing system performance measures. In Section 5.5, a cost function is formulated to determine the optimal values of several decision parameters at a minimal expected total cost. The description and overview of the grey wolf optimization technique, used for obtaining optimality of the studied model are briefed in Section 5.6. The special

cases studied in the past of the present model are discussed in Section 5.7. Some numerical results are provided in tabular and graphical form to illustrate the optimal analysis and simulations of various system performance measures in Section 5.8. Lastly, Section 5.9 gives concluding comments, contributions, and offer a future perspective.

5.2 Model Description

We examine a F-policy reliable single server $M/M/1$ Markovian queueing system with unreliable service and exponential startup time. The primary assumptions and notations for the studied model are characterized as follows.

Arrival Process

- The prospective customers arrive for intended service according to a Poisson process with arrival rate λ.
- For admission control to avoid long queues in waiting, if the queue size reaches a threshold $K\,(K<\infty)$, then no prospective customer is permitted to join the queue until the queue size diminishes a pre-specified threshold value $F\,(1\leq F\leq K-1)$.
- When customers are permissible to join, the service provider takes startup time, which follows an exponential distribution with a mean time of $1/\gamma$.
- Since the studied model is finite, the customer beyond the threshold K is assumed to be a lost customer.

Service Process

- A reliable server provides the service following the queue discipline of First Come First Served.
- The service times of customers are independent and identically distributed random variables that follow an exponential distribution with service rate μ.
- At the completion epoch of service, the customer assesses the nature of service as reliable or unreliable. This delay includes the time to check bills, descriptions, details, dates, quality, etc.
- If the customers receive reliable service, they leave the system with random time-to-leave, which follows an exponential distribution with the mean rate β_1; otherwise,

they remain in the system for a random period that follows an exponential distribution with a mean rate β_2.

The arrival and service processes are independent, i.e., all the events like arrival, service, reliable service, unreliable service, startup, customer allowed, or customer not allowed are statistically independent of each other.

For the stochastic modeling of the studied queueing problem, we have also used the following notations to describe the different states at any instant t.

$$N(t) \equiv \text{System size of the customers in the system at time} \, t$$
$$J(t) \equiv \text{The state of the server at time} \, t$$

where

$$J(t) = \begin{cases} 0 & ; \text{the customer is not permissible to enter the service system and customer immediately after service is rendered} \\ 1 & ; \text{the customer is not permissible to enter the service system and the server is busy} \\ 2 & ; \text{the customer is permissible to enter the service system, and the server is busy.} \\ 3 & ; \text{the customer is permissible to enter the service system and customer immediately after service is rendered} \end{cases}$$

Then, $(N(t), J(t); t \geq 0)$ is a continuous time Markov chain (CTMC) on the state space Ω

$$\Omega = \{(n,j) \mid n = 1,2,3,...,K-1,K \; ; \; j = 0\} \cup \{(n,j) \mid n = 0,1,2,...,K-1,K \; ; \; j = 1\} \cup$$
$$\{(n,j) \mid n = 0,1,2,...,K-2,K-1 \; ; \; j = 2\} \cup \{(n,j) \mid n = 1,2,...,K-2,K-1 \; ; \; j = 3\}$$

As $t \to \infty$, the system tends to stable condition. The governing steady-state probabilities are denoted as follows.

$$\pi_{n,0} = \lim_{t\to\infty} \text{Prob}\{N(t) = n, J(t) = 0\}; \; n = 1,2,3,...,K-1,K$$
$$\pi_{n,1} = \lim_{t\to\infty} \text{Prob}\{N(t) = n, J(t) = 1\}; \; n = 0,1,2,3,...,K-1,K$$
$$\pi_{n,2} = \lim_{t\to\infty} \text{Prob}\{N(t) = n, J(t) = 2\}; \; n = 0,1,2,3,...,K-2,K-1$$
$$\pi_{n,3} = \lim_{t\to\infty} \text{Prob}\{N(t) = n, J(t) = 3\}; \; n = 1,2,3,...,K-2,K-1$$

The Chapman-Kolmogrove forward system of linear equations for the studied F-policy $M/M/1$ queueing system with unpredictable unreliable service and random startup time, in terms of inflow-outflow rates and state probabilities, are as follows.

$$-(\beta_1 + \beta_2 + \gamma)\pi_{n,0} + \mu\pi_{n,1} = 0; \; 1 \leq n \leq F \tag{5.1}$$
$$-(\beta_1 + \beta_2)\pi_{n,0} + \mu\pi_{n,1} = 0; \; F+1 \leq n \leq K-1 \tag{5.2}$$
$$-(\beta_1 + \beta_2)\pi_{K,0} + \mu\pi_{K,1} + \lambda\pi_{K-1,3} = 0 \tag{5.3}$$
$$-\gamma\pi_{0,1} + \beta_1\pi_{1,0} = 0 \tag{5.4}$$

$$-(\mu+\gamma)\pi_{n,1}+\beta_2\pi_{n,0}+\beta_1\pi_{n+1,0}=0;\ 1\le n\le F \tag{5.5}$$

$$-\mu\pi_{n,1}+\beta_2\pi_{n,0}+\beta_1\pi_{n+1,0}=0;\ F+1\le n\le K-1 \tag{5.6}$$

$$-\mu\pi_{K,1}+\beta_2\pi_{K,0}+\lambda\pi_{K-1,2}=0 \tag{5.7}$$

$$-\lambda\pi_{0,2}+\gamma\pi_{0,1}+\beta_1\pi_{1,3}=0 \tag{5.8}$$

$$-(\lambda+\mu)\pi_{n,2}+\lambda\pi_{n-1,2}+\beta_2\pi_{n,3}+\beta_1\pi_{n+1,3}+\gamma\pi_{n,1}=0;\ 1\le n\le F \tag{5.9}$$

$$-(\lambda+\mu)\pi_{n,2}+\lambda\pi_{n-1,2}+\beta_2\pi_{n,3}+\beta_1\pi_{n+1,3}=0;\ F+1\le n\le K-2 \tag{5.10}$$

$$-(\lambda+\mu)\pi_{K-1,2}+\lambda\pi_{K-2,2}+\beta_2\pi_{K-1,3}=0 \tag{5.11}$$

$$-(\beta_1+\beta_2+\lambda)\pi_{1,3}+\gamma\pi_{1,0}+\mu\pi_{1,2}=0 \tag{5.12}$$

$$-(\beta_1+\beta_2+\lambda)\pi_{n,3}+\lambda\pi_{n-1,3}+\gamma\pi_{n,0}+\mu\pi_{n,2}=0;\ 2\le n\le F \tag{5.13}$$

$$-(\beta_1+\beta_2+\lambda)\pi_{n,3}+\lambda\pi_{n-1,3}+\mu\pi_{n,2}=0;\ F+1\le n\le K-1 \tag{5.14}$$

Following the law of total probability, the normalizing condition for state probabilities is given below.

$$\sum_{n=1}^{K}\pi_{n,0}+\sum_{n=0}^{K}\pi_{n,1}+\sum_{n=0}^{K-1}\pi_{n,2}+\sum_{n=1}^{K-1}\pi_{n,3}=1 \tag{5.15}$$

Practical Justification of the Model

There exist numerous real-life practical instances that illustrate both notions of unreliable service and admission control F-policy. One example of a F-policy with unreliable service in healthcare is the management of emergency departments. In a busy emergency department, limited resources, such as available beds or staff, may lead to delays and overcrowding. To manage this, an F-policy can be implemented, where patients are only allowed to enter the department if the number of patients currently waiting is below a certain threshold (K). Additionally, a patient receiving unreliable services, such as a misdiagnosis or improper treatment, may result in a longer stay or readmission. By incorporating the concept of unreliable service into the F-policy, healthcare providers can better manage patient flow and ensure quality care. This policy is used in real-life situations like popular dining, where tokens are given for table numbers, and orders are received after the announcement of the table number. This policy also considers unreliable service, where customers may not receive the intended service due to technical faults, leading to system overcrowding and unsuccessful service. Another application is computer networking systems, where messages are transmitted as data between processors. In the event of technical issues, the transmitted data may not be received despite being sent. This example is elaborated more in [196]. Overall, the F-policy is a commonly used technique to manage queueing systems in various industries, including telecommunication, transportation, and healthcare, where it is necessary to ensure that the waiting times are reasonable and the system remains stable. In this chapter, we have

presented the mathematical modeling of the model mentioned above. However, industrial validation and simulation remain future endeavors.

5.3 Matrix Analytic Solutions

The matrix-analytic method is employed to compute the steady-state probabilities for the studied F-policy $M/M/1/K$ queueing system with unreliable service and exponential startup time. The transition rate matrix $\mathbf{Q}$ of the Markov chain model of interest has a tridiagonal block structure, as follows.

$$\mathbf{Q} = \begin{bmatrix}
\mathbf{X}_0 & \mathbf{Z}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
\mathbf{Y}_0 & \mathbf{X}_1 & \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_1 & \mathbf{Z}_1 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_1 & \mathbf{Z}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_2 & \mathbf{Z}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_2 & \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_2 & \mathbf{Z}_1 & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Y}_1 & \mathbf{X}_2 & \mathbf{Z}_2 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Y}_2 & \mathbf{X}_3
\end{bmatrix}$$

The block tridiagonal matrix $\mathbf{Q}$ is a square matrix of order $4K$. The principal diagonal block entries $\mathbf{X}_0$, $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ are square matrices of order 2, 4, 4, and 2 respectively. The first diagonal below block entries $\mathbf{Y}_0$, $\mathbf{Y}_1$, and $\mathbf{Y}_2$ are of order 4×2, 4×4, and 2×4 respectively. The first diagonal above block entries $\mathbf{Z}_0$, $\mathbf{Z}_1$, and $\mathbf{Z}_2$ are of order 2×4, 4×4, and 4×2 respectively. All elements in the matrix form of transition matrix $\mathbf{Q}$ are defined as follows.

$$\mathbf{X}_0 = \begin{bmatrix} -\gamma & \gamma \\ 0 & -\lambda \end{bmatrix}, \quad \mathbf{X}_1 = \begin{bmatrix} -(\gamma+\beta_1+\beta_2) & \beta_2 & 0 & \gamma \\ \mu & -(\gamma+\mu) & \gamma & 0 \\ 0 & 0 & -(\mu+\lambda) & \mu \\ 0 & 0 & \beta_2 & -(\lambda+\beta_1+\beta_2) \end{bmatrix},$$

$$\mathbf{X}_2 = \begin{bmatrix} -(\beta_1+\beta_2) & \beta_2 & 0 & 0 \\ \mu & -\mu & 0 & 0 \\ 0 & 0 & -(\mu+\lambda) & \mu \\ 0 & 0 & \beta_2 & -(\lambda+\beta_1+\beta_2) \end{bmatrix}, \quad \mathbf{X}_3 = \begin{bmatrix} -(\beta_1+\beta_2) & \beta_2 \\ \mu & -\mu \end{bmatrix},$$

$$\mathbf{Y}_0 = \begin{bmatrix} \beta_1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \beta_1 \end{bmatrix}, \quad \mathbf{Y}_1 = \begin{bmatrix} 0 & \beta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \beta_1 & 0 \end{bmatrix}, \quad \mathbf{Y}_2 = \begin{bmatrix} 0 & \beta_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

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

The vector $\mathbf{\Pi}$, steady-state probabilities, is partitioned as $(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_{K-1}, \mathbf{P}_K)$. The sub-vectors $\mathbf{P}_0 = \{\pi_{0,1}, \pi_{0,2}\}$, $\mathbf{P}_n = \{\pi_{n,0}, \pi_{n,1}, \pi_{n,2}, \pi_{n,3}\}; 1 \leq n \leq K-1$, and $\mathbf{P}_K = \{\pi_{K,0}, \pi_{K,1}\}$ have dimensions of two, four, and two respectively. Due to the complexity involved in the studied model, the analytical solution for state probabilities is not feasible. For determining the numerical value of state probabilities, we merely solve the following problem.

$$\mathbf{\Pi Q} = \mathbf{0} \tag{5.16}$$

with the initial condition 5.15

$$\mathbf{\Pi e} = 1 \tag{5.17}$$

where $\mathbf{e}$ is column vector of ones of order $4K$. Therefore, the governing system of equations in matrix form in 5.16 are manifested for computing state probabilities as follows.

$$\mathbf{P}_0\mathbf{X}_0 + \mathbf{P}_1\mathbf{Y}_0 = \mathbf{0} \tag{5.18}$$

$$\mathbf{P}_0\mathbf{Z}_0 + \mathbf{P}_1\mathbf{X}_1 + \mathbf{P}_2\mathbf{Y}_1 = \mathbf{0} \tag{5.19}$$

$$\mathbf{P}_{n-1}\mathbf{Z}_1 + \mathbf{P}_n\mathbf{X}_1 + \mathbf{P}_{n+1}\mathbf{Y}_1 = \mathbf{0};\ 2 \leq n \leq F \tag{5.20}$$

$$\mathbf{P}_{n-1}\mathbf{Z}_1 + \mathbf{P}_n\mathbf{X}_2 + \mathbf{P}_{n+1}\mathbf{Y}_1 = \mathbf{0};\ F+1 \leq n \leq K-2 \tag{5.21}$$

$$\mathbf{P}_{K-2}\mathbf{Z}_1 + \mathbf{P}_{K-1}\mathbf{X}_2 + \mathbf{P}_K\mathbf{Y}_2 = \mathbf{0} \tag{5.22}$$

$$\mathbf{P}_{K-1}\mathbf{Z}_2 + \mathbf{P}_K\mathbf{X}_3 = \mathbf{0} \tag{5.23}$$

5.3.1 State Probabilities

We get the following solution using basic matrix manipulation to derive the state probabilities in the vector form. Since, matrix $\mathbf{X}_0$ is non-singular, Eqn. 5.18 gives

$$\mathbf{P}_0 = \mathbf{P}_1\mathbf{V}_0, \text{ where } \mathbf{V}_0 = -\mathbf{Y}_0\mathbf{X}_0^{-1} \tag{5.24}$$

From Eqns. 5.19 and 5.24, we have following result

$$\mathbf{P}_1 = \mathbf{P}_2\mathbf{V}_1, \text{where } \mathbf{V}_1 = -\mathbf{Y}_1(\mathbf{V}_0\mathbf{Z}_0 + \mathbf{X}_1)^{-1} \tag{5.25}$$

Using Eqns. 5.20 and 5.25, we have following iterative result

$$\mathbf{P}_n = \mathbf{P}_{n+1}\mathbf{V}_n, \text{where } \mathbf{V}_n = -\mathbf{Y}_1(\mathbf{V}_{n-1}\mathbf{Z}_1 + \mathbf{X}_1)^{-1};\ 2 \leq n \leq F \tag{5.26}$$

A similar result for different states can be derived using Eqns. 5.21 and 5.26 as follows

$$\mathbf{P}_n = \mathbf{P}_{n+1}\mathbf{V}_n, \text{where } \mathbf{V}_n = -\mathbf{Y}_1(\mathbf{V}_{n-1}\mathbf{Z}_1 + \mathbf{X}_2)^{-1};\ F+1 \le n \le K-2 \tag{5.27}$$

Using Eqns. 5.22 and 5.27, we get the following result

$$\mathbf{P}_{K-1} = \mathbf{P}_K\mathbf{V}_{K-1}, \text{where } \mathbf{V}_{K-1} = -\mathbf{Y}_2(\mathbf{V}_{K-2}\mathbf{Z}_1 + \mathbf{X}_2)^{-1} \tag{5.28}$$

Hence, computing Eqns. 5.24 - 5.28 in recursive manner, the state probabilities $\mathbf{P}_n$; $0 \le n \le K-1$ can be expressed in terms of state probability $\mathbf{P}_K$ as follow.

$$\mathbf{P}_n = \mathbf{P}_{n+1}\mathbf{V}_n = \mathbf{P}_{n+2}\mathbf{V}_n\mathbf{V}_{n+1} = \ldots = \mathbf{P}_K \prod_{\zeta=1}^{K-n} \mathbf{V}_{K-\zeta} = \mathbf{P}_K\boldsymbol{\Psi}_n^* \tag{5.29}$$

where $\boldsymbol{\Psi}_n^* = \prod_{\zeta=1}^{K-n}\mathbf{V}_{K-\zeta}$ and $\mathbf{V}_n$; $0 \le n \le K-1$ are given above in Eqns. 5.24 - 5.28. By the normalizing condition $\mathbf{Pe} = 1$ and Eqn. 5.29, we get

$$\begin{aligned}\mathbf{P}_0\mathbf{e}_2 + \sum_{n=1}^{K-1}\mathbf{P}_n\mathbf{e}_1 + \mathbf{P}_K\mathbf{e}_2 &= \mathbf{P}_0\mathbf{e}_2 + [\mathbf{P}_1 + \mathbf{P}_2 + \ldots + \mathbf{P}_{K-1}]\,\mathbf{e}_1 + \mathbf{P}_K\mathbf{e}_2 \\ &= \mathbf{P}_K\boldsymbol{\Psi}_0^*\mathbf{e}_2 + \left[\mathbf{P}_K\boldsymbol{\Psi}_1^* + \mathbf{P}_K\boldsymbol{\Psi}_2^* + \ldots + \mathbf{P}_K\boldsymbol{\Psi}_{K-1}^*\right]\mathbf{e}_1 + \mathbf{P}_K\mathbf{e}_2 \\ &= \mathbf{P}_K\left[\boldsymbol{\Psi}_0^*\mathbf{e}_2 + \sum_{n=1}^{K-1}\boldsymbol{\Psi}_n^*\mathbf{e}_1 + \mathbf{e}_2\right] = 1\end{aligned} \tag{5.30}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are column vectors defined as $\mathbf{e}_1 = [1,1,1,1]^T$, $\mathbf{e}_2 = [1,1]^T$. Hence, Eqn. 5.23 can be written as

$$\mathbf{P}_K\left[\mathbf{V}_{K-1}\mathbf{Z}_2 + \mathbf{X}_3\right] = \mathbf{0} \tag{5.31}$$

Therefore, on solving Eqns. 5.30 and 5.31, we can obtain state probability $\mathbf{P}_K$. Hence, we can compute the steady-state probabilities for $\mathbf{P}_n; 0 \le n \le K-1$ from Eqn. 5.29. We have also developed the MATLAB program to compute the numerical value of steady-state probabilities.

5.4 System Performance Measures

Our optimal analysis is based on the following system performance characteristics of the studied F-policy $M/M/1/K$ queueing system with unreliable service and exponential startup time.

The expected number of customers in the system

$$L_S = \mathbf{P}_K\left[\sum_{n=1}^{K-1} n\boldsymbol{\Psi}^*{}_n\mathbf{e}_1 + K\mathbf{e}_2\right] \tag{5.32}$$

Throughput of the system

$$\tau_p = \beta_1\mathbf{P}_K\left[\sum_{n=1}^{K-1}\boldsymbol{\Psi}^*{}_n\mathbf{u}_1 + \mathbf{e}_3\right] \tag{5.33}$$

The effective arrival rate

$$\lambda_{eff} = \mathbf{P}_K \left[\lambda \left(\mathbf{\Psi^*}_0 \mathbf{e}_4 + \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_2 \right) + \beta_2 \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_1 \right] \tag{5.34}$$

The expected waiting time in the system

$$W_S = \frac{\mathbf{P}_K \left[\sum_{n=0}^{K-1} n \mathbf{\Psi^*}_n \mathbf{e}_1 + K \mathbf{e}_2 \right]}{\mathbf{P}_K \left[\lambda (\mathbf{\Psi^*}_0 \mathbf{e}_4 + \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_2) + \beta_2 \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_1 \right]} \tag{5.35}$$

The probability that the server is busy

$$P_B = \mathbf{P}_K \left[\sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_3 + \mathbf{e}_4 \right] \tag{5.36}$$

The probability that the server starts to allow customers to enter the system

$$P_S = \mathbf{P}_K \left[\mathbf{\Psi^*}_0 \mathbf{e}_3 + \sum_{n=1}^{F} \mathbf{\Psi^*}_n \mathbf{u}_4 \right] \tag{5.37}$$

The probability that the system is blocked

$$P_L = \mathbf{P}_K \left[\mathbf{\Psi^*}_0 \mathbf{e}_3 + \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_4 + e_2 \right] \tag{5.38}$$

The probability that the customer is allowed to enter the system

$$P_A = \mathbf{P}_K \left[\mathbf{\Psi^*}_0 \mathbf{e}_4 + \sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_2 \right] \tag{5.39}$$

The probability that customer immediately leave after service is rendered

$$P_U = \mathbf{P}_K \left[\sum_{n=1}^{K-1} \mathbf{\Psi^*}_n \mathbf{u}_1 + \mathbf{e}_3 \right] \tag{5.40}$$

The frequency that customers are not allowed to join the system

$$FF = \mathbf{P}_K \left[\beta_2 \mathbf{e}_3 + \lambda \mathbf{\Psi}^*_{K-1} \mathbf{u}_2 \right] \tag{5.41}$$

where $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_{K-1}, \mathbf{P}_K]$, $\mathbf{e}_3$=$[1,0]^T$, $\mathbf{e}_4$=$\mathbf{e}_2 - \mathbf{e}_3$, $\mathbf{u}_1$=$[1,0,0,1]^T$, $\mathbf{u}_2$=$[0,0,1,1]^T$, $\mathbf{u}_3$=$\mathbf{e}_1$-$\mathbf{u}_1$ and $\mathbf{u}_4$=$\mathbf{e}_1$-$\mathbf{u}_2$.

5.5 Cost Analysis

We evolve an expected cost function for the studied Markovian single server queueing system with F-policy, unreliable service, and exponential startup time. The parameters μ and γ are decision variables. The cost elements related to the different states of the Markovian

model are defined as follows.

$C_H \equiv$ unit holding cost for each customer queued in the system.

$C_B \equiv$ unit cost when the reliable server is busy.

$C_S \equiv$ unit startup cost for letting the customer enter the system.

$C_L \equiv$ unit cost for every lost customer when the limited capacity is full, and the system is blocked.

$C_A \equiv$ unit cost when the server permits the customer to join the system.

$C_U \equiv$ unit cost for the customer immediately after the service rendered.

$C_1 \equiv$ unit cost for service by the reliable server during a normal busy period.

$C_2 \equiv$ unit cost for the server being startup mode.

The expected total cost function is given as

$$TC = C_H L_S + C_B P_B + C_S P_S + C_L \lambda P_L + C_A P_A + C_U P_U + C_1 \mu + C_2 \gamma \tag{5.42}$$

Hence, the optimization problem is formulated as

$$TC(\mu^*, \gamma^*) = \min_{\mu,\gamma} (TC) \tag{5.43}$$

5.6 Grey Wolf Optimizer

With the increased complexity and dimensionality of real-time problems under socio-economic-techno-oriented constraints, the metaheuristic optimization techniques have become very practical because of their flexibility, simplicity, derivation-free mechanism, and local optima avoidance. Metaheuristic techniques have inspirations from animal behaviors, physical phenomena, or evolutionary concepts and apply to diverse problems without any algorithm's structural changes. Metaheuristics techniques are derivation-free mechanisms that optimize stochastically and have a superior ability to avoid local optima.

Mirjalili et al. [136] proposed a new metaheuristic called grey wolf optimizer (GWO) stimulated by the natural leadership hierarchy and hunting mechanism of grey wolves. In recent years, there has been growing interest in developing the variant of GWO and exploring its applicability in managerial decision-making. Zamfirache et al. [225] explore the use of GWO to solve the optimal tuning of fuzzy controllers for the policy iteration reinforcement learning-based control approach. To determine a solar system's highest power point tracking, Aguila et al. [6] developed a discrete proportional-integral-derivative controller optimized using the GWO algorithm. To anticipate and optimize indoor air quality, thermal comfort levels, and energy usage, a back propagation neural network model is integrated by Li et al. [120] with an adaptive multi-objective particle swarm optimizer and GWO algorithm. Thobiani et al. [10] proposed a hybrid optimization technique based on particle swarm optimization to improve GWO for crack detection utilizing inverse analysis. Zhao et

al. [228] utilized GWO to obtain the optimal control variables sequences, which safeguard a lower turbine outlet temperature with unchanged thrust. Chouar et al. [31] dealt with the cost reduction and lead time improvement in a physical internet-supply chain network using a hybrid framework based on an improved GWO and an artificial neural network. Indramaya et al. [86] compared the strengths, weaknesses, nature, and behavior of the various collective intelligence metaheuristic algorithms in solving various benchmarking problems and concluded that GWO is the most efficient algorithm.

5.6.1 Inspiration

Grey wolves are well-thought-out apex predators and mostly favor to render the livelihood in a pack of size 5-12 on average with a strict social dominant hierarchy: W_α, W_β, W_δ, and W_ω.

- W_α: The W_α, the best in management, is mainly accountable for making decisions democratic about hunting and verbalized to the pack.
- W_β: The wolves W_β are subordinates who help the W_α in decision-making as an advisor or their pack activities as discipliners. The W_β reinforces the W_α's guidelines throughout the pack and gives feedback to the W_α and probably the best candidate to be the W_α if W_α becomes inefficient for hunting.
- W_δ: Wolf responsible as elders, caretakers, scouts, sentinels, and hunters belong to category W_δ, which are superior to W_ω but have a submission to W_α and W_β.
- W_ω: The bottommost ranking wolf is W_ω, which acts as a scapegoat and always has to submit to all the other dominant wolves.

Hunting in the group is another motivating social behavior of grey wolves. The main phases of hunting are as follows: (i) Tracking, chasing, and impending the prey; (ii) pursuing, encircling, and niggling the prey until it stops moving; (iii) attack towards the prey. Next, the hunting technique and the social hierarchy of grey wolves are modeled mathematically to design stochastic GWO and optimize.

5.6.2 Mathematical Model and Algorithm

For mathematically modeling the wolves' social hierarchy system, the fittest solution is coined as W_α; consequently, the second and third best solutions are titled as W_β and W_δ, and the rest are classified as W_ω. In the GWO algorithm, the hunting mechanism is steered by best solution wolves W_α, W_β, and W_δ, wherein the wolves W_ω trail these three dominant

wolves. Grey wolves can recognize the position of prey and encircle them. The second step of encircling the prey by the grey wolves during the hunt is simulated mathematically as

$$\tilde{\mathbf{D}} = |\tilde{\mathbf{C}}.\tilde{\mathbf{G}}_p(n) - \tilde{\mathbf{G}}(n)| \tag{5.44}$$

$$\tilde{\mathbf{G}}(n+1) = \tilde{\mathbf{X}}_p(n) - \tilde{\mathbf{A}}.\tilde{\mathbf{D}} \tag{5.45}$$

$$\tilde{\mathbf{A}} = 2\tilde{\mathbf{a}}.\tilde{\mathbf{r}}_1 - \tilde{\mathbf{a}} \tag{5.46}$$

$$\tilde{\mathbf{C}} = 2.\tilde{\mathbf{r}}_2 \tag{5.47}$$

where

The grey wolves' position (solutions) will change in hyper-cubes (or hyper-spheres) around the optimal solution attained till the current iteration.

For simulating the hunting mechanism guided mainly by the W_α and occasionally by the W_β and W_δ, we assume that the W_α, W_β, and W_δ have better knowledge about the potential location of prey. We save the first three best solutions obtained till the current iteration and oblige the other search agents, including W_ωs, to keep posted on their positions according to the current position of the best search agent employing the following formula.

$$\tilde{\mathbf{D}}_\alpha = |\tilde{\mathbf{C}}_1.\tilde{\mathbf{G}}_\alpha(n) - \tilde{\mathbf{G}}(n)|; \tag{5.48a}$$

$$\tilde{\mathbf{D}}_\beta = |\tilde{\mathbf{C}}_2.\tilde{\mathbf{G}}_\beta(n) - \tilde{\mathbf{G}}(n)|; \tag{5.48b}$$

$$\tilde{\mathbf{D}}_\delta = |\tilde{\mathbf{C}}_3.\tilde{\mathbf{G}}_\delta(n) - \tilde{\mathbf{G}}(n)|; \tag{5.48c}$$

Hence,

$$\tilde{\mathbf{G}}_1 = \tilde{\mathbf{G}}_\alpha - \tilde{\mathbf{A}}_1.\tilde{\mathbf{D}}_\alpha; \tag{5.49a}$$

$$\tilde{\mathbf{G}}_2 = \tilde{\mathbf{G}}_\beta - \tilde{\mathbf{A}}_2.\tilde{\mathbf{D}}_\beta; \tag{5.49b}$$

$$\tilde{\mathbf{G}}_3 = \tilde{\mathbf{G}}_\delta - \tilde{\mathbf{A}}_3.\tilde{\mathbf{D}}_\delta; \tag{5.49c}$$

So,

$$\tilde{\mathbf{G}}(n+1) = \frac{\tilde{\mathbf{G}}_1 + \tilde{\mathbf{G}}_2 + \tilde{\mathbf{G}}_3}{3} \tag{5.50}$$

The W_α, W_β, and W_δ estimate the approximate position of the target, and other search agents (wolves) update their position nearby randomly around the prey.

The last step of hunting is attacking the target when the prey stops moving. For simulating this mechanism mathematically, we decrease the value of $\tilde{\mathbf{a}}$ and hence the fluctuation range of $\tilde{\mathbf{A}}$.

In the GWO algorithm, search agents keep their position posted based on the location of the W_α, W_β, and W_δ, and attack towards the target. The GWO algorithm is susceptible to stagnation in local solutions, and the encircling mechanism discussed prompts exploration to some extent, but GWO needs more operators to accentuate exploration. The position of wolves W_α, W_β, and W_δ are mainly governed by search. W_α, W_β, and W_δ diverge from each other to hunt for prey and converge to attack the target. To represent the diverging behavior

of the wolves (search agent) from the prey to find fitter prey, we take $|\tilde{\mathbf{A}}| > \mathbf{1}$. The diverging behavior allows the proposed algorithm to search globally, i.e., exploration by a search agent. The exploration in GWO is also enhanced by vector $\tilde{\mathbf{C}}$ containing random values in $[0,2]$ that provide arbitrary weights for prey. The random weights emphasize the effect of prey in defining the distance in Eqn. 5.44 if the value is greater than or equal to one or otherwise deemphasized. Since the element's value in vector $\tilde{\mathbf{C}}$ is not linearly decreasing in contrast to the element of vector $\tilde{\mathbf{A}}$, it promotes GWO to illustrate a more random behavior throughout the optimization. It favors exploration during initial and final iterations and local optima avoidance. The vector component $\tilde{\mathbf{C}}$ is beneficial in local optima stagnation, especially in the terminating iterations. The realistic effect of obstacles in hunting to approaching prey in nature is also modeled by considering random vector $\tilde{\mathbf{C}}$. The random vector $\tilde{\mathbf{C}}$ figures the obstacles in nature that act in the hunting paths of wolves. It prevents them from quickly and conveniently approaching prey, i.e., depending on the position of a wolf. It can randomly give the prey a weight, making it harder and farther to reach for wolves or vice versa.

In GWO, the optimization process is initiated by a random population of grey wolves (candidate solutions). As iterations proceed, α, β, and δ wolves estimate the probable position of the prey, and each candidate solution keeps posted its distance from the prey. The proposed social hierarchy assists GWO in saving the best solutions acquired so far throughout the iteration. The proposed encircling mechanism outlines a circle-shaped neighborhood candidate around the solutions, which can be protracted to higher dimensions as a hypersphere with different random radii, governed by the random parameters $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$. The proposed hunting method consents to candidate solutions to locate the probable position of the prey. The algorithm GWO considers that the parameter $\tilde{\mathbf{a}}$ is decreased from 2 to 0, which governs the value of vector $\tilde{\mathbf{A}}$ to emphasize exploration and exploitation. Exploration and exploitation are certified by the adaptive values of $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{A}}$, allowing GWO to transition smoothly between exploration and exploitation. Candidate solutions tend to deviate from the prey when $|\tilde{\mathbf{A}}| > \mathbf{1}$ and congregate towards the prey when $|\tilde{\mathbf{A}}| < \mathbf{1}$. With decreasing $\tilde{\mathbf{A}}$, half of the iterations are favored to search (exploration) ($|\tilde{\mathbf{A}}| \geq \mathbf{1}$) and the other half are committed to exploitation ($|\tilde{\mathbf{A}}| < \mathbf{1}$). In the GWO, only two main parameters $\tilde{\mathbf{a}}$ and $\tilde{\mathbf{C}}$ must be adjusted. Finally, the GWO iterative algorithm ends on achieving the terminating criterion. The flow chart for grey wolf optimization technique is depicted in Fig.5.1.

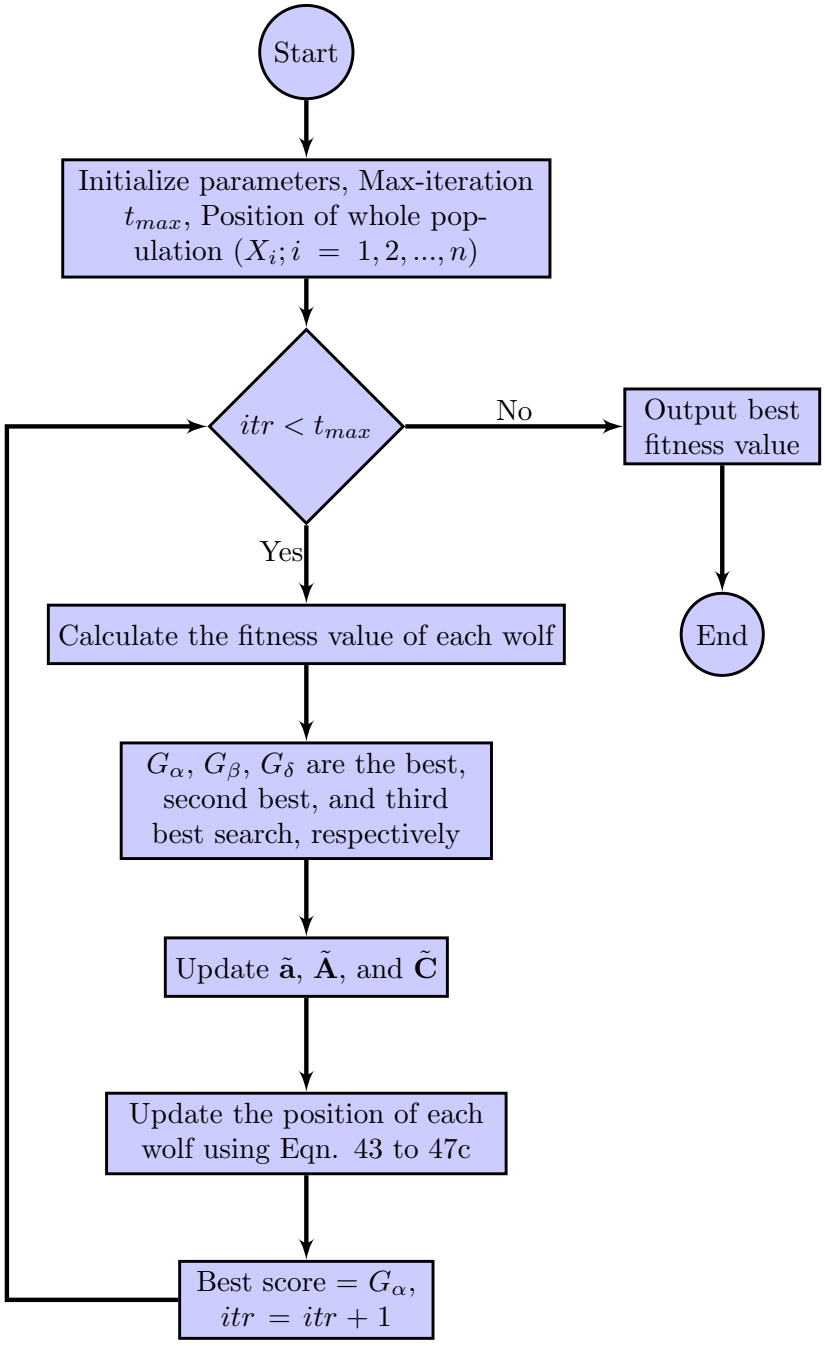


Figure 5.1: The flow chart of the grey wolf optimization algorithm

Algorithm 5 The pseudo-code for the iterative GWO algorithm

Pre-set the grey wolf population $\widetilde{\mathbf{X}}_i(i=1,2,...,n)$
Set $\tilde{\mathbf{a}}$, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$
Estimate the fitness of each search agent
$\tilde{\mathbf{G}}_\alpha$ = the best search agent
$\tilde{\mathbf{G}}_\beta$ = the second best search
$\tilde{\mathbf{G}}_\delta$ = the third best search
while (terminating criterion) or ($n <$ Max number of iterations) **do**
    for each search agent **do**
        Keep posted on the position of the current search agent by Eqn. 5.50
    end for
    Keep posted $\tilde{\mathbf{a}}$,$\tilde{\mathbf{A}}$, and $\tilde{\mathbf{C}}$
    Estimate the fitness of all search agents
    Update $\tilde{\mathbf{G}}_\alpha$, $\tilde{\mathbf{G}}_\beta$ and $\tilde{\mathbf{G}}_\delta$
    $itr = itr+1$
end while
Output: Return $\tilde{\mathbf{G}}_\alpha$.

Table 5.1: The control parameters of algorithms and corresponding value

Control Parameter	Numerical Value	Equation Number
Number of dimension	2	
Number of population	50	
Number of iteration	100	
Number of run	10	
μ_0	$[1,3]$	
γ_0	$[0.01,0.1]$	
$\tilde{\mathbf{r}}_1$	$U[0,1)$	
$\tilde{\mathbf{r}}_2$	$U[0,1)$	
$\tilde{\mathbf{a}}$	$2-itr\left(\frac{2-0}{100}\right)$	
$\tilde{\mathbf{A}}_1,\tilde{\mathbf{A}}_2,\tilde{\mathbf{A}}_3$	$2\tilde{\mathbf{a}}\tilde{\mathbf{r}}_1-\tilde{\mathbf{a}}$	5.46
$\tilde{\mathbf{C}}_1,\tilde{\mathbf{C}}_2,\tilde{\mathbf{C}}_3$	$2\tilde{\mathbf{r}}_2$	5.47
$\tilde{\mathbf{D}}_\alpha,\tilde{\mathbf{D}}_\beta,\tilde{\mathbf{D}}_\delta$		5.48
$\tilde{\mathbf{G}}_1,\tilde{\mathbf{G}}_2,\tilde{\mathbf{G}}_3$		5.49

5.7 Special Cases

We present the following special cases of our studied model that give similar results that are known for the queueing systems published in the literature.

Case 1: When $\gamma \to \infty$ and $F = K - 1$, the studied model reduces as finite capacity $M/M/1$ queueing model with unreliable service ([148]).

Case 2: When $\beta_1 \to \infty$, the model resembles with Markovian single server queueing model with F-policy ([71]).

Case 3: For $\beta_1 \to \infty, \gamma \to \infty$ and $F = K - 1$, the model approaches to classical finite capacity single server Markovian queueing model.

Case 4: For $0 < \beta_1 < \infty$, $\beta_2 = 0$, $\gamma \to \infty$ and $F = K - 1$, if we set $\mu = \beta_1$, the model reduces to standard $M/E_{k=2}/1/K$ queueing model.

Case 5: In case 4, if we set $\beta_2 = 0$ & $\mu > \beta_1$, the model deduces to a finite capacity single server with hyperexponential service time distribution $M/HE/1/K$ queueing model.

5.8 Numerical Results

With the technological advancement, the service systems have a wide area of applications such as shopping malls, service windows, internet of things, cloud computing, communication systems, computer systems, etc. In congestion, efficient service facilities are significant. The service facility is also directly related to optimal strategic design, reputation, availability, development, and competitiveness. In order to make efficient service system that meet users' requirements, it is necessary to measure and predict the system indices effectively. Using the presented theoretical and numerical results, system managers or policymakers can predict the efficiency of the service system of interest and do an excellent job of managing and controlling system quality.

The numerical results for different experiments conducted on MATLAB (R2018b, 64-bit, License number 925317) on a computing system with configuration Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz with RAM 16.0 GB for hospital management of 20 beds where patient arrive randomly with the aim of better diagnosis and treatment and the results are summarized in Figs. 5.2-5.12 and Table 5.2-5.7. For Figs. 5.2-5.12, the default parameters are fixed as follows $K = 20$, $F = 8$, $\lambda = 1$, $\mu = 8$, $\beta_1 = 0.9$, $\beta_2 = 0.1$, $\gamma = 8$ that are estimated from the records and examine the effect of parameters $K, F, \lambda, \mu, \beta_1, \beta_2$ on system

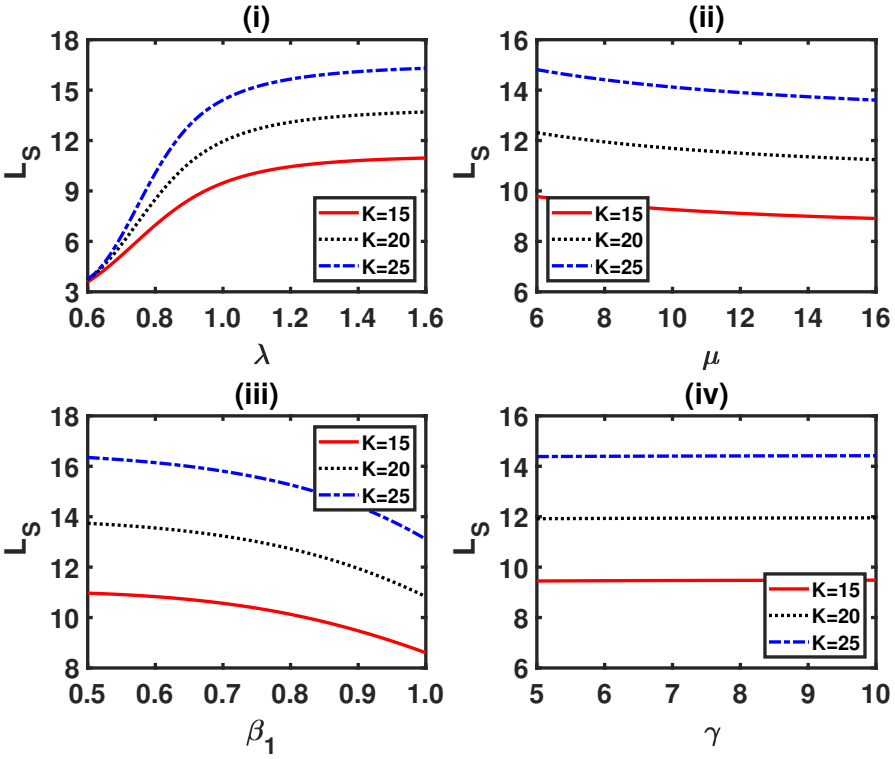


Figure 5.2: Expected number of customers in the system (L_S) for different parameters

performances as the values of one or two of these parameters vary given that others are fixed as above.

In Fig. 5.2 and Fig. 5.3, we depict the variation of the expected number of customers in the system L_S to governing parameters for thresholds K and F, respectively. As the threshold values K and F increase, the value of L_S increases. The increasing trend of L_S is observed for arrival rate λ whereas decreasing trends are observed wrt to service rate μ and β_1. The mild increasing change is detected for the startup rate γ. The apparent results verify the stochastic modeling of the studied service system.

The deviation in the expected waiting time of the customers in the system W_S to system parameters is presented in Fig. 5.4 and Fig. 5.5 for different threshold values K and F. Similar results for W_S are perceived as for L_S above for all thresholds and parameters. The apparent effects of W_S also support the correct mathematical modeling and theoretical results. The optimal service strategies are needed to reduce the long queue and long delays in waiting.

The bar graphs in Fig. 5.6 and Fig. 5.7 represent the changing trend in throughput of the system Th for varied values of system parameters. More system throughput is shown

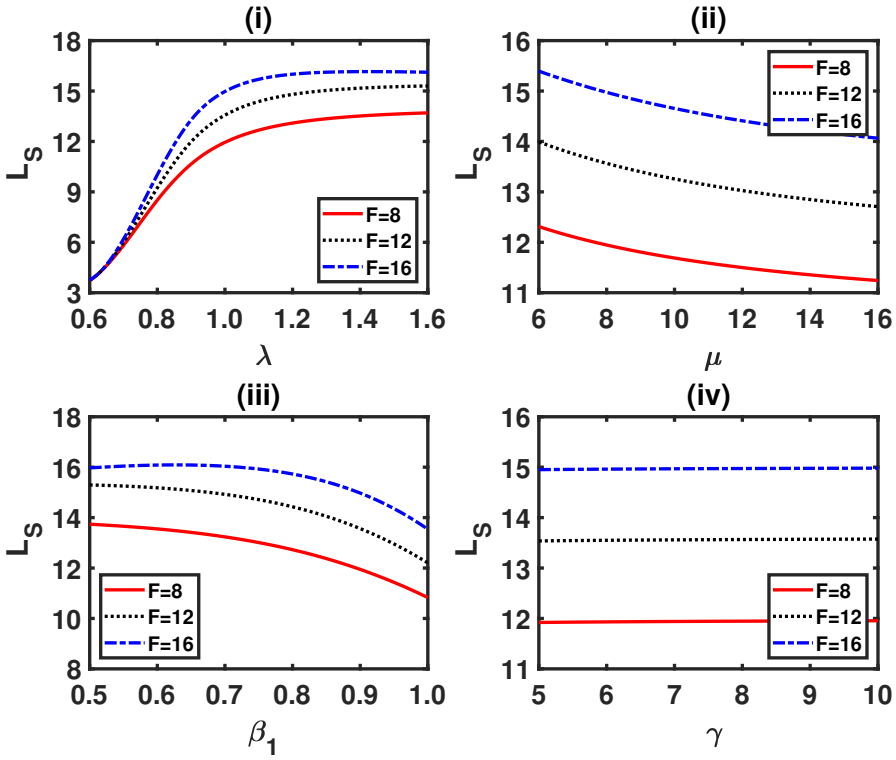


Figure 5.3: Expected number of customers in the system (L_S) for different parameters

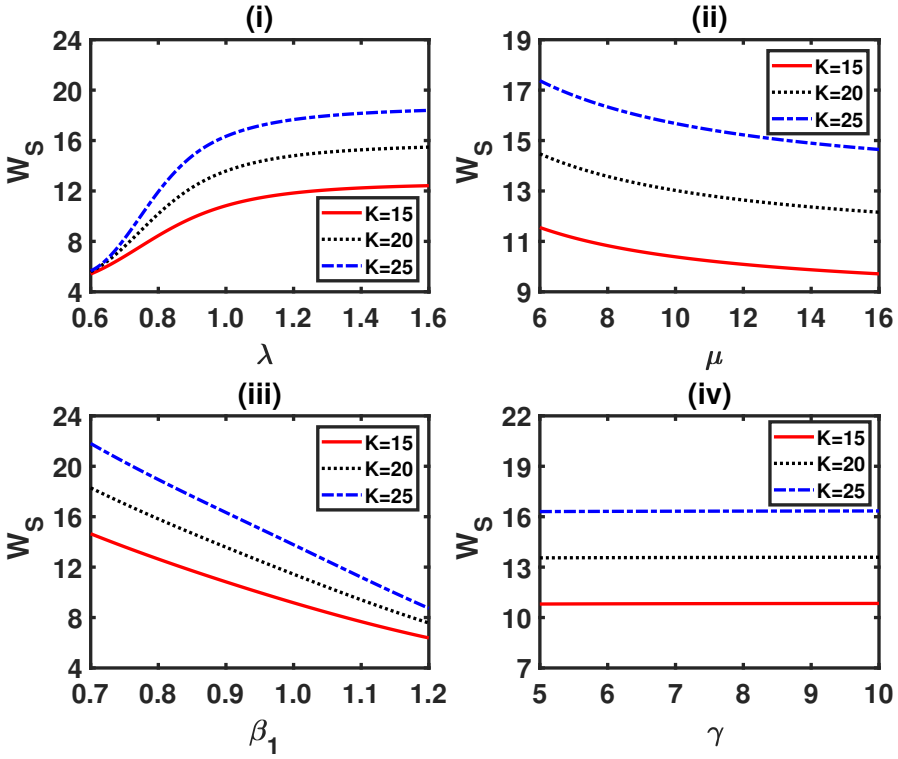


Figure 5.4: Expected waiting time of the customers in the system (W_S) for different parameters

for increased arrival rate λ. It is due to more number of customers in the system. The system's throughput Th increases with the service rate μ and rate of successful service β_1. The trifling results are witnessed for startup time γ.

Besides the above-considered default value of system parameters, for figuring the change in expected total cost TC formulated in Eqn. 5.42, we set different unit costs value as follows $C_H = 25$, $C_B = 200$, $C_S = 400$, $C_L = 100$, $C_A = 50$, $C_U = 10$, $C_1 = 50$, $C_2 = 10$. We plot the variation of TC for varied rates and thresholds in Fig. 5.8 and Fig. 5.9. The palpable trends are noticed, which is evident in our expected total cost formulation and modeling to be correct. The illustrated results prompt an exploration of the optimal strategies for an efficient service system at minimum incurred costs.

For that purpose, we portray a line graph, surface plot, and contour plot in Fig. 5.10 for decision parameters μ and γ for default values of thresholds, rates, and costs, as assumed above. All graphs prove that the expected total cost is a convex function of decision parameters μ and γ. In Fig. 5.12, we display the plot of optimal TC^* for different iterations for multiple runs and observe the convergence to the same value for all runs. It supports our choice of grey wolf optimization for optimal analysis. Analytically, it is impossible to establish since TC is the function of system performances which are the expression of state probabilities we get on solving the governing Chapman-Kolmogorov differential-difference

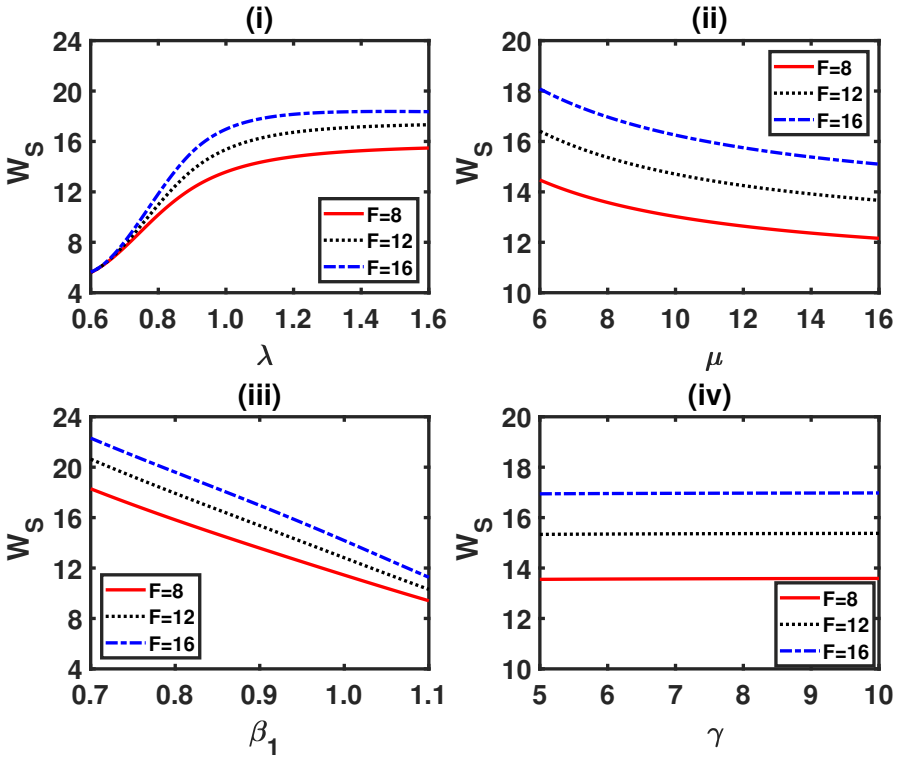


Figure 5.5: Expected waiting time of the customers in the system (W_S) for different parameters

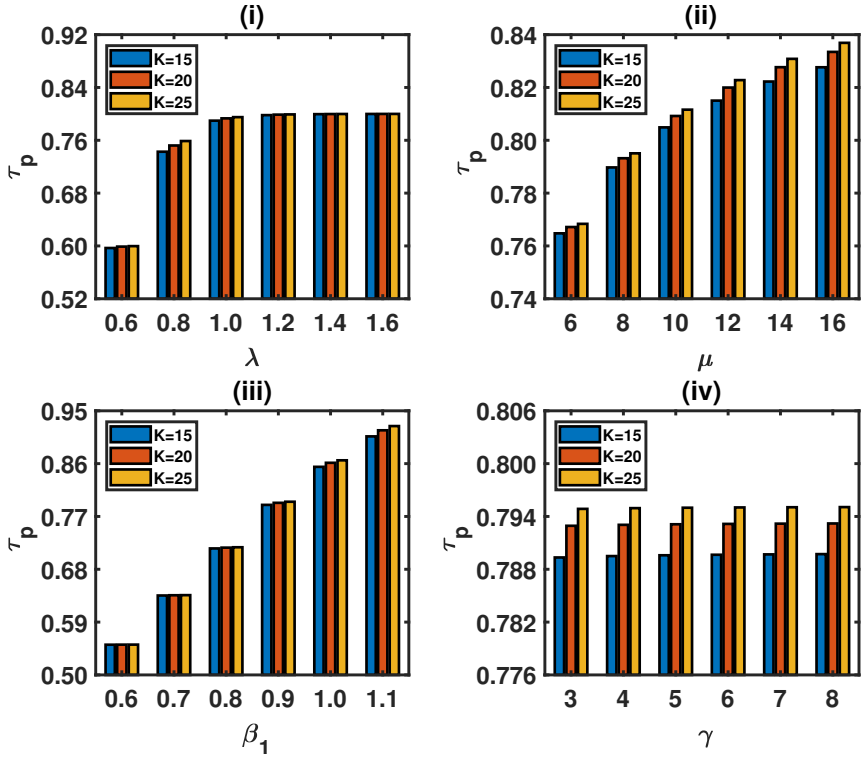


Figure 5.6: Throughput of the system (τ_p) for different parameters

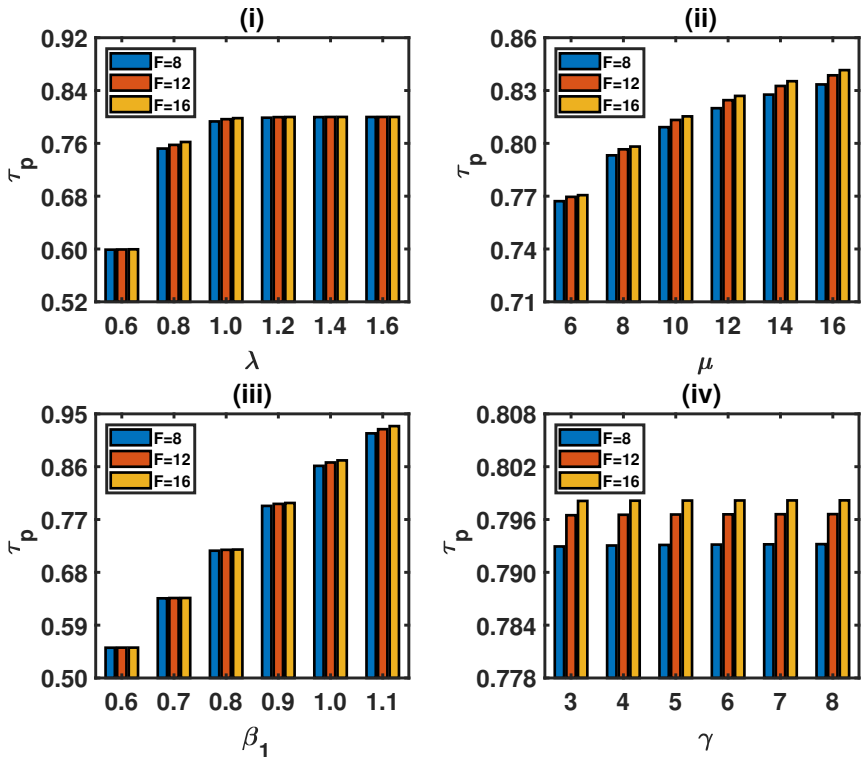


Figure 5.7: Throughput of the system(τ_p) for different parameters.

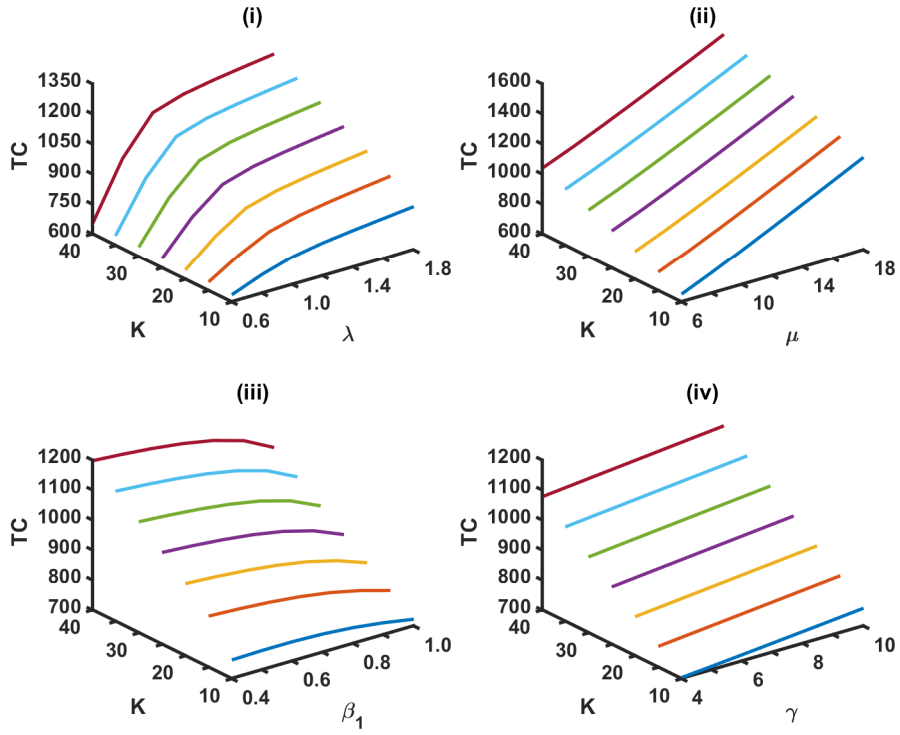


Figure 5.8: Expected total cost of the system (TC) for different parameters

equations. To obtain the optimal value of decision parameters μ and γ, we employ the quasi and metaheuristic optimization techniques and show that metaheuristic is very useful for the optimal analysis of complex real-time systems.

We employ the quasi-Newton method and grey wolf optimizer elaborated in the previous section to determine the optimal value of decision parameters μ and γ and associated optimal expected total cost TC in Eq.5.43. For this purpose, we set the default value of thresholds, rates, and incurred unit costs as follows K=20, F=8, λ=0.6, $\mu = 2$, β_1=0.7, β_2=0.3, $\gamma = 0.02$, C_H=22, C_B=190, C_S=380, C_L=95, C_A=40, C_U=8, C_1=40, C_2=8 and results are tabulated in Table 5.2-5.7.

In Table 5.2 and Table 5.3, we illustrate the iteration of the quasi-Newton method with the initial value of decision parameters $\mu_0 = 2$ and $\gamma_0 = 0.02$ with tolerance 10^{-7}. We compile the expected total cost corresponding to each iteration, the gradient of TC wrt μ and γ. The last row gives optimal value of μ, γ, and TC say μ^*, γ^*, and $TC(\mu^*, \gamma^*)$ where $\max\left[\frac{\partial TC}{\partial \mu}, \frac{\partial TC}{\partial \gamma}\right] < 10^{-7}$.

In Table 5.4-5.5, the results of optimal analyses via the quasi-Newton method are summarized different sets of system parameters and costs. For different sets of system parameters $K, F, \lambda, \beta_1, \beta_2$, the optimal value of decision parameters μ^* and γ^*, and the optimal value of expected total cost TC obtained by quasi-Newton method are tabulated in Table 5.4 with

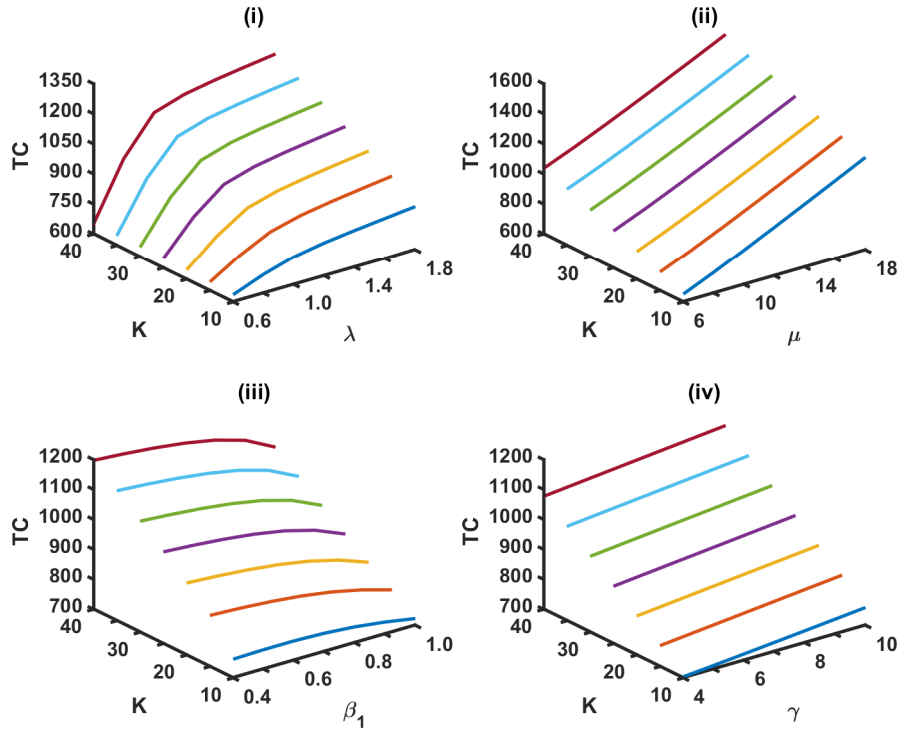


Figure 5.9: Expected total cost of the system(TC) for different parameters

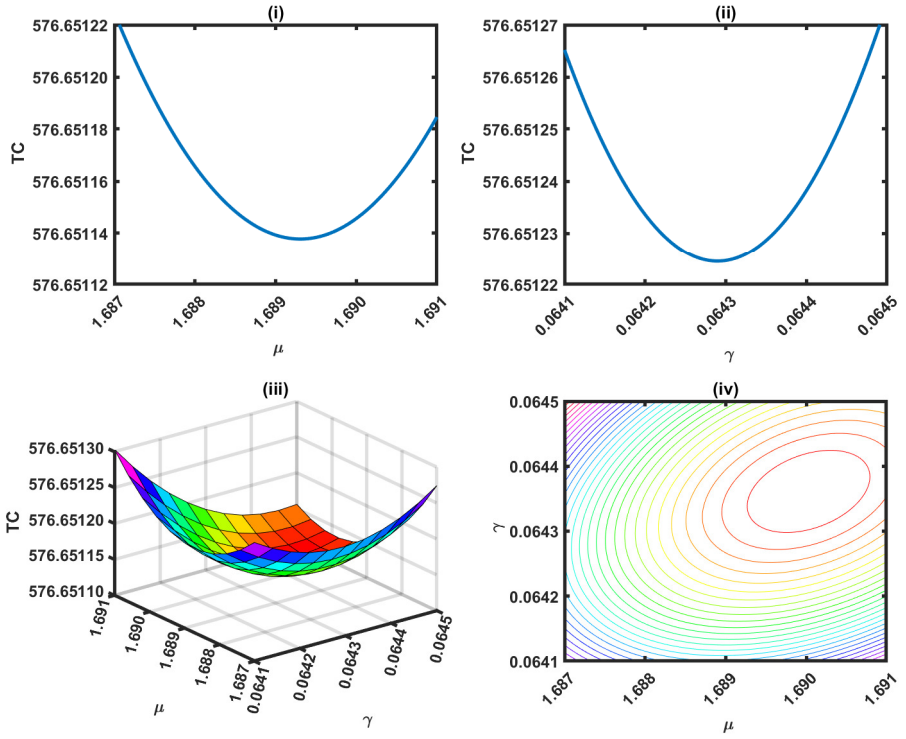
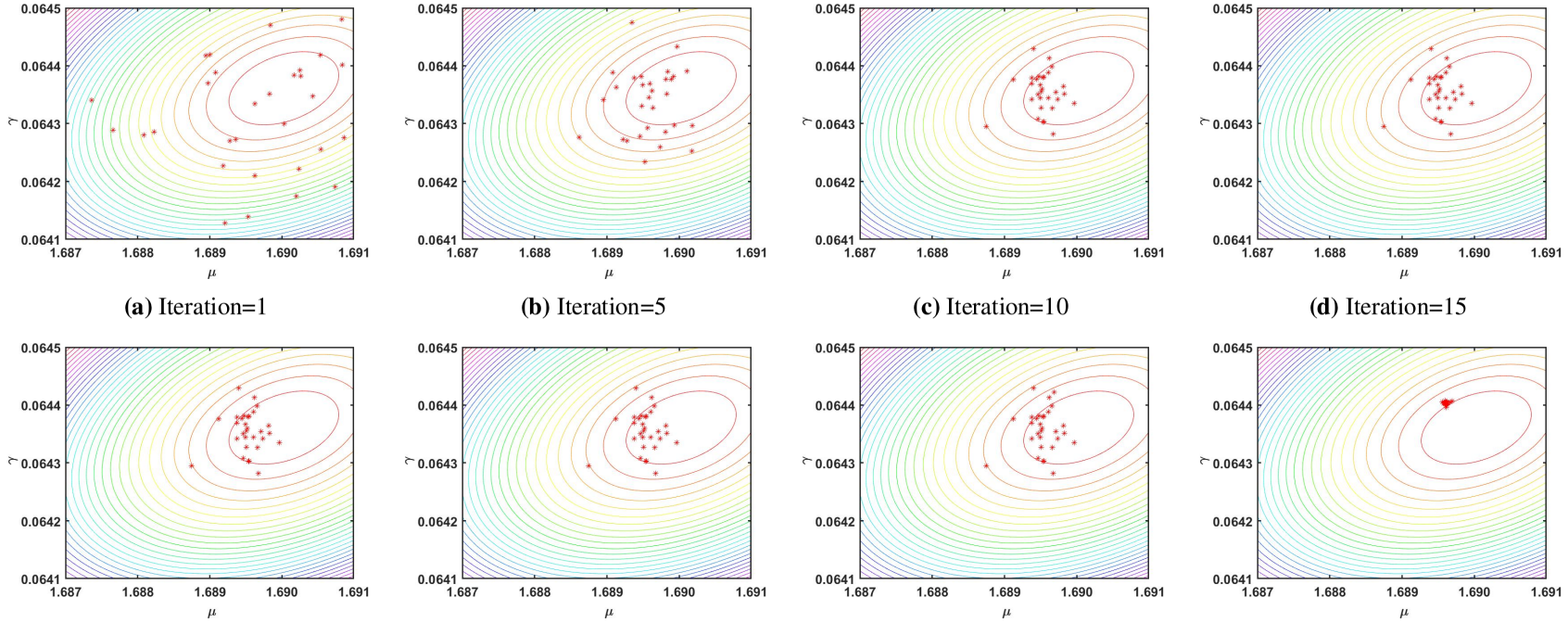


Figure 5.10: Expected total cost of the system (TC) for different parameters.



(a) Iteration=1 **(b)** Iteration=5 **(c)** Iteration=10 **(d)** Iteration=15

(e) Iteration=20 **(f)** Iteration=25 **(g)** Iteration=30 **(h)** Iteration=35

Figure 5.11: Several generations of GWO algorithm on the contour of $TC(\mu, \gamma)$

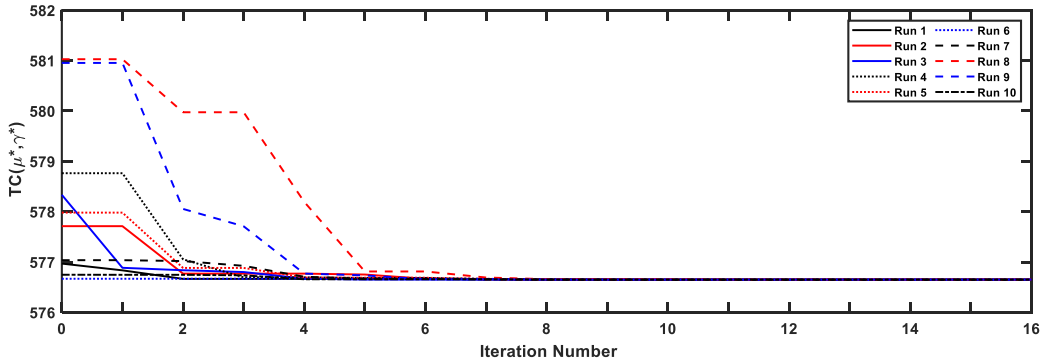


Figure 5.12: Convergence of iteration of Grey-wolf optimization.

the number of iterations for the initial value of decision parameters $\mu_0 = 2$ and $\gamma_0 = 0.02$. The optimal expected total cost $TC(\mu^*, \gamma^*)$ increases in general with K, F, λ, and β_2 and decreases with β_1 with few exceptional. The observation about the optimal value of μ and γ and the number of iterations can be inferred from the table. Similar data are generated for different set of unit costs C_H, C_B, C_S, C_L, C_A, C_U, C_1, C_2, and summarized in the Table 5.5. It is observed that the optimal value of TC increases by increasing the value of unit costs. There is a tremendous increase in cost related to K and F. The optimal value of decision parameters can also be a significant element for policymakers.

For the same sets of system parameters and unit costs considered in Table 5.4-5.5, the optimal analysis results through grey wolf optimizer (GWO) have been compiled in Table 5.6-5.7, respectively. For the implementation of GWO, we set parameters as per Table 5.1. Table 5.6-5.7 summarizes the results in terms of μ^*, γ^*, and $TC(\mu^*, \gamma^*)$ and be verified with results in Table 5.3-5.5. For almost all the sets, we have similar results. It evidences that a metaheuristic method GWO suits such complex real-time problems. For the statistical validation of GWO convergent results, we compute the mean and maximum of $\frac{\min TC_i}{TC_i}$. The *mean* $\left[\frac{\min TC_i}{TC_i}\right]$ ranges from 1.0000001 to 1.0000036 whereas *max* $\left[\frac{\min TC_i}{TC_i}\right]$ ranges from 0.0000007 to 0.0000082. It shows how GWO is close to the optimal solution for multiple runs. Table 5.8 summarizes the optimal value and time elapsed for the parameters in Table 5.6 using genetic algorithm and PSO. It supports our choice to choose GWO since in both criterion of minimum value and time elapsed, GWO is depicting the better results.

In a nutshell, we have the following inferential remarks;

- The incurred cost function is vital to adjudge the economic benefits.
- The setting of optimal strategies for providing the service is essential for making decisions optimally. The service rate μ and startup rate γ must be controlled per our constraints.

- The long queue or waiting delay must be reduced since it increases the expected total cost. For this purpose, we must optimally set thresholds F and K.
- The new metaheuristic optimization techniques GWO is suitable for complex real-time applications due to its highly and fastly convergent results.

5.9 Conclusion

In this chapter, we have conceptualized the notion of unreliable service for a F-policy $M/M/1/K$ service system with an arrival control policy. The proposed model is described with the help of system assumptions, notations, system states, and their steady-state equations. We computed the stationary distribution of the model using the matrix analytical method and derived various system performances in vector form. Further, we have constructed the total cost function of the system using derived performances and parameters by multiplying cost weights. Numerical experimentation and optimal analysis are carried out to perform the sensitivity of system parameters on different performances measures and total cost function. Numerical contribution to improving the system indices of this chapter is threefold: (i) increasing the service rate μ, (ii) setting the startup time optimally γ, and (iii) identifying the appropriate thresholds F and K to avoid long waiting delay. The numerical improvements by the implementation of quasi-Newton method and the GWO algorithm for the optimal analysis showed that for the initial input data for parameters taken in numerical results, the optimal range for decision parameters μ and γ varies in the range of $[1.022, 3.63]$ and $[0.009, 1.03]$, respectively, with initial values of $\mu_0 = 2$ and $\gamma_0 = 0.02$. The GWO works effectively in solving complex problems because the operators are intended to enable the GWO to avoid local optima successfully and swiftly converge to the optimum. The results showed that GWO could provide highly promising results and enhance the applicability of the proposed algorithm in solving real problems.

Table 5.2: Iteration of quasi-Newton method with $\mu_0 = 2$, $\gamma_0 = 0.02$

Iteration	μ	γ	TC	$\frac{\delta TC}{\delta \mu}$	$\frac{\delta TC}{\delta \gamma}$	**max**($\lvert \frac{\delta TC}{\delta \mu} \rvert, \lvert \frac{\delta TC}{\delta \gamma} \rvert$)
0	2.00000000	0.02000000	646.227860	8.84097990	787.89913146	787.89913146
1	2.85900237	0.03515981	635.414895	1.76549296	303.17835678	303.17835678
2	2.78283146	0.04925326	632.618038	0.04510714	112.74884768	112.74884768
3	2.81989965	0.06348239	631.628336	0.07679416	36.20630731	36.20630731
4	2.83657441	0.07358311	631.416489	0.02215468	7.99205526	7.99205526
5	2.84205447	0.07728663	631.400494	0.00238554	0.69596913	0.69596913
6	2.84258682	0.07767200	631.400324	0.00002436	0.00806698	0.00806698
7	2.84259332	0.07767656	631.400323	0.00000000	0.00002034	0.00002034
8	**2.84259334**(μ^*)	**0.07767657**(γ^*)	**631.400323**	**0.00000001**	**0.00000004**	**0.00000004**

Table 5.3: Iteration of quasi-Newton method with $\mu_0 = 2$, $\gamma_0 = 0.02$

Iteration	μ	γ	TC	$\frac{\delta TC}{\delta \mu}$	$\frac{\delta TC}{\delta \gamma}$	**max(**$\lvert \frac{\delta TC}{\delta \mu} \rvert, \lvert \frac{\delta TC}{\delta \gamma} \rvert$**)**
0	2.00000000(μ_0)	0.02000000(γ_0)	701.154910	19.41081754	492.28955726	492.28955726
1	0.80467822	0.02338923	698.481124	31.22801544	18.78154534	31.22801544
2	1.08981135	0.01597005	693.704680	5.07745123	108.65749383	108.65749383
3	0.93310801	0.00042313	690.276484	35.97110869	1008.98358385	1008.98358385
4	0.94779552	0.00438211	693.208133	11.20102563	397.76992355	397.76992355
5	0.98743480	0.00733549	694.192765	2.94030124	119.81651635	119.81651635
6	1.01471423	0.00895416	694.348535	0.51889882	21.54156598	21.54156598
7	1.02199049	0.00938425	694.355567	0.02834695	1.21354769	1.21354769
8	1.02244650	0.00941086	694.355603	0.00010437	0.01187982	0.01187982
9	1.02245142	0.00941105	694.355603	0.00000003	0.00007775	0.00007775
10	1.02245146	0.00941105	694.355603	0.00000001	0.00000049	0.00000049
11	**1.02245146**(μ^*)	**0.00941105**(γ^*)	**694.355603**	**0.00000002**	**0.00000005**	**0.00000005**

Table 5.4: Optimal expected total cost of the system ($TC(\mu^*, \gamma^*)$) for different parameters via quasi-Newton method with $\mu_0 = 2$, $\gamma_0 = 0.02$

K,F,λ,β_1,β_2	**Number of iteration**	μ^*	γ^*	$TC(\mu^*, \gamma^*)$
12,8,0.7,0.7,0.3	12	1.84219911	1.02742684	409.768440
16,8,0.7,0.7,03	10	2.21667825	0.15375096	448.469201
20,8,0.7,0.7,0.3	8	2.15352938	0.07265412	482.553711
24,8,0.7,0.7,0.3	7	1.97774011	0.03512229	516.703050
20,8,0.7,0.7,0.3	8	2.15352938	0.07265412	482.553711
20,10,0.7,0.7,0.3	7	2.27895689	0.05034637	498.380621
20,12,0.7,0.7,0.3	6	2.37039346	0.03603930	513.316935
20,14,0.7,0.7,0.3	7	2.40805553	0.02580218	527.133510
20,8,0.4,0.7,0.3	9	2.66908442	0.05495393	282.827206
20,8,0.6,0.7,0.3	8	2.84259334	0.07767657	447.400323
20,8,0.7,0.7,0.3	8	2.15352938	0.07265412	482.553711
20,8,0.8,0.7,0.3	9	1.90998398	0.08244469	503.061301
20,8,0.7,0.4,0.3	11	1.02245146	0.00941105	510.355603
20,8,0.7,0.6,0.3	8	1.81515633	0.07103110	492.717514
20,8,0.7,0.7,0.3	8	2.15352938	0.07265412	482.553711
20,8,0.7,0.8,0.3	9	2.76021334	0.08498858	466.044204
20,8,0.7,0.7,0.3	8	2.15352938	0.07265412	482.553711
20,8,0.7,0.7,0.4	8	2.15849534	0.07027238	490.369408
20,8,0.7,0.7,0.5	8	2.15845330	0.06787837	497.545129
20,8,0.7,0.7,0.6	8	2.15405544	0.06544041	507.164713

Table 5.5: Optimal expected total cost of the system ($TC(\mu^*, \gamma^*)$) for different costs via quasi-Newton method with μ_0=2, γ_0=0.02

C_h,C_b,C_s,C_I,C_a,C_u,C_1,C_2	**Iteration Number**	μ^*	γ^*	$TC(\mu^*, \gamma^*)$
16,190,380,95,40,8,40,8	10	2.51264066	0.12686454	387.352355
18,190,380,95,40,8,40,8	9	2.63335773	0.10556719	407.870956
20,190,380,95,40,8,40,8	9	2.74273600	0.08993149	427.865243
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	441.400323
22,180,380,95,40,8,40,8	8	2.79804651	0.07793918	444.920976
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,200,380,95,40,8,40,8	8	2.88595302	0.07741142	449.849892
22,210,380,95,40,8,40,8	8	2.92821456	0.07714374	452.271085
22,190,370,95,40,8,40,8	8	2.82036021	0.07324288	446.723290
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,390,95,40,8,40,8	9	2.86290327	0.08206858	448.040432
22,190,400,95,40,8,40,8	9	2.88159173	0.08643423	448.647710
22,190,380,75,40,8,40,8	8	2.77940270	0.07303782	445.166479
22,190,380,85,40,8,40,8	8	2.81158961	0.07537966	446.293890
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,380,105,40,8,40,8	8	2.87252232	0.07993293	448.487178
22,190,380,95,35,8,40,8	8	2.86760469	0.07955952	443.307341
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,380,95,45,8,40,8	8	2.81683592	0.07576547	451.479712
22,190,380,95,50,8,40,8	8	2.79027058	0.07382373	455.544707
22,190,380,95,40,6,40,8	8	2.85124983	0.07832521	445.998508
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,380,95,40,10,40,8	8	2.83384922	0.07702463	448.800541
22,190,380,95,40,12,40,8	8	2.82501508	0.07636928	450.199130
22,190,380,95,40,8,30,8	8	3.63826754	0.08835508	415.201084
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,380,95,40,8,50,8	8	2.24283827	0.06800464	472.696179
22,190,380,95,40,8,60,8	8	1.78754542	0.05921334	492.740465
22,190,380,95,40,8,40,6	8	2.84466686	0.07884673	447.243706
22,190,380,95,40,8,40,8	8	2.84259334	0.07767657	447.400323
22,190,380,95,40,8,40,10	8	2.84060201	0.07656842	447.554663
22,190,380,95,40,8,40,12	8	2.83868570	0.07551627	447.706843

Table 5.6: Optimal expected total cost of the system ($TC(\mu^*, \gamma^*)$) for different parameters via grey wolf optimizer

K,F,λ,β_1,β_2	μ^*	γ^*	**TC**(μ^*, γ^*)	**mean** $\frac{TC_i}{TC^*}$	**max** $\frac{TC_i}{TC^*}$	**time elapsed**
12,8,0.7,0.7,0.3	1.8431884689	1.0325070091	409.768621	1.0000001491	1.0000017439	1763
16,8,0.7,0.7,0.3	2.2143374754	0.1545298087	448.469316	1.0000002194	1.0000025547	1904
20,8,0.7,0.7,0.3	2.1556934604	0.0727751349	482.553749	1.0000003475	1.0000048061	1936
24,8,0.7,0.7,0.3	1.9805291261	0.0351228455	516.703128	1.0000009014	1.0000015926	1965
20,8,0.7,0.7,0.3	2.1556934604	0.0727751349	482.553749	1.0000003475	1.0000048061	1936
20,10,0.7,0.7,0.3	2.2741238240	0.0505077588	498.380943	1.0000005096	1.0000045061	1836
20,12,0.7,0.7,0.3	2.3705880865	0.0360333850	513.316939	1.0000003431	1.0000091126	1845
20,14,0.7,0.7,0.3	2.4093529960	0.0258576606	527.133468	1.0000001529	1.0000018074	1827
20,8,0.4,0.7,0.3	2.6655501808	0.0537961313	282.827758	1.0000008461	1.0000011539	1826
20,8,0.6,0.7,0.3	2.8434909463	0.0774845883	447.400376	1.0000003016	1.0000020993	2105
20,8,0.7,0.7,0.3	2.1556934604	0.0727751349	482.553749	1.0000003475	1.0000048061	1936
20,8,0.8,0.7,0.3	1.9082800134	0.0827186100	503.061373	1.0000007661	1.0000046925	1839
20,8,0.7,0.4,0.3	1.2308877407	0.5005673021	496.114694	1.0000004832	1.0000082480	1820
20,8,0.7,0.6,0.3	1.8168786510	0.0702710969	492.718065	1.0000004791	1.0000076015	1811
20,8,0.7,0.7,0.3	2.1556934604	0.0727751349	482.553749	1.0000003475	1.0000048061	1936
20,8,0.7,0.8,0.3	2.7559686810	0.0849062047	466.044339	1.0000008056	1.0000017105	1798
20,8,0.7,0.7,0.3	2.1556934604	0.0727751349	482.553749	1.0000003475	1.0000048061	1936
20,8,0.7,0.7,0.4	2.1603465768	0.0702083501	490.369455	1.0000007059	1.0000077977	1805
20,8,0.7,0.7,0.5	2.1551815131	0.0680339802	497.545260	1.0000006277	1.0000050431	1821
20,8,0.7,0.7,0.6	2.1582301028	0.0651698432	504.165018	1.0000006503	1.0000041073	1802

Table 5.7: Optimal expected total cost of the system ($TC(\mu^*, \gamma^*)$) for different costs via grey wolf optimizer

C_h,C_b,C_s,C_l,C_a,C_u,C_1,C_2	μ^*	γ^*	TC^*	**mean**$(\frac{TC_i}{TC^*})$	**max**$(\frac{TC_i}{TC^*})$	**time elapsed**
18,190,380,95,40,8,40,8	2.6332854464	0.1060480891	407.870956	1.0000005893	1.0000028177	1882
20,190,380,95,40,8,40,8	2.7429275988	0.0890508003	427.865315	1.0000003832	1.0000007983	1822
22,190,380,95,40,8,40,8	2.8434909463	0.0774845883	431.400376	1.0000003016	1.0000020993	2105
22,180,380,95,40,8,40,8	2.8089873178	0.0782848657	444.921007	1.0000013282	1.0000067148	1785
22,200,380,95,40,8,40,8	2.8810982783	0.0776312830	449.850125	1.0000036178	1.0000071254	1922
22,190,370,95,40,8,40,8	2.8216544948	0.0728172669	446.723546	1.0000002187	1.0000014124	1874
22,190,390,95,40,8,40,8	2.8633003873	0.0825180919	448.040555	1.0000003116	1.0000070892	1835
22,190,380,75,40,8,40,8	2.7796874516	0.0736342418	445.166770	1.0000003112	1.0000092191	1833
22,190,380,85,40,8,40,8	2.8114119104	0.0762947958	446.294596	1.0000002608	1.0000014072	1932
22,190,380,95,35,8,40,8	2.8718059403	0.0802513995	443.307747	1.0000003685	1.0000007922	2247
22,190,380,95,45,8,40,8	2.8295829423	0.0764890536	451.481079	1.0000002170	1.0000031247	1917
22,190,380,95,40,6,40,8	2.8537438344	0.0785267357	445.998558	1.0000004134	1.0000058923	1758
22,190,380,95,40,10,40,8	2.8404650090	0.0777960455	448.801195	1.0000003184	1.0000079418	1840
22,190,380,95,40,8,30,8	3.6354787641	0.0887718965	415.201224	1.0000003328	1.0000010765	1799
22,190,380,95,40,8,50,8	2.2431160010	0.0681445857	472.696185	1.0000001476	1.0000012311	1792
22,190,380,95,40,8,40,6	2.8466725593	0.0791390205	447.243767	1.0000001175	1.0000014215	1855
22,190,380,95,40,8,40,10	2.8368538211	0.0769681013	447.554925	1.0000072762	1.0000076668	2031

Table 5.8: Optimal expected total cost of the system ($TC(\mu^*, \gamma^*)$) for different parameters via genetic algorithm and particle swarm optimization

	Genetic Algorithm		**Particle Swarm Optimization**	
K,F,λ,β_1,β_2	$TC(\mu^*, \gamma^*)$	**Time Elapsed**	$TC(\mu^*, \gamma^*)$	**Time Elapsed**
12,8,0.7,0.7,0.3	415.505382	2079.16	420.422605	2267.72
16,8,0.7,0.7,0.3	457.494825	2264.57	461.026457	2461.87
20,8,0.7,0.7,0.3	492.204824	2268.18	496.547808	2513.19
24,8,0.7,0.7,0.3	523.420269	2346.98	530.736785	2542.71
20,8,0.7,0.7,0.3	492.204824	2268.18	496.547808	2513.19
20,10,0.7,0.7,0.3	507.351799	2173.82	513.088165	2387.35
20,12,0.7,0.7,0.3	520.503376	2175.25	525.970202	2392.05
20,14,0.7,0.7,0.3	535.567603	2188.78	538.487923	2397.94
20,8,0.4,0.7,0.3	294.140868	2163.81	297.607968	2362.78
20,8,0.6,0.7,0.3	452.769181	2504.81	458.822507	2714.608
20,8,0.7,0.7,0.3	492.204824	2268.18	496.547808	2513.19
20,8,0.8,0.7,0.3	513.122601	2186.57	516.392499	2364.22
20,8,0.7,0.4,0.3	508.517561	2193.58	510.273807	2375.1
20,8,0.7,0.6,0.3	502.036017	2148.33	505.8194384	2347.68
20,8,0.7,0.7,0.3	492.204824	2268.18	496.547808	2513.19
20,8,0.7,0.8,0.3	480.491868	2132.80	478.705108	2349.25
20,8,0.7,0.7,0.3	492.204824	2268.18	496.547808	2513.19
20,8,0.7,0.7,0.4	504.590169	2093.37	504.901554	2343.79
20,8,0.7,0.7,0.5	512.972256	2120.05	514.801972	2385.52
20,8,0.7,0.7,0.6	519.749134	2145.91	523.115527	2339.72

Chapter 6

Finite Capacity Service System with Partial Server Breakdown and Recovery Policy: An Economic Perspective

Developing a comprehensive service strategy for optimizing customer satisfaction is an ongoing challenge for a successful facility provider. The critical points of comprehensive systems are selecting the suitable service design, establishing an effective service delivery process, and building continuous improvement. This study analyzes a finite capacity service system with several realistic customer-server phenomena: customer impatience, server's partial breakdown, and threshold recovery policy. When the number of customers is more, the server is under pressure to increase the service rate to reduce the service system's load.

6.1 Background

Under socio-econo-techno constraints, an efficient service system is vital for continuous and sustainable development in the fast-growing competitive world. Effective service includes customer satisfaction and uninterrupted, quality, cost-effective, time-prompt service. In congestion, we have to experience the cognition of waiting in a queue and waiting for our turn to seek the hassle-free service. Waiting is ubiquitous and reinforces strategic research on critical areas of the service facility. The optimal service design ranges from optimal capacity to uninterrupted availability, optimal service rate to optimal cost, quality to prompt service, etc. Decision-makers must explore the existing technological innovation and the acceptance of the customers to design a service system.

In today's technological era, the study of queueing-based service systems has centered on the mainstream due to the growing importance and complexity. This chapter highlights essential reflections of a queueing-based service system using several realistic queueing notions such as balking, service pressure coefficient, threshold-based recovery policy, and partial server breakdown. A thorough survey of the literature on queueing systems with the above-mentioned queueing notions shows that these queueing notions have been rarely studied in conjunction with different theoretical concepts. Quality of providing service and operational efficiency is the most crucial factor for organizations, either service or production. Over the past few decades, there has been an increased interest among researchers, system analysts, and decision-makers/policymakers in congestion problems, including work related to server breakdown, threshold-based recovery policies, and service pressure coefficients.

In congestion, impatience is prevalent among the customers. In general, at the epoch of arrival, if the server is unavailable either due to busy in serving waiting customers or breaks down, customers may show a reluctance attitude to join the queue and therefore may be uncertain whether to enter the service system. The longer the waiting queue, the higher the likelihood of customers balking. Haight [72] was the first researcher who introduced the notion of customer balking in the queueing literature for a Markovian environment. Later, Haight [74] again envisaged a single service provider Markovian queue that characterized the customers' continuous abandonment. Abou-El-Ata and Shawky [3] investigated Markovian overflow queue with balking behavior of customers. Abou-El-Ata and Hariri [4] extended the analytical solution for the multi-server Markovian queue with customer impatience. Drekic and Woolford [45] investigated a priority queue assigning low priority to impatient customers. Lozano and Moreno [125] studied the abandonment behavior of arrived customers in a single-server service system in a discrete-time environment with an

infinite/finite buffer. Sun et al. [179] explored the customer impatience (balking) in a single server Markovian environment with the double-adaptive working vacation (WV) policy. Since impatience attributes directly affect the quality of service (QoS), queueing problems with the attribute of impatience customers have motivated many scholars to investigate a distinguished service environment (*cf.* [168], [167]).

The efficient service system is dynamic with a load of customers improving customer service to impact customer retention levels. Under the pressure of increased congestion, the server may tempt to increase the service efficiency. This chapter also incorporates the concept of service pressure coefficient to model real-time strategic policy. The pressure coefficient is an absolute constant value and defines as the amount to which the server increases the service capacity (rate) to diminish the over waiting load of the service system. For the higher backlog of waiting, there is a high chance that the servers may start operating intensely until the backlog becomes small or non-existent. Wang and Lin[197] anticipated the concept of pressure conditions for the service systems for the first time in the queueing literature. Wang et al. [198] examined the warm-standby provisioning machine interference problem with multiple-imperfect coverage and multiple-server with the pressure condition for improving the repair rate. More recently, Shekhar et al. [170] conceptualized service pressure conditions for retaining the reneged customers in the multi-server Bernoulli's vacation queueing problem.

The literature on queue-based service systems is rich with assumptions about reliable servers, which is seldom. The service provider is subject to breakdowns randomly at any instant in practice. Most research findings on queueing-based service systems with server breakdown consider that the server terminates working completely when the breakdown occurs. Nevertheless, in practice, some real-time systems exist in which the service provider still works at a lesser rate in breakdown state, which referred to working breakdown or partial breakdown in the queueing literature (*cf.* [178], [96], [96], [119], [124]) studied the single server Markovian queue with working breakdown. A detailed survey on queueing-based service systems with the breakdown of the server is provided by Krishnamoorthy et al. [111]. Liou [122] explored the matrix method for a single server queue with customer impatience and servers' working breakdown. Yang and Chen [210] analyzed a single server service system with the working breakdown and optional service policies. Rajadurai [157] employed the supplementary variable technique to analyze the general retrial queue with the catastrophic conditions and working breakdown under multiple working vacation policies. Recently, Yen et al. [223] dealt with a retrial MRP with the working breakdown & exponentially start-up time and implemented the PSO algorithm to establish the optimal management policy with optimal joint values of the faster and slower service rates simultaneously at the

minimum mean cost of the system.

The breakdown of the server leads to massive congestion or high impatience attributes among the customers, which increases the economic losses, customer dissatisfaction, etc. The breakdown of the service facility needs strategic recovery. The present study uses strategic corrective measures: threshold recovery policy. According to these economic corrective measures, when the active server is broken down, the recovery can be performed if there exists a pre-specified ($T(1 \leq T \leq K)$) number of customers in the service system. The concept of threshold recovery policy was firstly introduced by Efrosinin and Semenova [53]. Jain and Bhagat [89] envisaged a finite capacity retrial queueing-based service system with a threshold recovery policy for unreliable servers. Yang et al. [212] formulated a cost optimization problem for a threshold-based recovery policy for repairable $M/M/1/N$ system. Yang and Chiang [211] incorporated the concept of threshold recovery policy for a machine interference problem and employed the metaheuristics and PSO algorithm to obtain the converging results along with the mean cost of the machine interference problem.

The cost optimization problems are developed to infer the strategic policies employed in the optimal design. For better understanding of the converging results and utilization of several nature-inspired optimization techniques, one can refer the research works (*cf.* [169], [168], [170], [171]) and references therein.

To the best of our knowledge, no research in the queueing literature has addressed threshold-based recovery policy, servers' working breakdown, customer impatience, and service pressure conditions. This research gap in the literature motivates us for the present study. Moreover, motivated by the results of the nature-inspired algorithms: PSO and CS algorithm, we employ these techniques to optimize the system parameters (*i.e.*, decision variables) and the mean cost of the developed model. A comparative study among CS algorith & PSO algorithm, and QN method has also been conferred to prove the excellence of the metaheuristics. The significant contribution of the present study is to implement the optimization algorithms and to develop MATLAB codes for comparing the findings of the CS algorithm, PSO algorithm, and the QN method in terms of statistical parameters, computation time, and operating policies in optimal conditions, et cetera.

The proposed model has many real-life applications in service systems like computer and communication systems, supply chain management, production systems, inventory control, and machine repair problems. The hardware unit consisting of routers, computers, switches, etc., processes the data packets in several communication systems. When a data packet arrives and finds a long latency, it may lose the information. As the number of data packets load increases in a hardware unit, it extends its built-in standby power to a faster processing

rate. The processing slows down regarding technical issues in the hardware unit or associated software. The persistent technical issues are recovered following some state-dependent strategic policies.

The remaining content of this chapter is framed as follows. Section 6.2 familiarizes the proposed queueing modeling and defines its states with several assumptions and notations. The repeated substitution method and corresponding solution algorithm to compute the steady-state probability distribution are discussed in Section 6.3. Section 6.4 showcases how the system performance indicators are defined and formulated in vector form. Section 6.5 confers the cost function as a constrained optimization problem. Besides this, some of the special cases are provided in Section 6.6. Next, the QN method, PSO & CS algorithms are discussed in detail along with their pseudo-codes in Subsections 6.7.1, 6.7.2, and 6.7.3, respectively. In Section 6.8, several numerical illustrations with the help of numerous graphs and tables are explained. Lastly, in Section 6.9 some of the concluding remarks and future prospects are provided.

6.2 Proposed Model and State Description

The present study develops a finite capacity service system with numerous realistic queueing notions like customer impatience, service pressure coefficient, partial server breakdown, and threshold-based recovery policy. The capacity of the studied service system is proposed as K. The prospective customer joins the service system for intended service following the Poisson process with parameter $\lambda\,(>0)$. If the service facility is idle at the arrival epoch, the customer gets the intended service instantly; otherwise, arrived customer queues in the waiting line. The service provider selects the customer to serve from the queue following the First-Come, First-Served (FCFS) queue discipline. It is assumed that the service times to serve the customers follow an exponential distribution with parameter μ_b during the normal busy state. The server is deteriorated (partially broken down) due to some technical issues that occur following the Poisson process with parameter υ. It continues service uninterruptedly to waiting customers at a slower rate instead of complete termination. The service times during the partial breakdown period of the server also follow an independent and identically (iid) exponentially distributed with rate parameter μ_d. The notion of the threshold recovery policy is employed to abridge the mean cost of the service system due to customers in waiting. According to this, the partial breakdown server is not recovered until the number of customers in the system attains a pre-specified threshold value $T\,(1 \leq T \leq K)$. The recover times of the breakdown server follow an iid exponential distribution with rate parameter ϑ. After accomplishing the recovery action, the server is ready to furnish the service to the

waiting customers immediately at a standard efficiency. When the service provider is busy or malfunctioning, the customers who join the service system tend to become impatient, causing them to depart the system with a probability of $1-\xi$. These customers may remain in the system with the complimentary probability ξ. If the number of customers in the system is T or more, the concept of the service pressure coefficient is considered. The pressure factor is assumed to be dependent on number of customers in the system and parameter ψ. Additionally, we assume that all continuous random variables, namely, inter-arrival times, breakdown times, and service/repair times, are mutually independent. The events arrival, service, repair, recovery, and balking are independent to each other.

Let

$N(t) =$ number of customers in the service system at time instant t, and

$J(t) =$ the server's state at the time t, where

$$J(t) = \begin{cases} 0; & \text{if the service provider is in normal working attribute} \\ 1; & \text{if the service provider is in working breakdown state} \end{cases}$$

Thus, the process $\{(N(t),J(t)); t \geq 0\}$ is Markov chain defined in continuous time as a two-tuple irreducible CTMC with the state-space $\Omega = \{\{(n,0); n = 0,1,2,\ldots,K\} \cup \{(n,1); n = 1,2,\ldots,K\}\}$. Hence, at time instant $t\,(t \geq 0)$, all the system-state probabilities are outlined as follows

$$\pi_{n,0}(t) = \text{Prob}\{N(t) = n, J(t) = 0\}; n = 0,1,2,...,K$$
$$\pi_{n,1}(t) = \text{Prob}\{N(t) = n, J(t) = 1\}; n = 1,2,...,K$$

Assuming all the considerations, the state-dependent mean service rate of the service provider is defined as

$$\mu_b^{(n)} = \begin{cases} \mu_b; & 1 \leq n \leq T-1 \\ \left(\frac{2n}{n+1}\right)^{\psi} \mu_b; & T \leq n \leq K \end{cases}$$
$$\mu_d < \mu_b^{(n)}; \forall n$$

Now, using the theoretical concept and axioms of the QBD (quasi birth and death) process, the system of Chapman–Kolmogorov forward differential-difference equations, that governs the proposed model, is delineated to exhibit the transient-state probabilities representing the likelihood of distinguished states of the service system. Following the different system states, we have

When the server is Idle

$$\pi'_{0,0}(t) = -\lambda \pi_{0,0}(t) + \mu_b^{(1)} \pi_{1,0}(t) \tag{6.1}$$

When the server is in the regular working attribute

$$\pi'_{1,0}(t) = -\Big(\lambda\xi + \mu_b^{(1)} + \upsilon\Big)\pi_{1,0}(t) + \lambda\pi_{0,0}(t) + \mu_b^{(2)}\pi_{2,0}(t) \tag{6.2}$$

$$\pi'_{n,0}(t) = -\Big(\lambda\xi + \mu_b^{(n)} + \upsilon\Big)\pi_{n,0}(t) + \lambda\xi\pi_{n-1,0}(t) + \mu_b^{(n+1)}\pi_{n+1,0}(t); \quad 2 \le n \le T-1 \tag{6.3}$$

$$\pi'_{T,0}(t) = -\Big(\lambda\xi + \mu_b^{(T)} + \upsilon\Big)\pi_{T,0}(t) + \lambda\xi\pi_{T-1,0}(t) + \mu_b^{(T+1)}\pi_{T+1,0}(t) + \vartheta\pi_{T,1}(t) \tag{6.4}$$

$$\pi'_{n,0}(t) = -\Big(\lambda\xi + \mu_b^{(n)} + \upsilon\Big)\pi_{n,0}(t) + \lambda\xi\pi_{n-1,0}(t) + \mu_b^{(n+1)}\pi_{n+1,0}(t) + \vartheta\pi_{n,1}(t); T+1 \le n \le K-1 \tag{6.5}$$

$$\pi'_{K,0}(t) = -\Big(\mu_b^{(K)} + \upsilon\Big)\pi_{K,0}(t) + \lambda\xi\pi_{K-1,0}(t) + \vartheta\pi_{K,1}(t) \tag{6.6}$$

When the server is in working breakdown state

$$\pi'_{0,1}(t) = -\lambda\pi_{0,1}(t) + \upsilon\pi_{0,0}(t) + \mu_d\pi_{1,1}(t) \tag{6.7}$$

$$\pi'_{1,1}(t) = -(\lambda\xi + \mu_d)\pi_{1,1}(t) + \lambda\pi_{0,1}(t) + \upsilon\pi_{1,0}(t) + \mu_d\pi_{2,1}(t) \tag{6.8}$$

$$\pi'_{n,1}(t) = -(\lambda\xi + \mu_d)\pi_{n,1}(t) + \lambda\xi\pi_{n-1,1}(t) + \upsilon\pi_{n,0}(t) + \mu_d\pi_{n+1,1}(t); \quad 2 \le n \le T-1 \tag{6.9}$$

$$\pi'_{n,1}(t) = -(\lambda\xi + \mu_d + \vartheta)\pi_{n,1}(t) + \lambda\xi\pi_{n-1,1}(t) + \upsilon\pi_{n,0}(t) + \mu_d\pi_{n+1,1}(t); \quad T \le n \le K-1 \tag{6.10}$$

$$\pi'_{K,1}(t) = -(\mu_d + \vartheta)\pi_{K,1}(t) + \lambda\xi\pi_{K-1,1}(t) + \upsilon\pi_{K,0}(t) \tag{6.11}$$

At $t = 0$, the initial condition is

$$\begin{cases} \pi_{0,0}(0) = 1 \\ \pi_{n,0} = 0; n = 1,2,\ldots,K \\ \pi_{n,1}(0) = 0; n = 1,2,\ldots,K \end{cases} \tag{6.12}$$

6.3 Steady-State Analysis

In equilibrium condition, *i.e.*, $t \to \infty$, the following are the state probabilities for the analysis of the service system, which are depicted as

$$\text{for} n = 0,1,2,\ldots,K, \lim_{t\to\infty}\pi_{n,0}(t) = \pi_{n,0} \text{ and } \lim_{t\to\infty}\pi'_{n,0}(t) = 0$$

$$\text{for} n = 1,2,\ldots,K, \lim_{t\to\infty}\pi_{n,1}(t) = \pi_{n,1} \text{ and } \lim_{t\to\infty}\pi'_{n,1}(t) = 0$$

Now, to derive the state probability distribution, we adopt the repeated substitution approach as the system of equations is highly complicated to calculate the closed/vector-form of expression of the state probabilities because of intricate constraints like multi-equation,

multi-variable, and multiple parameters. The matrix analytic method was first familiarized by Neuts [142] utilizing the concept of embedded Markov chains for numerous realistic queue-based service systems. For the matrix approach, we characterize the probability vector $\tilde{\mathbf{P}}_n; n = 0,1,2,...,K$ as row vector having steady-state probabilities as elements, i.e., $\tilde{\mathbf{P}}_0 = [\pi_{0,0}]$ and $\tilde{\mathbf{P}}_n = [\pi_{n,0}, \pi_{n,1}]; n = 1,2,\ldots,K$. The transition rate matrix of the Markov chain can equivalently be defined using the QBD process. Hence, by balancing the incoming and outgoing transitions, the tridiagonal generator matrix $\mathbf{Q}$ of the studied CTMC is defined as follows

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_0 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_1 & \mathbf{A}_2 & \mathbf{B}_1 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 & \mathbf{A}_3 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{K-2} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{K-2} & \mathbf{A}_{K-1} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_{K-1} & \mathbf{A}_K \end{bmatrix}$$

The elements of the transition rate matrix $\mathbf{Q}$ as block submatrices are represented as follows.

$$\mathbf{A}_0 = \begin{bmatrix} -\lambda \end{bmatrix}; \quad \mathbf{B}_0 = \begin{bmatrix} -\lambda & 0 \end{bmatrix}; \quad \mathbf{B}_1 = \begin{bmatrix} \lambda\xi & 0 \\ 0 & \lambda\xi \end{bmatrix};$$

$$\mathbf{C}_0 = \begin{bmatrix} \mu_b^1 \\ 0 \end{bmatrix}; \quad \mathbf{A}_n = \begin{bmatrix} a_{11}^{(n)} & a_{12}^{(n)} \\ a_{21}^{(n)} & a_{22}^{(n)} \end{bmatrix}$$

We depict each element of the block submatrix $\mathbf{A}_n; n = 1,2,\ldots,K$ as the scalar $a_{ij}^{(n)}$ whose closed form structure is defined as follows

$$a_{ij}^{(n)} = \begin{cases} -\left(\lambda\xi + \upsilon + \mu_b^{(n)}\right); & i = j = 1 \;\&\; 1 \le n \le K-1 \\ -\left(\upsilon + \mu_b^{(n)}\right); & i = j = 1 \;\&\; n = K \\ \upsilon; & i < j \;\&\; 1 \le n \le K \\ \vartheta; & i > j \;\&\; 1 \le n \le K \\ -(\lambda\xi); & i = j = 2 \;\&\; n = 1 \\ -(\lambda\xi + \mu_d); & i = j = 2 \;\&\; 2 \le n \le T-1 \\ -(\lambda\xi + \vartheta + \mu_d); & i = j = 2 \;\&\; T \le n \le K-1 \\ -(\vartheta + \mu_d); & i = j = 2 \;\&\; n = K \\ 0; & \text{otherwise} \end{cases}$$

Similarly, we define the block submatrix $\mathbf{C}_n; n = 1,2,\ldots,K-1$ as

$$\mathbf{C}_n = \begin{bmatrix} c_{11}^{(n)} & 0 \\ 0 & c_{22}^{(n)} \end{bmatrix}$$

where, element of the matrix $\mathbf{C}_n$ for $n = 1,2,\ldots,K-1$ is the scalar $c_{ii}^{(n)}$ outlined as

$$c_{ii}^{(n)} = \begin{cases} \mu_b^{(n+1)}; & i = 1 \;\&\; 1 \leq n \leq K-1 \\ \mu_d; & i = 2 \;\&\; 1 \leq n \leq K-1 \\ 0; & \text{otherwise} \end{cases}$$

Let $\mathbf{\Pi} = \left[\tilde{\mathbf{P}}_0, \tilde{\mathbf{P}}_1, ..., \tilde{\mathbf{P}}_{K-1}, \tilde{\mathbf{P}}_K\right]$ be the probability vector in equilibrium associated to the pre-defined generator matrix $\mathbf{Q}$. Considering the partition of the probability vector $\mathbf{\Pi}$, we represent governing system of equations in matrix form as

$$\mathbf{\Pi Q} = \mathbf{0} \tag{6.13}$$

The homogeneous governing system of equations 6.13 can straightforwardly be represented in the form of pre-defined block submatrices as

$$\tilde{\mathbf{P}}_0\mathbf{A}_0 + \tilde{\mathbf{P}}_1\mathbf{C}_0 = \mathbf{0} \tag{6.14}$$

$$\tilde{\mathbf{P}}_0\mathbf{B}_0 + \tilde{\mathbf{P}}_1\mathbf{A}_1 + \tilde{\mathbf{P}}_2\mathbf{C}_1 = \mathbf{0} \tag{6.15}$$

$$\tilde{\mathbf{P}}_{n-1}\mathbf{B}_1 + \tilde{\mathbf{P}}_n\mathbf{A}_n + \tilde{\mathbf{P}}_{n+1}\mathbf{C}_n = \mathbf{0}; n = 2,3,...,K-1 \tag{6.16}$$

$$\tilde{\mathbf{P}}_{K-1}\mathbf{B}_1 + \tilde{\mathbf{P}}_K\mathbf{A}_K = \mathbf{0} \tag{6.17}$$

Now, after appropriate matrix operation and recursive substitution of each element, we obtain

$$\tilde{\mathbf{P}}_0 = \tilde{\mathbf{P}}_1\mathbf{C}_0\left(-\mathbf{A}_0^{-1}\right) = \tilde{\mathbf{P}}_1\mathbf{\Xi}_0$$

$$\tilde{\mathbf{P}}_1 = \tilde{\mathbf{P}}_2\mathbf{C}_1\left[-\left(\mathbf{\Xi}_0\mathbf{B}_0 + \mathbf{A}_1\right)^{-1}\right] = \tilde{\mathbf{P}}_2\mathbf{\Xi}_1$$

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_{n+1}\mathbf{C}_n\left[-\left(\mathbf{\Xi}_{n-1}\mathbf{B}_1 + \mathbf{A}_n\right)^{-1}\right] = \tilde{\mathbf{P}}_{n+1}\mathbf{\Xi}_n;\; n = 2,3,...,K-1$$

where,

$$\mathbf{\Xi}_n = \begin{cases} -\mathbf{C}_0\mathbf{A}_0^{-1}; & n = 0 \\ -\mathbf{C}_1\left(\mathbf{\Xi}_0\mathbf{B}_0 + \mathbf{A}_1\right)^{-1}; & n = 1 \\ -\mathbf{C}_n\left(\mathbf{\Xi}_{n-1}\mathbf{B}_1 + \mathbf{A}_n\right)^{-1}; & 2 \leq n \leq K-1 \end{cases}$$

Again by the recursive back substitution, we redefine each of the state probability vector $\tilde{\mathbf{P}}_n$ in the closed product form of $\mathbf{\Xi}_n$; $n = 0,1,2,...,K-1$ as

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_K\{\mathbf{\Xi}_{K-1}\mathbf{\Xi}_{K-2}\mathbf{\Xi}_{K-3}\ldots\mathbf{\Xi}_{n+2}\mathbf{\Xi}_{n+1}\mathbf{\Xi}_n\}; n = 0,1,2,\ldots,K-1$$

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_K\left(\prod_{i=n}^{K-1}\mathbf{\Xi}_i\right) = \tilde{\mathbf{P}}_K\mathbf{\Phi}_n; n = 0,1,2,\ldots,K-1 \tag{6.18}$$

Following the total probability rule, we define the normalization condition for the state-probability distribution as $\mathbf{\Pi e} = 1$, which can be equivalently rewritten using the partition

of the probability vector as

$$\left[\tilde{\mathbf{P}}_0\mathbf{e}_1+\tilde{\mathbf{P}}_1\mathbf{e}_2+\tilde{\mathbf{P}}_2\mathbf{e}_2+\ldots+\tilde{\mathbf{P}}_{K-1}\mathbf{e}_2+\tilde{\mathbf{P}}_K\mathbf{e}_2\right]=1 \tag{6.19}$$

where, $\mathbf{e}_1=[1]$ and $\mathbf{e}_2=[1\ 1]^T$. Now using the eq.^{n}(6.18), the eq.^{n}(6.19) can be redefined as

$$\begin{aligned}
\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1+\left[\tilde{\mathbf{P}}_1+\tilde{\mathbf{P}}_2+\ldots+\tilde{\mathbf{P}}_{K-1}+\tilde{\mathbf{P}}_K\right]\mathbf{e}_2&=1\\
\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1+\left[\tilde{\mathbf{P}}_K\mathbf{\Phi}_1+\tilde{\mathbf{P}}_K\mathbf{\Phi}_2+\ldots+\tilde{\mathbf{P}}_K\mathbf{\Phi}_{K-1}+\tilde{\boldsymbol{P}}_K\right]\mathbf{e}_2&=1\\
\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1+\tilde{\mathbf{P}}_K\left[\mathbf{\Phi}_1+\mathbf{\Phi}_2+\ldots+\mathbf{\Phi}_{K-1}+\mathbf{I}\right]\mathbf{e}_2&=1\\
\Longrightarrow\ \tilde{\mathbf{P}}_K\left[\mathbf{\Phi}_0\mathbf{e}_1+\left(\prod_{n=1}^{K-1}\mathbf{\Phi}_n+\mathbf{I}\right)\mathbf{e}_2\right]&=1
\end{aligned} \tag{6.20}$$

The state probability vector $\tilde{\mathbf{P}}_K$ is evaluated from eq.^{n}(6.17) and eq.^{n}(6.20), henceforth, all the other steady-state probabilities $\tilde{\mathbf{P}}_0,\tilde{\mathbf{P}}_1,\ldots,\tilde{\mathbf{P}}_{K-1}$ are evaluated from the eq.^{n}(6.18).

After the computation of state probabilities, we define various performance indices in the next section to tract the modeling and analyze the efficiency of the service system.

6.4 System Performance Measures

In general, there are many standard system performance indicators that can illustrate the quality performance of the service systems. This chapter also provides several queueing-based system performance indices for finite capacity service systems with service pressure coefficient, threshold-based recovery policy, and working breakdown to outline the modeling and procedure used. These system performance measures are also useful in demonstrating the parametric investigation to achieve the objective of decision-making. Moreover, all the system performance indicators defined in this section are correlated and recognized as prime importance in a specific situation. Next, we characterize these system performance indicators in the closed/vector form in terms of governing state probabilities.

- Mean number of customers in the queueing system

$$L_S=\tilde{\mathbf{P}}_K\left(\sum_{n=1}^{K-1}n\mathbf{\Phi}_n\mathbf{e}_2+K\mathbf{e}_2\right) \tag{6.21}$$

- Mean number of customers in the waiting queue

$$L_Q=\tilde{\mathbf{P}}_K\left(\sum_{n=1}^{K-1}(n-1)\mathbf{\Phi}_n\mathbf{e}_2+(K-1)\mathbf{e}_2\right) \tag{6.22}$$

- Probability that server is in working breakdown state

$$P_{WD}=\tilde{\mathbf{P}}_K\left(\sum_{n=1}^{K-1}\mathbf{\Phi}_n\mathbf{e}_3+\mathbf{e}_3\right) \tag{6.23}$$

  where, $\mathbf{e}_3=[0\ 1]^T$

- Probability that the server is in a busy state

$$P_B = \tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} \mathbf{\Phi}_n \mathbf{e}_4 + \mathbf{e}_4 \right) \tag{6.24}$$

where, $\mathbf{e}_4 = [1\ 0]^T$

- Probability that server is idle

$$P_I = \tilde{\mathbf{P}}_0 \mathbf{e}_1 \tag{6.25}$$

- Throughput of the service system

$$\tau_p = \tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} \mu_b^{(n)} \mathbf{\Phi}_n \mathbf{e}_4 + \mu_b^{(K)} \mathbf{e}_4 + \sum_{n=1}^{K-1} \mu_d \mathbf{\Phi}_n \mathbf{e}_3 + \mu_d \mathbf{e}_3 \right) \tag{6.26}$$

- Average balking rate

$$\text{ABR} = \tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} (1-\xi) \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right) \tag{6.27}$$

- Effective arrival rate

$$\lambda_{\text{eff}} = \lambda \tilde{\mathbf{P}}_0 \mathbf{e}_1 + \tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} \xi \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right) \tag{6.28}$$

- Mean waiting time in the service system

$$W_S = \frac{\tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} n \mathbf{\Phi}_n \mathbf{e}_2 + K \mathbf{e}_2 \right)}{\lambda \tilde{\mathbf{P}}_0 \mathbf{e}_1 + \tilde{\mathbf{P}}_K \left(\sum_{n=1}^{K-1} \xi \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right)} \tag{6.29}$$

Using these defined performance indices, we develop the cost optimization problem in the next section with pertinent decision parameters and design parameters.

6.5 Cost Analysis

For the economical analysis of the studied Markovian single server finite capacity service system, this section comprises the formulation of the mean cost function utilizing different cost factors incurred. The parameters μ_d and μ_b are considered as decision variables. The core purpose of the study is to use the joint stationary probability distribution and system performance characteristics of the developed model to optimize the long-run mean cost at the optimal value of the decision parameters. The cost elements unified to the different system states of the queueing model are defined as follows.

$C_h \equiv$ The unit cost associated with customers waiting in the system

$C_d \equiv$ The unit cost incurred with the partial breakdown of the server

$C_b \equiv$ The unit cost incurred with the busy state of the server

$C_i \equiv$ The unit cost incurred with the idle server

$C_{\mu_b} \equiv$ Unit cost for rendering the service with rate μ_b

$C_{\mu_d} \equiv$ Unit cost for rendering the service with rate μ_d

$C_w \equiv$ cost associated with each waiting customer present in the system

We use the above-defined components related to the mean cost and performance indices defined in the previous section to formulate the cost optimization problem as follows

$$TC(\mu_b, \mu_d) = C_h L_S + C_d P_{WD} + C_b P_B + C_i P_I + C_{\mu_b}\mu_b + C_{\mu_d}\mu_d + C_w W_S \tag{6.30}$$

The considered model's cost optimization (minimization) problem is framed mathematically as an optimal control problem.

$$TC(\mu_b^*, \mu_d^*) = \min_{\mu_d < \mu_b} \{TC(\mu_b, \mu_d)\} \tag{6.31}$$

where μ_b^* and μ_d^* are the optimized values of decision variables that minimize the mean cost.

We employ the classical and meta-heuristic optimization techniques to determine the optimal mean cost. The details and results are discussed in the forthcoming sections.

6.6 Special Cases

In this section, for the validity and tractability of developed model, the comparative study with several existing research articles is provided by relaxing one or more assumptions. It proves that, the results of the governing model resemble with the actual findings in the queueing literature.

Case 1: For $\xi = 1$, $\mu_b^{(n)} = \mu$, and $\upsilon = 0$, the studied model analogous to classical $M/M/1/K$ queueing model (*cf.* Kleinrock [107]).

Case 2: For $\xi \neq 1$, $\mu_b^{(n)} = \mu$, and $\upsilon = 0$, our model and findings match with the outcomes of a queueing system with balking proposed by Haight [72].

Case 3: By substituting $\xi \neq 1$, $\mu_b^{(n)} = \mu$, $\upsilon \neq 0$, and $\vartheta = 0$, the governing model converts to the queueing problem with working breakdown and customer impatience investigated by Liou [122].

Case 4: By taking $\xi = 1$, and $\upsilon = 0$, the studied model deduces to a queueing model with service pressure coefficient proposed by Hiller and Lieberman [82].

Case 5: In the case when $\xi = 1$, $\mu_b^{(n)} = \mu$, $\upsilon \neq 0$, and $\vartheta = 0$, the current model resembles with the single server service system with working breakdown of the server proposed by Kalidass et al. [96].

Case 6: By setting the combination of parameters as $\xi = 1$, $\mu_b^{(n)} = \mu$, $\upsilon \neq 0$, and $\vartheta \neq 0$, the model becomes a finite capacity queue-based service system with working breakdown and threshold-based recovery policy which was examined by Efrosinin et al. [53] in the literature.

6.7 Optimization Techniques

We employ the classical and meta-heuristic techniques to determine the optimal value of decision parameters at the minimum mean cost. The results of each technique are compared to others to validate the newly evolved meta-heuristic techniques. The results are compiled in the next section. In this section, we give detail, algorithm, and pseudo-code of classical method: quasi-Newton method (QN), and meta-heuristic optimization techniques: particle swarm optimization (PSO), cuckoo search (CS).

6.7.1 Quasi-Newton Method

The literature on algorithms shows that gradient-based optimization algorithms have errors (*i.e.* zigzagging) when dealing with ill-conditioned optimization problems. Therefore, the quasi-Newton technique of order two is gaining interest as it uses curvature information and efficacy in dealing with ill-conditioned cost optimization problems. Second-order techniques have various benefits over the first-order methods, including a high rate of local converging simulations (usually super-linear) and preserving invariance (non-sensitiveness to the choice of coordinates). Inspiring by this fact, we have incorporated the semi-classical optimizer: the QN method, for the governing multi-objective problem. The advantage of the QN method for multi-objective and multi-constraint optimization is that the estimation of Jacobian matrices is reasonably faster than their actual estimation. This change is significantly more apparent when the range of the problem's solution space is extensive. The QN method is explained in more details in Subsection 1.7.5 along with its pseudo-code.

6.7.2 Particle Swarm Optimization

PSO is a bio-inspired process that searches for an optimal solution in the solution space globally. PSO algorithm is best suited for non-linear, non-convex, and multi-modal optimization problems. Multiple local and global optimal are present, and we need to obtain the global optimum of the problem. In PSO, we use both global best (p_{gb}^t) and the individual (particle) best $(p_i^{t^*})$ simultaneously at the iteration t. Using certain individuals best aims to

escalate the diversity in the promising solutions; however, this diversity may be mimicked by employing randomization. As a result, if the optimization problem of interest is substantially non-linear and multi-modal, there is no convincing justification for choosing the individual best [218]. An elementary set of locations (solutions) (p_i^0) and velocities (v_i^0) are generated randomly for each particle (bird) in the swarm (flock). Each particle's speed is stochastically accelerated towards its prior best position (individual best) and the global best solution across iterations in the search space [121].

$$v_i^{t+1} = v_i^t + c_1 r_1 (p_i^{t^*} - p_i^t) + c_2 r_2 (p_{gb}^t - p_i^t) \tag{6.32}$$

where c_1 and c_2 are positive constants chosen at the initiation of the process. The vector $p_i^{t^*}$ is the finest position (best solution) for the particles till time instant t, determined using the objective function $f(p_i)$ in the local search region. The vector p_{gb}^t is defined as the universally best (*i.e.*, global best) position vector for all the particles. At each iteration, the solution vector is updated to provide the terminating optimum position. The vectors p_i^t and v_i^t are the current values of the position and velocity vector respectively. Furthermore, r_1 and r_2 are the random vectors chosen from the uniformly distributed random variate r_u in the continuous range $[0, 1]$, re-selected at each iteration of the algorithm. Here, randomness shows a significant role in avoiding getting trapped at a local optimum.

The second term of eq.^{n}(6.32) assures complete exploitation of the local area in the search space to find an exact value of the local optimum. Similarly, the third term of eq.^{n}(6.32) prompts that the entire search space is explored to find a global optimum and escape getting trapped at a local optimum. Thus, the choice of c_1 and c_2 is critical in confirming compatibility, and hence their selection should be made sensibly.

Concurrently, each particle is updated according to its velocity. The position updating formula is defined as

$$p_i^{t+1} = p_i^t + v_i^{t+1} \tag{6.33}$$

The above equation explores the process from point to point globally. So we assure that each new point is evaluated for potential improvement. Further, we adopt the concept of inertia function $\Omega(t)$ (*cf.* Shi and Eberhart [172]) to stabilize the exploration of the particles. It stops particles to be stuck in a local region or overshoot from optimum value. Henceforth, the velocity formula is restructured as follows

$$v_i^{t+1} = \Omega v_i^t + c_1 r_1 (p_i^{t^*} - p_i^t) + c_2 r_2 (p_{gb}^t - p_i^t) \tag{6.34}$$

The appropriate value of the inertia function $\Omega(t)$ is taken among the range $[0.5\,0.9]$. The pseudo-code of the PSO algorithm can be characterized as

Algorithm 6 The pseudo-code for PSO algorithm

Input: Objective function, population size, r_1, r_2, c_1, c_2, starting particle position, t_{max};
Initialization: Find position of n particles;
while $t < t_{max}$ or convergence criterion **do**
    for All n particles and all d dimensions **do**
        Update new velocity v_i^{t+1} according to eq.^{n}(6.34);
        Update new position of particle p_i^{t+1} according to eq.^{n}(6.33);
        Evaluate objective function at new position;
        Find the current best position (p_i) for each particle;
    end for
    Update global best p_{gb};
end while
Output: optimal objective value TC^* at p^*.

6.7.3 Cuckoo Search

Generally, cuckoos are captivating birds, not just for their beautiful sounds but also for their aggressive reproduction method. Most of the cuckoo species lay their eggs in communal nests, yet they may throw down the eggs of others to maximize the chances of their eggs hatching. Nevertheless, some species practice obligatory brood parasitism, which involves laying their eggs in the nests of other host birds. Some cuckoo species have evolved due to genetic variation where female parasitic cuckoos are capable of imitating the color and pattern of eggs of certain host species. The behavior lessens the likelihood of their eggs forsaking, increasing their reproductive potential. The competitiveness between cuckoos and host species forms a combat system where cuckoos' eggs can be exposed and thrown down with a probability of P_*.

The resemblance of two eggs (solutions) p_i and p_j can be roughly evaluated by their difference $(p_j - p_i)$. Thus, the location at iteration t can be modified by

$$p_i^{t+1} = p_i^t + \upsilon s \otimes H(P_a - \varepsilon) \otimes (p_j^t - p_k^t) \tag{6.35}$$

where s is step-size, which is ranged by a variable υ consisting of positive values, H is a Heaviside step-function used to simulate the discovery probability with the help of random number ε taken from a uniform distributed range $[0, 1]$. Furthermore, the product notation $\otimes$ of two vectors means entry-wise multiplications. Now, for generating new solution p_i^{t+1} for the i^{th} cuckoo, a L*é*vy flight is performed as

$$p_i^{t+1} = p_i^t + \upsilon L(s, \lambda) \tag{6.36}$$

where the L*é*vy flights are random walks with phases being taken from

$$L(s, \lambda) \sim \frac{1}{s^{1+\lambda}} \left(\frac{\lambda \Gamma(\lambda) \sin(\pi\lambda)/2}{\pi} \right) \tag{6.37}$$

to approximate a Lévy probability distribution with an exponent $0 \leq \lambda \leq 2$. Here, the gamma function is defined as

$$\Gamma(\lambda) = \int_0^\infty z^{\lambda-1} e^{-u} du \tag{6.38}$$

The pseudo-code for the CS algorithm is defined as follows.

Algorithm 7 The pseudo-code for CS algorithm

Input: Objective function $TC(x)$, $x = \{x_1^0, x_2^0, ..., x_d^0\}$, population size, t_{max};
Initialization: Population of n host nests $x_i (1 \leq i \leq n)$;
while $t < t_{max}$ or convergence criterion **do**
    Get a cuckoo randomly (say, i) by Lévy distribution;
    Evaluate its fitness value F_i;
    Choose a nest among n (say, j) randomly;
    Evaluate its fitness value F_j;
    if $(F_i > F_j)$ **then**
        replace j by the new solution;
    end if
    Abandon a fraction (P_a) of worse nests and built new ones;
    Keep the best solutions/nests;
    Rank the solutions/nests and find the current best;
end while
Output: If the stopping criterion is met, then p^* is the best global solution found so far.

6.8 Results and Discussion

In this section, several numerical examples are given to perform the sensitivity analysis of the stationary system performance indices of the proposed single server finite capacity service system for various intricate system parameters. The numerical results and illustrations are outlined in Figs. 6.1–6.4, which show the outcome of several system parameters on the system performance indices, namely, the mean number of customers in the service system (L_S), and throughput of the service system (τ_p). For illustrations, we standardize the capacity of the service system as $K = 15$ and threshold $T = 7$. The other system parameters are fixed as follows: $\lambda = 1.5$, $\xi = 0.7$, $\mu_b = 3.0$, $\mu_d = 1.5$, $\psi = 1.0$, $\upsilon = 0.01$, $\vartheta = 8.0$.

In Figs. 6.1 and 6.2, we illustrate the line graphs for the mean number of customers in the service system wrt λ and μ_b, respectively, for the varied parametric values of design parameters T, ξ, υ, and ψ. It is easy to observe that L_S shows a growing trend for increasing values of λ and the reverse effect for increased values of μ_b as intuitively expected. For the

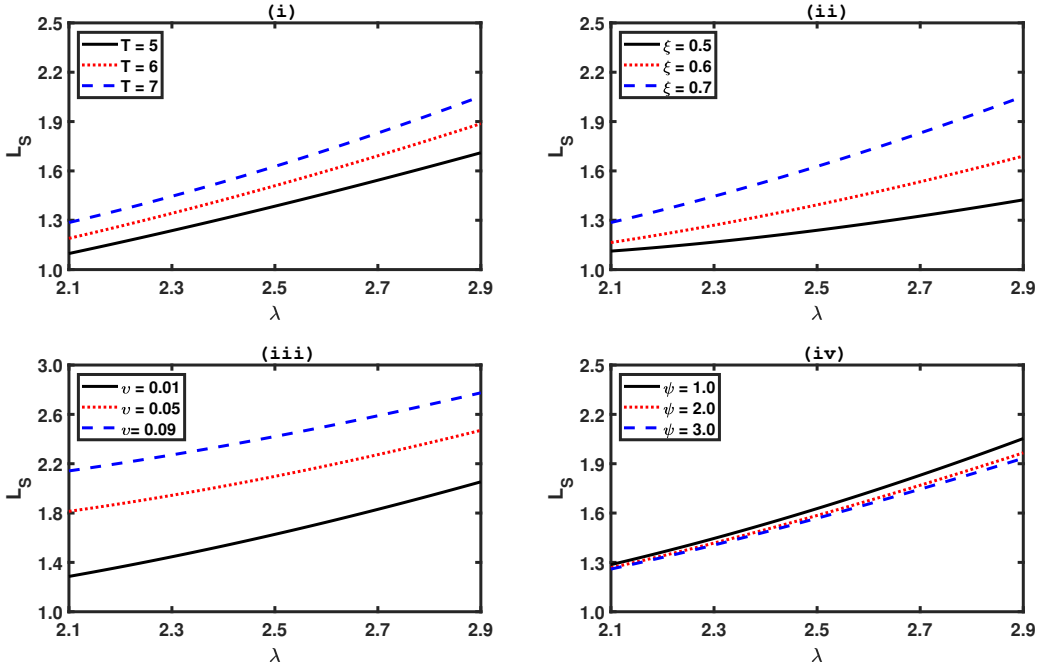


Figure 6.1: Effect of varied (i) T, (ii) ξ, (iii) υ, and (iv) ψ wrt λ on mean number of customers in the service system.

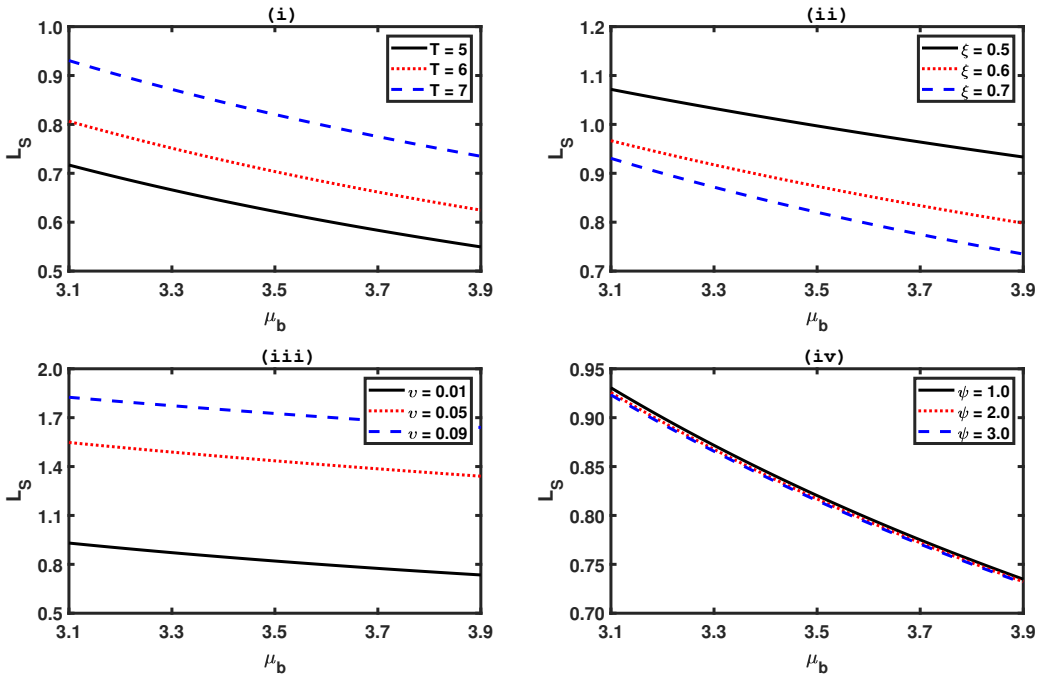


Figure 6.2: Effect of varied (i) T, (ii) ξ, (iii) υ, and (iv) ψ wrt μ_b on mean number of customers in the service system.

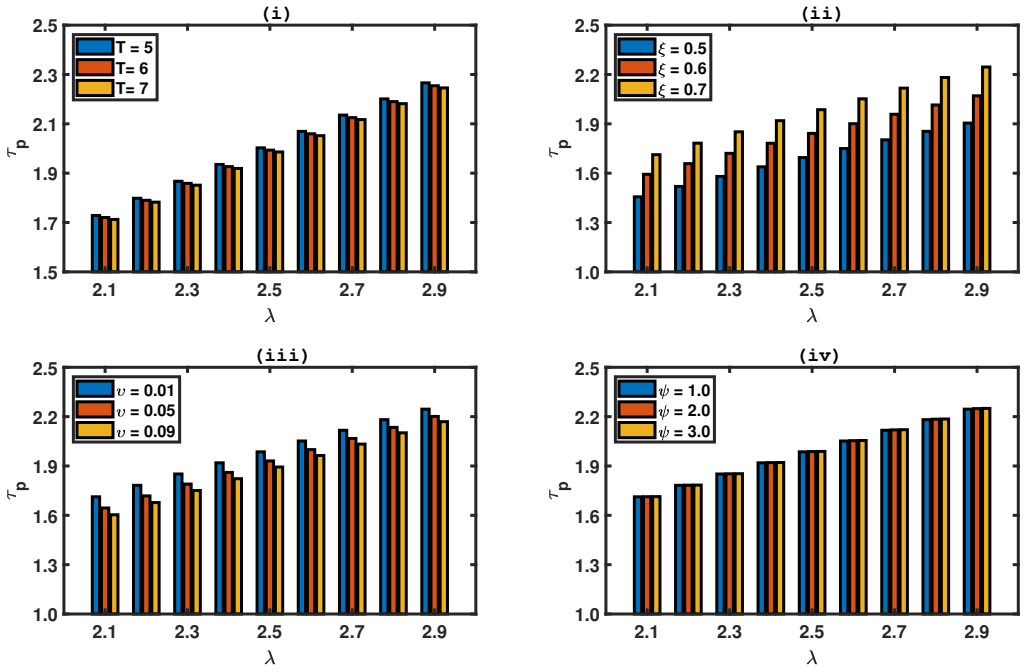


Figure 6.3: Effect of varied (i) T, (ii) ξ, (iii) υ, and (iv) ψ wrt λ on the throughput of the service system.

fixed value of λ, an increasing trend is observed for the higher values of T, ξ, and υ as in Fig. 6.1. Nevertheless, at the same time, the reverse trend is observed in the case of ψ. Similarly, in Fig. 6.2-(iv), it is observed that for the definite values of μ_b and increasing ψ, L_S is decreasing. It is apparent from the fact that as the pressure factor increases, the active servers' service rate increases, which results in a decreasing trend in L_S.

The influence of system parameters λ and μ_b on the throughput (τ_p) of the service system is depicted in Figs. 6.3 and 6.4, respectively, as bar graphs. These figures provide a better and more important understanding to the system analysts on distinguishing the variations of throughput of the service system wrt to various system parameters value. Throughput gives the mean number of customers served by the server either in normal mode or partial breakdown state; subsequently, it increases when the number of arrivals in the service system increases and the service rate increases. The trend is expected since there is more likelihood of customers. The parameter ξ positively affects throughput, as shown in Figs. 6.3(ii) & 6.4(ii), whereas T and υ negatively that can be observed in Figs. 6.3(i) & (iii) and Figs. 6.4(i) & (iii). Moreover, τ_p is the least sensitive wrt ψ, which results in a minor change with higher values of ψ, as presented in Figs. 6.3(iv) & 6.4(iv).

Besides the earlier fixed default value of system parameters, the default values of several cost elements are also considered as $C_h = 5$, $C_d = 60$, $C_b = 250$, $C_i = 170$; $C_{\mu_b} = 2$, $C_{\mu_d} = 17$,

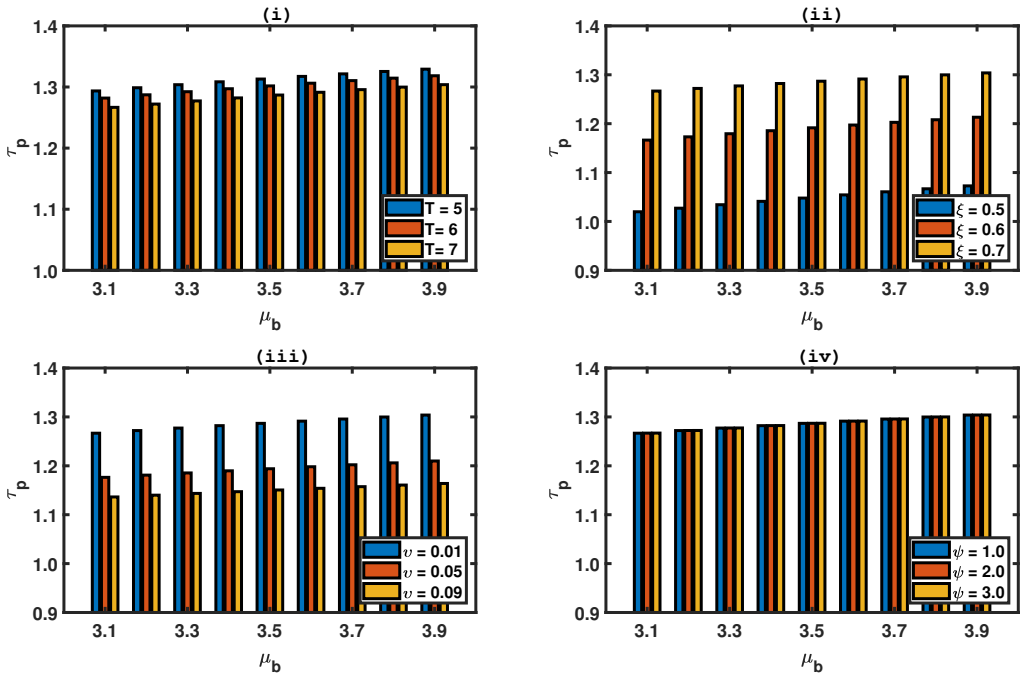


Figure 6.4: Effect of varied (i) T, (ii) ξ, (iii) υ, and (iv) ψ wrt μ_b on the throughput of the service system.

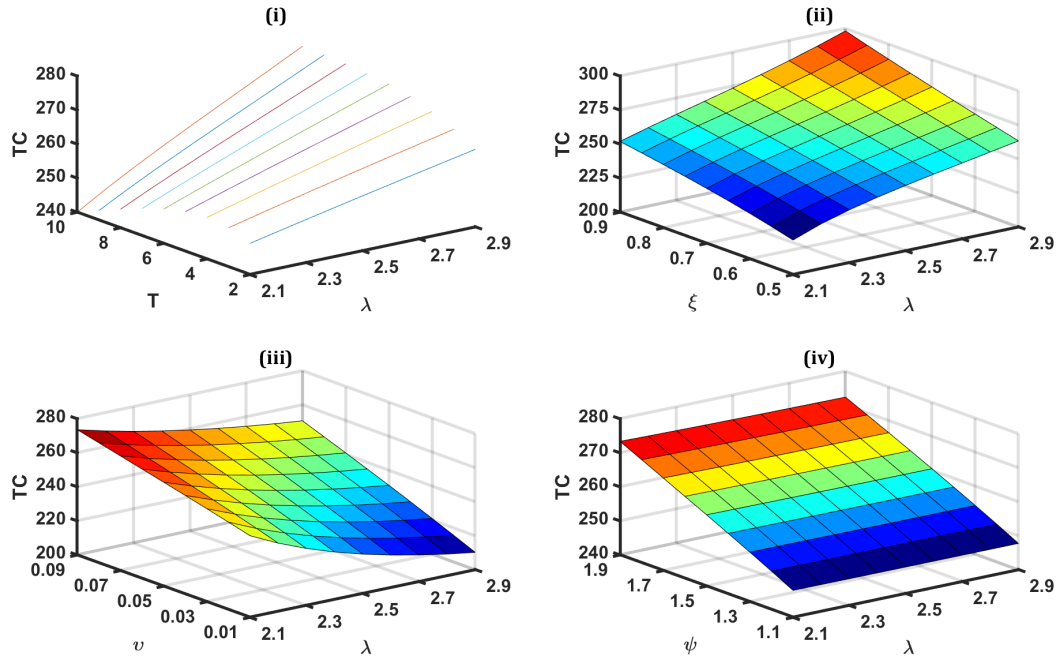


Figure 6.5: Mean cost (TC) wrt varied (i) (T,λ), (ii) (ξ,λ), (iii) (λ,υ), and (iv) (ψ,λ).

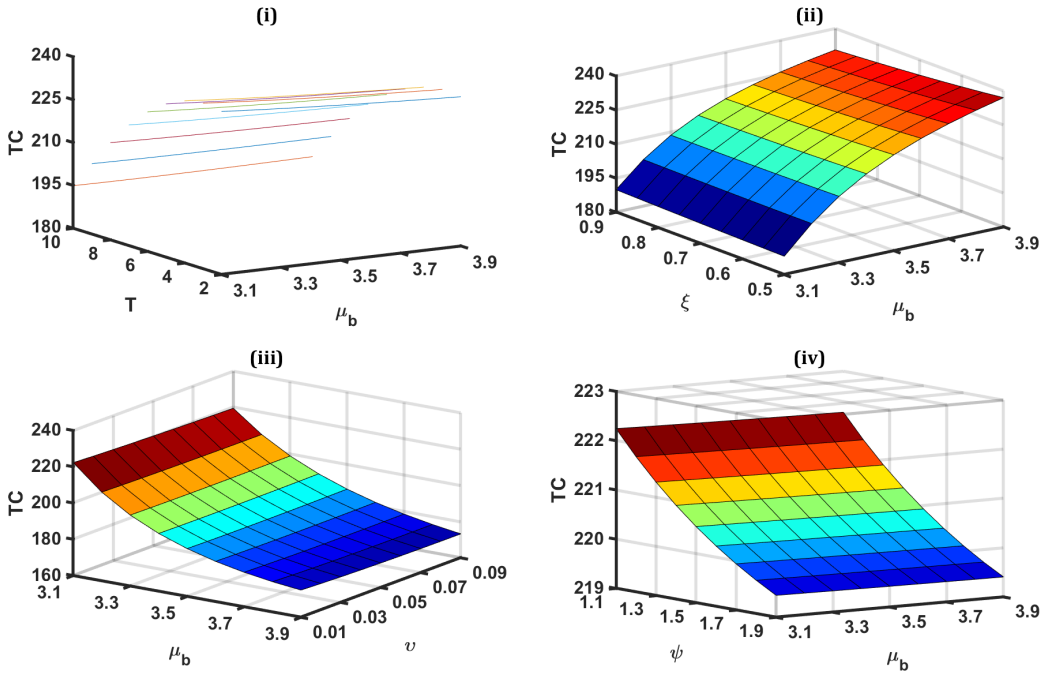


Figure 6.6: Mean cost (TC) wrt varied (i) (T, μ_b), (ii) (ξ, μ_b), (iii) (μ_b, υ), and (iv) (μ_b, ψ).

and $C_w = 100$ to analyze studied service system economically. For the various combinations of default parameters, Figs. 6.5 and 6.6 depict the variation in the value of the mean cost (TC) of the system, given in eq.^{n} (6.30). Fig. 6.5(i) characterizes the variation on TC for increasing values of T and λ, revealing that the mean cost (TC) enhances as intuitively expected. From Figs. 6.5(ii) & (iv), we notice that for higher values of combinations (λ, ξ) and (λ, ψ), the mean cost TC is deduced rapidly in comparison to Fig. 6.5(iii). Correspondingly, TC significantly raises with the higher values of parameters μ_b and T as in Fig. 6.6(i). In Fig. 6.6(ii), it is noticeable that, first, the TC increases more rapidly wrt positively varied (ξ, μ_b) and remains almost constant later. Similar findings are exhibited for the remaining figures as well. Therefore, all of these statistics incite that the default parametric values used here are praiseworthy in decision making, planning, and designing the service system, which plays a significant role in the development of the governing model.

From the results provided in the above Figs. 6.1–6.6, it is perceived that there is a strategic need to estimate the optimal operating policy to minimize the mean cost incurred in the service system. Generally, it is highly typical to evaluate the analytical and closed-form of μ_b^* and μ_d^*, because of the high order complexity and non-linearity involved in the cost optimization problem. The trend for incurred TC wrt to the system design parameters μ_b and μ_d respectively, have been calculated numerically with the help of Fig. 6.7–6.9. In this

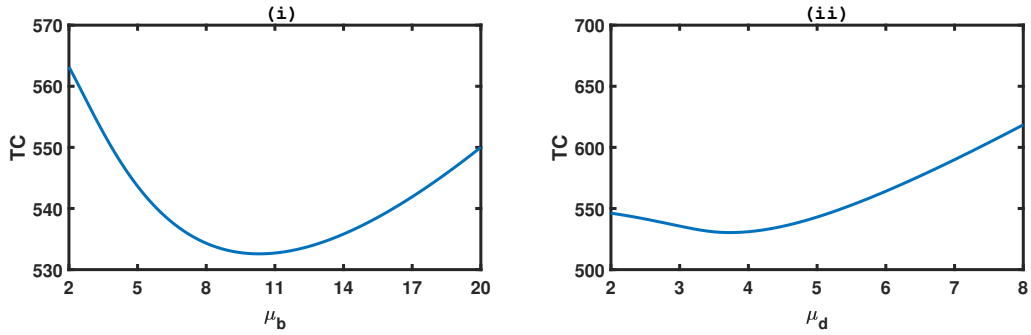


Figure 6.7: Mean cost (TC) wrt decision variables μ_b and μ_d.

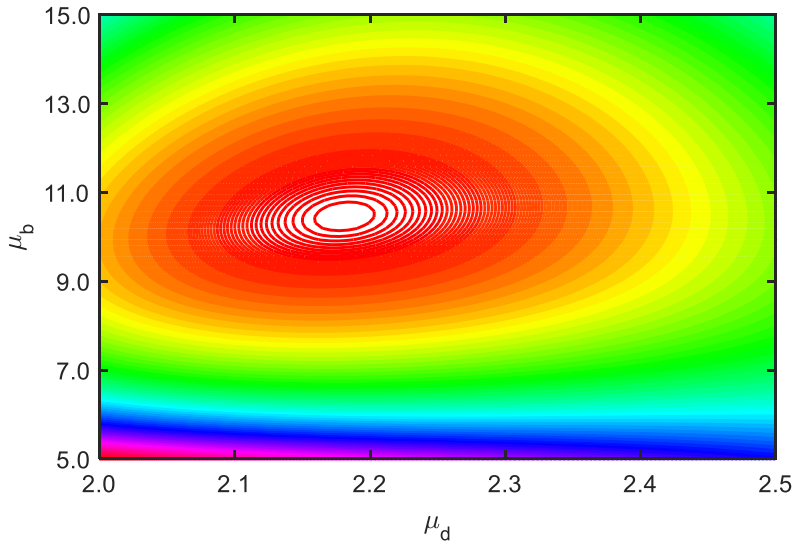


Figure 6.8: Contour plot for mean cost (TC) wrt varied μ_b and μ_d.

context, the values of different default system parameters and performance associated unit cost, are considered as follows: $K = 20$; $T = 10$; $\lambda = 4.0$, $\xi = 0.3$, $\psi = 1.0$, $\upsilon = 0.2$, $\vartheta = 3.0$, $C_h = 130$, $C_d = 60$, $C_b = 100$, $C_i = 350$, $C_{\mu_b} = 5$, $C_{\mu_d} = 35$, and $C_w = 100$. The lower and upper limits of the decision/system design parameters μ_b and μ_d are taken as $[2\,20]$ and $[1\,7]$ respectively. From Fig. 6.7, the conclusion be inferred that the mean cost $TC(\mu_b, \mu_d)$ is convex in nature as intuitively anticipated.

To calculate the optimal combinations of the design decision parameters μ_b and μ_d, the nature-inspired optimization technique: PSO and CS algorithm are utilized. The results are compared with the results of the quasi-Newton method. The results delineated in Figs. 6.8–6.10 infer the convex nature of cost function wrt to decision parameters. Several generations

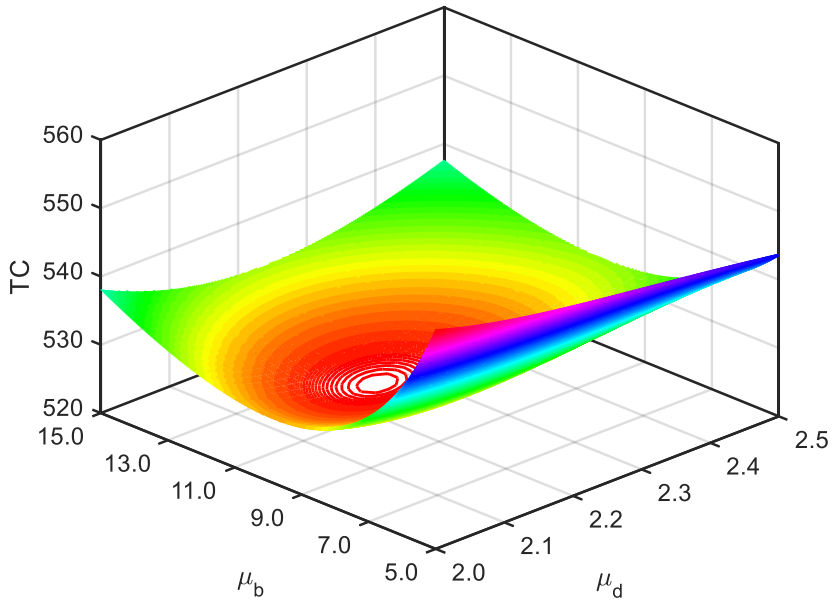


Figure 6.9: Three dimensional contour plot for mean cost (TC) wrt varied μ_b and μ_d.

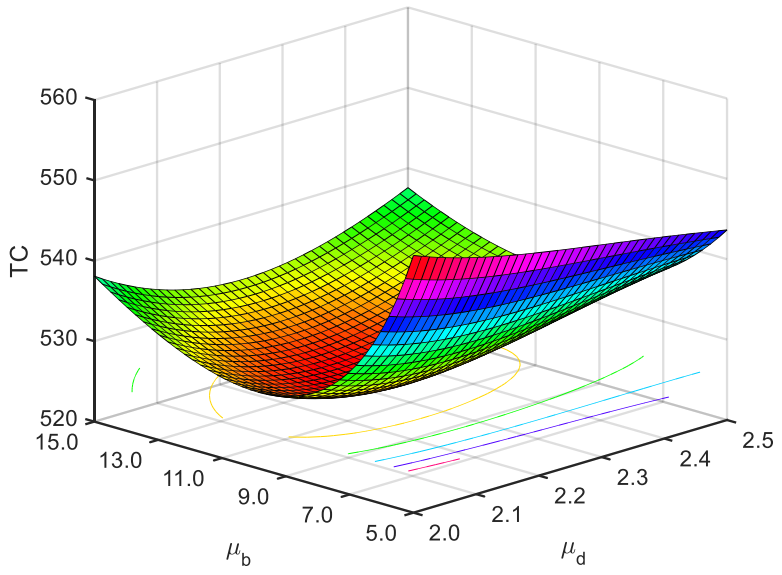


Figure 6.10: Surface plot for the mean cost (TC) wrt varied (μ_b, μ_d).

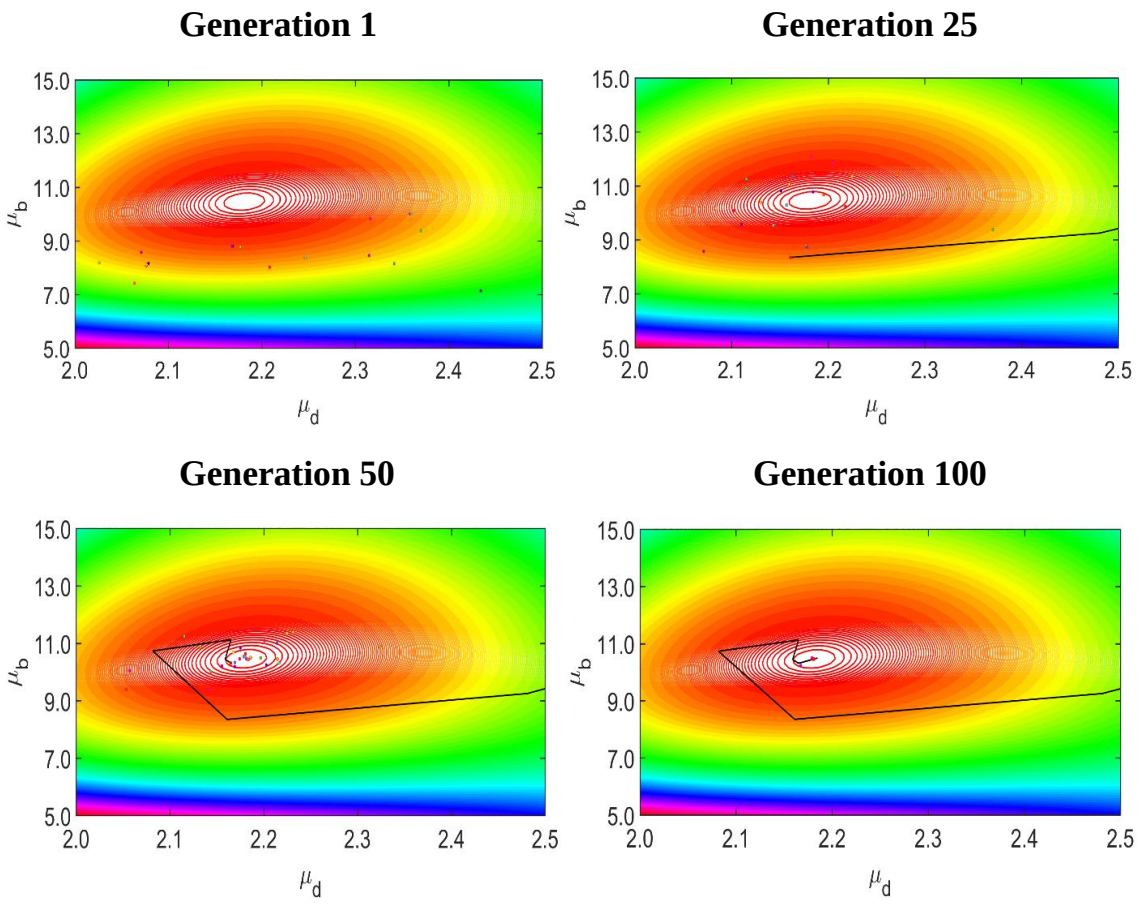


Figure 6.11: PSO algorithm's different generations.

of the PSO algorithm have also been depicted in Fig 6.11 to display the robustness and working nature of the PSO algorithm. These results show that the mean cost of the service system wrt combined values of continuous system design parameters, μ_b and μ_d, is optimal, and the used algorithm plays an essential role in providing converging results.

Table 6.1: Iterations of QN method in finding the optimal values of μ_b and μ_d.

Iterations	μ_d	μ_b	$TC(\mu_b, \mu_d)$
0	3.000000	12.000000	549.252374
1	2.005001	11.651540	530.572348
2	2.307443	10.644663	527.526168
3	2.212510	10.519532	526.369986
4	2.175001	10.475921	526.289346
5	2.181594	10.460087	526.285852
6	2.181004	10.457057	526.285811
7	2.180910	10.456292	526.285810
8	2.180910	10.456295	526.285810
9	2.180910	10.456296	526.285810
10	2.180910	10.456297	526.285810
11	**2.180910**	**10.456297**	**526.285810**

Next, we also provide numerous simulations wrt several combinations of system parameters to validate the converging results and the convexity of the formulated cost function (6.30) in Tables 6.1–6.5. We have incorporated the semi-classical optimizer: QN method and meta-heuristics like PSO and CS algorithm. Because the PSO algorithm does not involve the computation of gradients, it is an appropriate technique to calculate the optimum of single/multi-modal optimization problems. The advantage of the meta-heuristics like PSO and CS algorithms is that these can be employed to examine the optimal values of decision variables whether discrete or continuous. The parametric values of the system components are taken as the same as in the previous simulation to demonstrate the converging results. The PSO algorithm pertinent parameters are fixed as $c_1 = 2$, $c_2 = 2$ and $\Omega = 0:5$. We conventionally fix the lower and upper bounds for μ_b and μ_d as $[5.0\ 15.0]$ and $[2.0\ 5.0]$ respectively, and obtained the optimal operating decision parameters in Table 6.4 up to the tenth place of decimal. The numerical results in Table 6.4 are depicted by considering 20 independent runs with 100 generations in each run and 50 particles generated randomly for each PSO simulation. For the validity purpose, we have also used the notion of statistical

Table 6.2: Optimal values of μ_b and μ_d with optimal mean cost (TC^*) using QN method.

$(\lambda, \xi, \upsilon, \vartheta)$	(3.8,0.3,0.2,3.0)	(4.0,0.3,0.2,3.0)	(4.2,0.3,0.2,3.0)	(4.0,0.4,0.2,3.0)	(4.0,0.5,0.2,3.0)
(μ_d^0, μ_b^0)	(3,12)	(3,12)	(3,12)	(3,12)	(3,12)
Total Iterations	10	11	14	12	15
μ_d^*	2.062034	2.18090	2.300244	2.982308	3.808015
μ_b^*	10.642130	10.456297	10.278914	9.384290	8.486856
$TC(\mu_d^*, \mu_b^*)$	524.706586	526.28581	528.034607	540.489366	558.81171
$\frac{\partial TC}{\partial \mu_d}$	-0.957199	-0.707379	-0.695862	-5.273421	-9.221684
$\frac{\partial TC}{\partial \mu_b}$	-0.264034	-0.421598	-0.565635	-0.959683	-1.510192

Table 6.3: Optimal values of μ_b and μ_d with optimal mean cost (TC^*) using QN method.

$(\lambda, \xi, \upsilon, \vartheta)$	(4.0,0.3,0.1,3.0)	(4.0,0.3,0.15,3.0)	(4.0,0.3,0.1,4.0)	(4.0,0.3,0.1,3.5)	(4.0,0.3,0.1,2.5)
(μ_d^0, μ_b^0)	(3,12)	(3,12)	(3,12)	(3,12)	(3,12)
Total Iterations	14	12	11	13	14
μ_d^*	2.246358	2.205677	2.254851	2.251055	2.240368
μ_b^*	8.623722	9.647259	8.539841	8.576771	8.685731
$TC(\mu_d^*, \mu_b^*)$	497.738489	513.662107	495.710162	496.603728	499.234312
$\frac{\partial TC}{\partial \mu_d}$	2.743551	1.027515	2.787818	2.771251	2.695925
$\frac{\partial TC}{\partial \mu_b}$	-0.565639	-0.513366	-0.571368	-0.569079	-0.560221

Table 6.4: Optimal values of μ_b^* and μ_d^* with minimal mean cost (TC^*) using PSO algorithm.

$(K,T,\lambda,\xi,\upsilon,\vartheta)$	μ_d^*	μ_b^*	$TC^*(\mu_b^*,\mu_d^*)$	**mean** $\left\{\frac{TC_i}{TC^*}\right\}$	**max** $\left\{\frac{TC_i}{TC^*}\right\}$	**CPU time**
(20, 10, 4.0, 0.3, 0.2, 3.0)	2.180950	10.456491	526.285810	1.0000000020	1.0000000046	294.88
(25, 10, 4.0, 0.3, 0.2, 3.0)	2.205506	10.035226	517.335965	1.0000000033	1.0000000093	475.34
(30, 10, 4.0, 0.3, 0.2, 3.0)	2.226941	9.698844	510.423085	1.0000000115	1.0000000332	438.36
(20, 8, 4.0, 0.3, 0.2, 3.0)	2.368139	7.890944	471.088229	1.0000000090	1.0000000021	290.35
(20, 10, 3.8, 0.3, 0.2, 3.0)	2.062034	10.642108	524.706586	1.0000000005	1.0000000014	288.39
(20, 10, 4.2, 0.3, 0.2, 3.0)	2.300265	10.278137	528.034607	1.0000000041	1.0000000117	287.64
(20, 10, 4.0, 0.4, 0.2, 3.0)	2.982303	9.384467	540.489366	1.0000000177	1.0000000486	287.64
(20, 10, 4.0, 0.5, 0.2, 3.0)	3.808010	8.486571	558.811711	1.0000000071	1.0000000209	288.38
(20, 10, 4.0, 0.3, 0.10, 3.0)	2.246466	8.622392	497.738492	1.0000000056	1.0000000147	287.85
(20, 10, 4.0, 0.3, 0.15, 3.0)	2.205811	9.646706	513.662109	1.0000000007	1.0000000014	287.74
(20, 10, 4.0, 0.3, 0.1, 2.5)	2.240297	8.685781	499.234313	1.0000000029	1.0000000072	287.47
(20, 10, 4.0, 0.3, 0.1, 3.5)	2.251017	8.578444	496.603730	1.0000000041	1.0000000069	287.44
(20, 10, 4.0, 0.3, 0.1, 4.0)	2.254923	8.538948	495.710164	1.0000000026	1.0000000056	287.26

Table 6.5: Optimal values of μ_b^* and μ_d^* with minimal mean cost (TC^*) using CS algorithm.

$(K,T,\lambda,\xi,\upsilon,\vartheta)$	μ_d^*	μ_b^*	$TC^*(\mu_b^*,\mu_d^*)$	**mean** $\left\{\frac{TC_i}{TC^*}\right\}$	**max** $\left\{\frac{TC_i}{TC^*}\right\}$	**CPU time**
(20, 10, 4.0, 0.3, 0.2, 3.0)	2.180949	10.456489	526.285810	1.0000000139	1.0000000251	320.71
(25, 10, 4.0, 0.3, 0.2, 3.0)	2.205506	10.035226	517.335965	1.0000000105	1.0000000263	512.84
(30, 10, 4.0, 0.3, 0.2, 3.0)	2.226941	9.698844	510.423086	1.0000000451	1.0000000659	649.73
(20, 8, 4.0, 0.3, 0.2, 3.0)	2.368204	7.890913	471.088229	1.0000000223	1.0000000247	329.51
(20, 10, 3.8, 0.3, 0.2, 3.0)	2.062093	10.642086	524.706586	1.0000000045	1.0000000074	338.13
(20, 10, 4.2, 0.3, 0.2, 3.0)	2.300295	10.278109	528.034608	1.0000000087	1.0000000135	350.62
(20, 10, 4.0, 0.4, 0.2, 3.0)	2.982291	9.384449	540.489367	1.0000000197	1.0000000502	309.67
(20, 10, 4.0, 0.5, 0.2, 3.0)	3.808075	8.486598	558.811711	1.0000000098	1.0000000179	310.88
(20, 10, 4.0, 0.3, 0.1, 3.0)	2.246501	8.622378	497.738493	1.0000000129	1.0000000213	325.11
(20, 10, 4.0, 0.3, 0.15, 3.0)	2.205852	9.646717	513.662111	1.0000000012	1.0000000025	314.54
(20, 10, 4.0, 0.3, 0.1, 2.5)	2.240315	8.685793	499.234313	1.0000000054	1.0000000096	299.38
(20, 10, 4.0, 0.3, 0.1, 3.5)	2.251009	8.578429	496.603730	1.0000000076	1.0000000104	305.15
(20, 10, 4.0, 0.3, 0.1, 4.0)	2.254927	8.538941	495.710164	1.0000000045	1.0000000062	307.69

characteristics: mean-ratio and maximum-ratio of the optimal mean cost for all independent runs, to show the robustness of the proposed PSO algorithm.

For a better understanding of the research findings, a comparative study between the QN, PSO, and CS algorithm is accomplished for several combinations of system parameters in Tables 6.1-6.5. The computation time among all the iterations and optimal results are fundamental aspects for comparing the efficacy and effectiveness of an algorithm. So inspired by this, we have used both in each table with each combination of system parameters. It is observed that the calculated optimal values of design parameters and mean cost by the proposed algorithms QN, PSO, and CS algorithm are almost equivalent. The CPU time (in seconds) for the PSO algorithm is slightly less than the CS algorithm in each iteration. The associated mean cost enlisted by the PSO algorithm meets the optimality considerably and efficiently for all considered test instances. The results of the PSO algorithm are also superior to the QN method, for each numerical example. Newton's method involves the computation of gradient for calculating the Hessian. We obtain gradient numerically due to the high non-linearity and complexity of the optimization problem. It includes the high-scale estimation, which minimizes the efficacy of the algorithm.

From the above examples/numerical experiments and deliberations, we can say that the PSO algorithm effectively gives optimal results compared to the CS algorithm and the quasi-Newton method. It is also noticed that optimum setup of system design parameters is essential to reduce the mean cost required in rendering service to the potential customers.

6.9 Conclusion

The uniqueness of the current work is to observe the effects of several queueing characteristics, viz customer impatience, threshold recovery policy, and partial server breakdown under the pressure condition, on the operational capability and performance of the service system. The Chapman-Kolmogorov differential-difference equations have been provided for modeling purposes. The steady-state probability distribution has been demonstrated using the repeated substitution approach. Further, to show the quality performance of the service system, numerous system performance indices have been provided. The function for the mean cost and associated cost optimization problem has been provided for the economic analysis. The nature-inspired optimizer, PSO, and CS algorithm has been used for the numerical illustration of cost analysis. Moreover, the comparative analysis between the semi-classical optimizer: QN method and meta-heuristics optimization techniques CS algorithm, PSO algorithm has also been performed to depict the optimal operating combination (*i.e.*, optimal service rates μ_b^* and μ_d^*) with the minimal mean cost TC^* of the service system.

Chapter 7

Transient Analysis of Queueing Based Congestion with Differentiated-Vacations and Customer's Impatience Attributes

This chapter studies the critical issue of the single-server congestion problem with prominent customer impatience attributes and server strategic differentiated vacation. Despite their apparent practical relevance, the proposed congestion problem has yet to be studied from a service/production perspective with transient analysis.

7.1 Introduction

The optimal service system emphasizes strategic congestion management to address the customer's traffic. Congestion management is an association between planning and operations. The research study's prime objective is to present a systematic process for managing customer congestion and provides critical information on the performance of the service system. The investigation identifies alternative strategies for alleviating congestion and enhancing customers' mobility to levels that attain a state of intended service. At the core, congestion management includes performance monitoring, alternative strategies for congestion, and norms for detecting when action is required. Studying congestion and its causes is used to develop more efficient and cost-effective services and systems. The critical goal of the studied service system is to prioritize strategies that would be most effective for congestion management. The queueing analysis is one of the most effective and practical mathematical tools for understanding and aiding decision-making in dealing with critical resources and managing congestion. The queueing theory aims to design efficient systems that render service competently to customers with minimum delay but do not cost too much to be sustainable. Several queueing systems representing different service designs, regimes, and strategies, wherein the common feature is that customers arrive randomly at facilities to get service, need to investigate.

The queueing problems with customer impatience attribute and service provider strategic vacation interest many researchers in neoteric times due to their broad applicability in real-time congestion. Server vacation may occur for several reasons, including a low workload, maintenance time, the failure to repair, and many more. In recent years, there has been considerable research on customer impatience attributes in queueing systems with strategic server vacations/failures. Levy and Yechiali[118] were the ones who introduced the server vacation policy initially. A thorough, excellent, and exhaustive study of vacation queueing models is found in Doshi's survey [44], as well as in several publications on vacation queueing models (*cf.* [181], [184], [8]).

The strategic vacation policy variants include multi-vacation, single-vacation, working vacation, Bernoulli vacation, gated vacation, N-policy etc. The server goes on vacation mode if found no waiting customer for service instead of continuing in an idle state and increasing the service cost. In a single-vacation policy, when the server returns from vacation, it serves any waiting customers in the system; otherwise, it stays idle. The server immediately takes another vacation in the multiple vacation policy when it resumes from vacation and discovers no waiting customer in the system. In a working vacation policy, the server remotely offers the service at a slower rate instead of terminating it or removing itself from

the system. In $N-policy$, the server remains on vacation until there is an accumulation of N customers. In the present study, we propose the multiple-vacation-based differentiated vacation queueing systems that are widely used strategies to control access to the service facility and simulate many energy-saving modes, such as wireless communications, flexible manufacturing systems, etc. Isijola *et al.* [87] studied the variant of multiple vacations wherein two sorts of vacations, each with a different random duration, are analyzed. Vijayashree and Janani [191] analyzed the single server queueing system incorporating differentiated-vacations policy and obtained the transient probability using modified Bessel function and Laplace transform techniques.

Kempa and Marjasz [102] derived the conditional probability distribution analytically for the queue size in a limited-buffer single-channel $M/G/1/N$ queueing model with batch arrivals operating under the multiple vacation policy. They calculated the time to a first buffer overflow employing Korolyuk's potential, integral equations, and embedded Markov chain notions. Ayyappan and Deepa [17] analyzed a non-Markovian batch arrival bulk service $M^{[X]}/G(a,b)/1$ queueing system featuring multiple vacation policies, service interruption & setup time with N-policy. In recent years many researchers (cf [66], [182], [16], [167], [112]) opted for the multiple vacation policy & several types of methodologies to analyze the performance characteristics and provided several numerical illustrations.

In everyday life, numerous queueing circumstances happen, and a long queue may deter customers. As a response, customers either elect not to join the line (i.e., balk) or leave after waiting due to impatience (i.e., renege). The dissatisfaction level of customers increases due to long waiting for service and deciding to leave the system without getting served at random times. Haight [72] first conceptualized customers' balking attribute in a single server queueing model. Later, Haight [74] again proposed the reneging attribute of customers for the $M/M/1$ queueing model. Many service systems originating in real-world applications may have intermittently inaccessible servers, impacting a customer's sojourn duration and willingness to join. Naor [140] pioneered the research of queueing systems concerned with customers' reluctance behavior from an economic perspective. The economic assessment of customer balking behavior is significant. Indeed, the approach and findings are also more important (*cf.*[105], [48], [25]). The decision of a waiting customer to stay or renege is continually offered till his departure from the system. The waiting time before reneging depends on the service type [27]. For instance, if a customer is waiting for a mode of transportation and an unexpected event occurs that might cause more delays, the customer may opt to renege and utilize one of the available alternative service options instead (*cf.* [52], [60]). Al-Seedy et al. [9] provided a technique for evaluating transient probabilities of the

queueing model $M/M/c$ incorporating reneging and balking. Hassin [79] presumed to renege as a crucial component for the realistic modeling of customers' strategic behavior in queueing models involving vacations. Due to their adaptability and applicability, these models with impatient customers have been thoroughly evaluated (*cf.* [135], [80], [130], [28], [162]). Customers' impatience attributes are comprehended as a possible loss of customers, resulting in a loss of total income owing to their insurmountable influence on a system's intended financial situation from a cost perspective.

Kumar [114] is the first researcher who introduced the efficient notion of retention of the reneging customer. Later, many researchers (*cf.* [108], [177], [117], [23], [115]) investigated retention of the reneging customer in the service sector in economic perspective. Bouchentouf and Guendouzi [24] studied the $M^X/M/C$ queueing model, including muti-working vacation variants in modeling and computed the steady-state solution and henceforth performance measure for economic analysis using the probability generating function (*PGF*).

To the best of our surveys, no studies have been undertaken on customers' impatience attributes: balking and reneging in queueing systems with differentiated-multiple vacations. The research gap makes a broader platform for our study. Customers may opt to be reluctant to service when a server goes on vacation and system congestion grows. In comparison to earlier research, the importance of our analysis is that we concentrate on the impact of balking and reneging options in systems with differentiated-multiple vacations.

The structure of the remaining chapter is organized in the following order. We describe the proposed queueing-based congestion model along with its states and notations in Section 7.2. Section 7.3 explains the proposed methodology: modified Bessel function and generating function. In Section 7.4, we discuss the transient analysis employing the Laplace transformation and derive the state probabilities of the studied model. The system's performance measures are derived in Section 7.5 with the help of transient probabilities computed in the previous Section. The following Section 7.6 contains different experimental results, numerical findings, and significant qualitative insights. In the end, we conclude and offer potential study prospects for the future in Section 7.7.

7.2 Problem Statement and Associated Equations

In this chapter, we have considered a single-server queueing system with the following assumptions and notations:

- The customers are generated randomly from the population of prospective customers of size infinite.

Table 7.1: List of parameters used

Notation	Description
λ	the arrival rate of the customers
ξ	the joining probability
ν	the reneging rate
μ	the service rate
θ_1	the type-1 vacation parameter
θ_2	the type-2 vacation parameter
$N(t)$	number of customers in the system at time t
$J(t)$	the state of the service provider at time t
$\pi_{n,j}$	the probability of n customers in the system and service provider in state j
$m(t)$	expected number of the customers in the system at time t
$V(t)$	variance of the number of the customers in the system at time t

- The inter-time between arrivals of customers for the intended service in the system is assumed exponentially with mean arrival rate λ.
- Upon arrival, the prospective customer gets the intended service immediately if the service provider is idle; otherwise, the customer joins the queue and waits for service.
- The customer may be impatient at the arrival epoch if the server is on vacation or busy. Each arrived customer may decide whether to join or balk the system with probability ξ or complementary probability $1-\xi$, respectively.
- After waiting for some subsequent time interval, the customer may renege from the system. The random waiting time before reneging is exponentially distributed with a mean time of $1/\nu$.
- There is one reliable server to serve the customer waiting in the system with finite capacity.
- The waiting customer is chosen for service following first-come-first-serve (*FCFS*) queue discipline.
- The continuous random variable, time-to-serve a customer, follows exponential distribution (memoryless distribution) with parameter μ.
- Under the strategic policy, we assume that there are two types of vacations: type-1 vacation and type-2 vacation.
- The type-1 vacation is initiated after a nonzero-length busy period and is independent of the busy period. The vacation time for type-1 is exponentially distributed with parameter θ_1.

- The type-2 vacation is initiated when no customer is queued for the service when the service provider returns from vacation. The duration of type-2 vacation follows an exponential distribution with parameter θ_2.

All events' arrival/service, balking/reneging, and vacation are independent of each other.

Let $(N(t), J(t))$ define a two-tuple continuous-time Markov chain (*CTMC*) with two-dimensional state space $S = \{(n, j) : n = 0, 1, 2, \ldots \& \, j = 0, 1, 2\}$, where

$$N(t) \equiv \text{number of customers present in the system at instant } t$$
$$J(t) \equiv \text{state of the service provider (}SP\text{) at instant } t$$

where

$$J(t) = \begin{cases} 0; & \text{the SP is in active busy mode at instant } t \\ 1; & \text{the SP is on a type-1 vacation at instant } t \\ 2; & \text{the SP is on a type-2 vacation at instant } t \end{cases}$$

For modeling purposes, we define the joint probability distribution as

$$\pi_{n,j}(t) = Prob[N(t) = n, J(t) = j]; (n, j) \in S$$

The Chapman-Kolmogorov differential-difference equations for the studied model are derived using the assumptions and notations stated above. We start the analysis with the formation of equations for rate of change of joint probabilities $\pi_{n,j}; \forall n, j$ (state probabilites) for different states by balancing the inflow-outflow rates, i.e., outflow rate with negative sign and inflow with positive sign along with state probabilities.

$$\frac{d\pi_{1,0}(t)}{dt} = -(\lambda\xi + \mu)\pi_{1,0}(t) + \theta_1\pi_{1,1}(t) + \theta_2\pi_{1,2}(t) + (\mu + \nu)\pi_{2,0}(t) \tag{7.1}$$

$$\frac{d\pi_{n,0}(t)}{dt} = -(\lambda\xi + \mu + (n-1)\nu)\pi_{n,0}(t) + \lambda\xi\pi_{n-1,0}(t) + \theta_1\pi_{n,1}(t) \tag{7.2}$$
$$+ \theta_2\pi_{n,2}(t) + (\mu + n\nu)\pi_{n+1,0}(t); \quad n = 2, 3, 4, \ldots$$

$$\frac{d\pi_{0,1}(t)}{dt} = -(\lambda + \theta_1)\pi_{0,1}(t) + \mu\pi_{1,0}(t) \tag{7.3}$$

$$\frac{d\pi_{1,1}(t)}{dt} = -(\lambda\xi + \theta_1)\pi_{1,1}(t) + \lambda\pi_{0,1}(t) \tag{7.4}$$

$$\frac{d\pi_{n,1}(t)}{dt} = -(\lambda\xi + \theta_1)\pi_{n,1}(t) + \lambda\xi\pi_{n-1,1}(t); \quad n = 2, 3, 4, \ldots \tag{7.5}$$

$$\frac{d\pi_{0,2}(t)}{dt} = -\lambda\pi_{0,2}(t) + \theta_1\pi_{0,1}(t) \tag{7.6}$$

$$\frac{d\pi_{1,2}(t)}{dt} = -(\lambda\xi + \theta_2)\pi_{1,2}(t) + \lambda\pi_{0,2}(t) \tag{7.7}$$

$$\frac{d\pi_{n,2}(t)}{dt} = -(\lambda\xi + \theta_2)\pi_{n,2}(t) + \lambda\xi\pi_{n-1,2}(t); \quad n = 2, 3, 4, \ldots \tag{7.8}$$

The system of differential-difference equation (7.1)-(7.8) dependent to the initial conditions

$$\pi_{n,j}(0) = \begin{cases} 1; n = 0, j = 1 \\ 0; \text{otherwise} \end{cases}$$

are solved to obtain state probabilities employing mathematical notions of hypergeometric Laplace transform, modified Bessel's function, generating function in the forthcoming section.

7.3 Mathematical Preliminaries

This section introduces some basic principles of modified Bessel functions and generating functions that the fellow researcher will need to comprehend this chapter better.

7.3.1 Modified Bessel Function

Bessel's modified equation is given by

$$t^2\frac{dy}{dt} + t\frac{dy}{dt} - (t^2 + r^2)y(t) = 0, \quad r \geq 0$$

The solution of the above equation is the first kind of modified Bessel function of order r, indicated by B_r, defined as

$$B_r(t) = \sum_{m=0}^{\infty} \frac{(t/2)^{2m+r}}{l!\Gamma(m+r+1)}, r > 0$$

In particular, $B_r(t) = B_{-r}(t)$ for $r \geq 0$.

7.3.2 Generating Function

The following is a definition of a generating function $G(z,t)$ in powers of t for a collection of functions $\{f_m(z)\}$.

$$G(z,t) = \sum_{m=1}^{\infty} c_m f_m(z) t^m \tag{7.9}$$

where c_m is a parameter coefficient function of m of the set $\{f_m(z)\}$ and independent to z and t. The symbol $\{f_m(z)\}$ is used to indicate the infinite set $\{f_0(z), f_1(z), \ldots, f_m(z), \ldots\}$. If $f_m(z)$ is also defined for negative, function $H(z,t)$ having a Laurent series expansion is of the form

$$H(z,t) = \sum_{-\infty}^{\infty} c_m f_m(z) t^m \tag{7.10}$$

If $f_m(z)$ is the point probability function of a *drv* z, then the generating function is called a probability generating function (*cf.* [132], [95]).

7.4 Transient Analysis

Using pre-stated mathematical notions of the Bessel function and generating function, we obtain the explicit formula for time-dependent queue-size distribution for the studied queueing-based congestion system in this section. We employ the following sequel for this purpose.

7.4.1 Laplace Transform

The following is the definition of the Laplace transform L of state probabilities $\pi_{n,j} \forall n, j$ and corresponding derivatives

$$\pi^*_{n,j}(s) = \mathrm{L}\left(\pi_{n,j}(t)\right) = \int_0^\infty e^{-st}\pi_{n,j}(t)dt; \quad \forall\, n, j \;\; \& \;\; s \in \mathbb{C}$$

$$\mathrm{L}\left(\frac{d\pi_{n,j}(t)}{dt}\right) = s\,\pi^*_{n,j}(s) - \pi_{n,j}(0); \quad \forall\, n, j$$

The system of differential-difference equations from $eq^n(7.1)$ to $eq^n(7.8)$ is converted as system of linear equations from $eq^n(7.11)$ to $eq^n(7.18)$ on applying pre-defined Laplace transform as follows

$$s\pi^*_{1,0}(s) - \pi_{1,0}(0) = -(\lambda\xi + \mu)\,\pi^*_{1,0}(s) + \theta_1\pi^*_{1,1}(s) + \theta_2\pi^*_{1,2}(s) + (\mu+\nu)\,\pi^*_{2,0}(s) \tag{7.11}$$

$$\begin{aligned} s\pi^*_{n,0}(s) - \pi_{n,0}(0) = &-(\lambda\xi + \mu + (n-1)\nu)\,\pi^*_{n,0}(s) + \lambda\xi\,\pi^*_{n-1,0}(s) + \theta_1\pi^*_{n,1}(s) \\ &+ \theta_2\pi^*_{n,2}(s) + (\mu + n\nu)\,\pi^*_{n+1,0}(s) \quad n = 2,3,4,\ldots \end{aligned} \tag{7.12}$$

$$s\pi^*_{0,1}(s) - \pi_{0,1}(0) = -(\lambda + \theta_1)\,\pi^*_{0,1}(s) + \mu\pi^*_{1,0}(s) \tag{7.13}$$

$$s\pi^*_{1,1}(s) - \pi_{1,1}(0) = -(\lambda\xi + \theta_1)\,\pi^*_{1,1}(s) + \lambda\pi^*_{0,1}(s) \tag{7.14}$$

$$s\pi^*_{n,1}(s) - \pi_{n,1}(0) = -(\lambda\xi + \theta_1)\,\pi^*_{n,1}(s) + \lambda\xi\,\pi^*_{n-1,1}(s); \quad n = 2,3,4,\ldots \tag{7.15}$$

$$s\pi^*_{0,2}(s) - \pi_{0,2}(0) = -\lambda\,\pi^*_{0,2}(s) + \theta_1\pi^*_{0,1}(s) \tag{7.16}$$

$$s\pi^*_{1,2}(s) - \pi_{1,2}(0) = -(\lambda\xi + \theta_2)\,\pi^*_{1,2}(s) + \lambda\pi^*_{0,2}(s) \tag{7.17}$$

$$s\pi^*_{n,2}(s) - \pi_{n,2}(0) = -(\lambda\xi + \theta_2)\,\pi^*_{n,2}(s) + \lambda\xi\,\pi^*_{n-1,2}(s); \quad n = 2,3,4,\ldots \tag{7.18}$$

Analytical solutions, even if approximate, give a straightforward method for decision-makers to estimate congestion and waiting time more quickly. They also typically lower the calculation time of traditional models by introducing better initial parameters into their optimization search space.

On applying initial condition $\pi_{0,1}(0) = 1$, from $eq^n(7.13)$ we have

$$\begin{aligned} s\pi^*_{0,1}(s) &= 1 - (\lambda + \theta_1)\,\pi^*_{0,1}(s) + \mu\pi^*_{1,0}(s) \\ (s + \lambda + \theta_1)\,\pi^*_{0,1}(s) &= 1 + \mu\pi^*_{1,0}(s) \\ \pi^*_{0,1}(s) &= \frac{1}{s+\lambda+\theta_1} + \frac{\mu}{s+\lambda+\theta_1}\pi^*_{1,0}(s) \end{aligned} \tag{7.19}$$

Simillarly on applying initial condition $\pi_{1,1}(0)=0$, from $eq^n(7.14)$ we get

$$s\pi^*_{1,1}(s) = -(\lambda\xi+\theta_1)\pi^*_{1,1}(s)+\lambda\pi^*_{0,1}(s)$$
$$\pi^*_{1,1}(s) = \frac{\lambda}{s+\lambda\xi+\theta_1}\pi^*_{0,1}(s) \tag{7.20}$$

With initial condition $\pi_{n,1}(0)=0; n=2,3,4,\ldots$, $eq^n(7.15)$ gives

$$s\pi^*_{n,1}(s) = -(\lambda\xi+\theta_1)\pi^*_{n,1}(s)+\lambda\xi\pi^*_{n-1,1}(s); \quad n=2,3,4,\ldots$$
$$\pi^*_{n,1}(s) = \frac{\lambda\xi}{s+\lambda\xi+\theta_1}\pi^*_{n-1,1}(s); \quad n=2,3,4,\ldots$$

which recursively yields

$$\pi^*_{n,1}(s) = \left(\frac{\lambda\xi}{s+\lambda\xi+\theta_1}\right)^{n-1}\pi^*_{1,1}(s); \quad n=2,3,4,\ldots$$

Hence, using the $eq^n(7.20)$, we get

$$\pi^*_{n,1}(s) = \left(\frac{\lambda}{s+\lambda\xi+\theta_1}\right)^{n}\xi^{n-1}\pi^*_{0,1}(s); \quad n=1,2,3,\ldots \tag{7.21}$$

We henceforth solve $eq^n(7.21)$ by substituting the value of $\pi^*_{0,1}(s)$ from $eq^n(7.19)$

$$\pi^*_{n,1}(s) = \frac{\lambda^n\xi^{n-1}}{(s+\lambda\xi+\theta_1)^{n+1}} + \frac{\mu\lambda^n\xi^{n-1}}{(s+\lambda+\theta_1)(s+\lambda\xi+\theta_1)^{n+1}}\pi^*_{1,0}(s); \quad n=1,2,3,\ldots \tag{7.22}$$

Since $\pi_{0,2}(0)=0$, the $eq^n(7.16)$ deduce as

$$(s+\lambda)\pi^*_{0,2}(s) = \theta_1\pi^*_{0,1}(s)$$
$$\pi^*_{0,2}(s) = \left(\frac{\theta_1}{s+\lambda}\right)\pi^*_{0,1}(s) \tag{7.23}$$

Hence, from $eq^n(7.19)$ & $eq^n(7.23)$ we get

$$\pi^*_{0,2}(s) = \frac{\theta_1}{(s+\lambda)(s+\lambda+\theta_1)} + \frac{\theta_1\mu}{(s+\lambda)(s+\lambda+\theta_1)}\pi^*_{1,0}(s) \tag{7.24}$$

Using initial condition $\pi_{1,2}(0)=0$, the $eq^n(7.17)$ reduces to

$$\pi^*_{1,2}(s) = \left(\frac{\lambda}{s+\lambda\xi+\theta_2}\right)\pi^*_{0,2}(s) \tag{7.25}$$

Simillarly, under the initial condition $\pi_{n,2}(0)=0$, the $eq^n(7.18)$ reduces as

$$\pi^*_{n,2}(s)=\left(\frac{\lambda\xi}{s+\lambda\xi+\theta_2}\right)\pi^*_{n-1,2}(s);\quad n=2,3,4,\ldots \tag{7.26}$$

which recursively yields

$$\pi^*_{n,2}(s)=\left(\frac{\lambda}{s+\lambda\xi+\theta_2}\right)^n\xi^{n-1}\pi^*_{0,2}(s);\quad n=1,2,3,\ldots \tag{7.27}$$

Using $eq^n(7.24)$ and $eq^n(7.27)$, we have

$$\begin{aligned}\pi^*_{n,2}(s)=&\frac{\theta_1\lambda^n\xi^{n-1}}{(s+\lambda)(s+\lambda+\theta_1)(s+\lambda\xi+\theta_2)^n}\\&+\frac{\theta_1\mu\lambda^n\xi^{n-1}}{(s+\lambda)(s+\lambda+\theta_1)(s+\lambda\xi+\theta_2)^n}\times\pi^*_{1,0}(s);\quad n=0,1,2,\ldots\end{aligned} \tag{7.28}$$

After taking partial fraction and the inverse Laplace transform in $eq^n(7.22)$ and $eq^n(7.28)$, we have

$$\pi_{n,1}(t)=\frac{\lambda^n\xi^{n-1}e^{-(\lambda\xi+\theta_1)t}}{n!}+\mu\lambda^n\xi^{n-1}\left\{\frac{t^ne^{-(\lambda+\theta_1)t}}{n!}\star\frac{t^ne^{-(\lambda\xi+\theta_1)t}}{n!}\star\pi_{1,0}(t)\right\};$$

$$n=1,2,3,\ldots$$

$$\begin{aligned}\pi_{n,2}(t)=&\theta_1\lambda^n\xi^{n-1}\left\{e^{-\lambda t}\star e^{-(\lambda+\theta_1)t}\star\frac{t^{(n-1)}e^{-(\lambda\xi+\theta_2)t}}{(n-1)!}\right\}\\&+\theta_1\mu\lambda^n\xi^{n-1}\left\{e^{-\lambda t}\star e^{-(\lambda+\theta_1)t}\star\frac{t^{(n-1)}e^{-(\lambda\xi+\theta_2)t}}{(n-1)!}\star\pi_{1,0}(t)\right\};\quad n=0,1,2,\ldots\end{aligned}$$

Define the probability generating function (*PGF*) as

$$\mathscr{P}(z,t)=\sum_{n=1}^{\infty}\pi_{n,0}(t)z^n$$

then,

$$\frac{\partial\mathscr{P}(z,t)}{\partial t}=\sum_{n=1}^{\infty}\frac{d\pi_{n,0}}{dt}z^n$$

Using $eq^n(7.1)$ and $eq^n(7.2)$, after some algebra we have

$$\begin{aligned}\frac{\partial\mathscr{P}(z,t)}{\partial t}-(\nu(1-z))\frac{\partial\mathscr{P}(z,t)}{\partial z}=&\left((1-z^{-1})(\nu-\mu)+\lambda\xi(z-1)\right)\mathscr{P}(z,t)\\&+\sum_{n=1}^{\infty}\theta_1\pi_{n,1}(t)z^n+\sum_{n=1}^{\infty}\theta_2\pi_{n,2}(t)z^n-\mu\pi_{1,0}(t)\end{aligned} \tag{7.29}$$

On solving the $eq^n(7.29)$, we obtain

$$\mathscr{P}(z,t) = \exp[\left(\left(z^{-1}-1\right)(\mu-\nu)+\lambda\xi\left(z-1\right)\right)]t + \int_0^t \exp\Bigg(\left[\left(z^{-1}-1\right)(\mu-\nu)+\lambda\xi\left(z-1\right)\right](t-u)\Bigg) \times \Bigg[\sum_{n=1}^{\infty}\theta_1\pi_{n,1}(t)z^n + \sum_{n=1}^{\infty}\theta_2\pi_{n,2}(t)z^n - \mu\pi_{1,0}(t)\Bigg] \tag{7.30}$$

It is well known that if

$$\Psi = 2\sqrt{\xi\lambda\left(\mu-\nu\right)} \quad \& \quad \Omega = \sqrt{\frac{\xi\lambda}{(\mu-\nu)}} \tag{7.31}$$

then,

$$\exp\left\{\left(\xi\lambda z + \frac{\mu-\nu}{z}\right)t\right\} = \sum_{n=-\infty}^{\infty}(\Omega z)^n I_n(\Psi t) \tag{7.32}$$

Using the $eq^n(7.32)$, we have

$$\begin{aligned}\mathscr{P}(z,t) =& \exp\left(\lambda\xi z + \frac{\mu-\nu}{z}\right)t \times \exp\left(-(\mu-\nu)+\lambda\xi\right)t \\ &+ \int_0^t \exp\left(\lambda\xi z + \frac{\mu-\nu}{z}\right)(t-u) \times \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\sum_{n=1}^{\infty}\theta_1\pi_{n,1}(t)z^n du \\ &+ \int_0^t \exp\left(\lambda\xi z + \frac{\mu-\nu}{z}\right)(t-u) \times \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\sum_{n=1}^{\infty}\theta_2\pi_{n,2}(t)z^n du \\ &- \int_0^t \exp\left(\lambda\xi z + \frac{\mu-\nu}{z}\right)(t-u) \times \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\mu\pi_{1,0}(t) du\end{aligned} \tag{7.33}$$

On equating the cofficient of n^{th} power of z of $eq^n(7.33)$ on the both side for $n = 0, 1, 2, \ldots,$ we have

$$\begin{aligned}\pi_{n,0}(t) =& \Omega^n I_n(\Phi t)\exp\left(-(\mu-\nu)+\lambda\xi\right)t \\ &+ \theta_1\int_0^t \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\left[\sum_{k=0}^{n}\Omega^k I_k(.)\pi_{n-k,1}(t) + \sum_{k=0}^{n}\Omega^{-k}I_k(.)\pi_{n+k,1}(u)\right]du \\ &+ \theta_2\int_0^t \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\left[\sum_{k=0}^{n}\Omega^k I_k(.)\pi_{n-k,2}(t) + \sum_{k=0}^{n}\Omega^{-k}I_k(.)\pi_{n+k,2}(u)\right]du \\ &- \mu\int_0^t \exp\left(-(\mu-\nu)+\lambda\xi\right)(t-u)\Omega^n I_n(\Phi t)\pi_{1,0}(t)\,du\end{aligned} \tag{7.34}$$

where $I_n = I_n(\alpha(t-u))$. The $eq^n(7.34)$ hold for negative integer $n = -1,-2,-3,\ldots$ with the *lhs* substituted as zero. Using $I_{-n}(.) = I_n(.)$ for $n = 1,2,3,\ldots$

$$\begin{aligned}
0 =& \Omega^{-n} I_{-n}(\Phi t)\exp\Big(-(\mu-\nu)+\lambda\xi\Big)t \\
&+\theta_1\int_0^t \exp\Big(-(\mu-\nu)+\lambda\xi\Big)(t-u)\left[\sum_{k=0}^{\infty}\Omega^{-(n+k)}I_{n+k}(.)\pi_{n+k,1}(t)\right]du \\
&+\theta_2\int_0^t \exp\Big(-(\mu-\nu)+\lambda\xi\Big)(t-u)\left[\sum_{k=0}^{\infty}\Omega^{-(n+k)}I_{n+k}(.)\pi_{n+k,2}(t)\right]du \\
&-\mu\int_0^t \exp\Big(-(\mu-\nu)+\lambda\xi\Big)(t-u)\Omega^{-n}I_{-n}(\Phi t)\pi_{1,0}(t)\,du
\end{aligned} \tag{7.35}$$

By $eq^n(7.34)$ & $eq^n(7.35)$, for $n = 1,2,3,\ldots$, we have state probabilities when service provider is in busy-state at instant t as

$$\begin{aligned}
\pi_{n,0}(t) =& \exp\Big(-(\mu-\nu)+\lambda\xi\Big)t\left[\Omega^n I_n(\Phi t)-\Omega^{-n}I_{-n}(\Phi t)\right] \\
&+\theta_1\int_0^t \exp\Big(-(\mu-\nu)+\lambda\xi\Big)(t-u) \\
&\times\left[\sum_{k=0}^{n}\Omega^k I_k(.)\pi_{n-k,1}(t)+\sum_{k=0}^{n}\Omega^{-k}I_k(.)\pi_{n+k,1}(u)-\sum_{k=0}^{\infty}\Omega^{n-k}I_{n+k}(.)\pi_{n+k,1}(t)\right]du \\
&+\theta_2\int_0^t \exp\Big(-(\mu-\nu)+\lambda\xi\Big)(t-u) \\
&\times\left[\sum_{k=0}^{n}\Omega^k I_k(.)\pi_{n-k,2}(t)+\sum_{k=0}^{n}\Omega^{-k}I_k(.)\pi_{n+k,2}(u)-\sum_{k=0}^{\infty}\Omega^{n-k}I_{n+k}(.)\pi_{n+k,2}(t)\right]du
\end{aligned} \tag{7.36}$$

Hence, state probabilities at instant t for $n = 0,1,2,\ldots$ when service provider is on type-1 vacation as

$$\pi_{n,1}(t) = \frac{\lambda^n\xi^{n-1}e^{-(\lambda\xi+\theta_1)t}}{n!}+\mu\lambda^n\xi^{n-1}\left\{\frac{t^n e^{-(\lambda+\theta_1)t}}{n!}\star\frac{t^n e^{-(\lambda\xi+\theta_1)t}}{n!}\star\pi_{1,0}(t)\right\}$$

and is on type-2 vacation as

$$\begin{aligned}
\pi_{n,2}(t) =& \theta_1\lambda^n\xi^{n-1}\left\{e^{-\lambda t}\star e^{-(\lambda+\theta_1)t}\star\frac{t^{(n-1)}e^{-(\lambda\xi+\theta_2)t}}{(n-1)!}\right\} \\
&+\theta_1\mu\,\lambda^n\xi^{n-1}\left\{e^{-\lambda t}\star e^{-(\lambda+\theta_1)t}\star\frac{t^{(n-1)}e^{-(\lambda\xi+\theta_2)t}}{(n-1)!}\star\pi_{1,0}(t)\right\}
\end{aligned}$$

respectively.

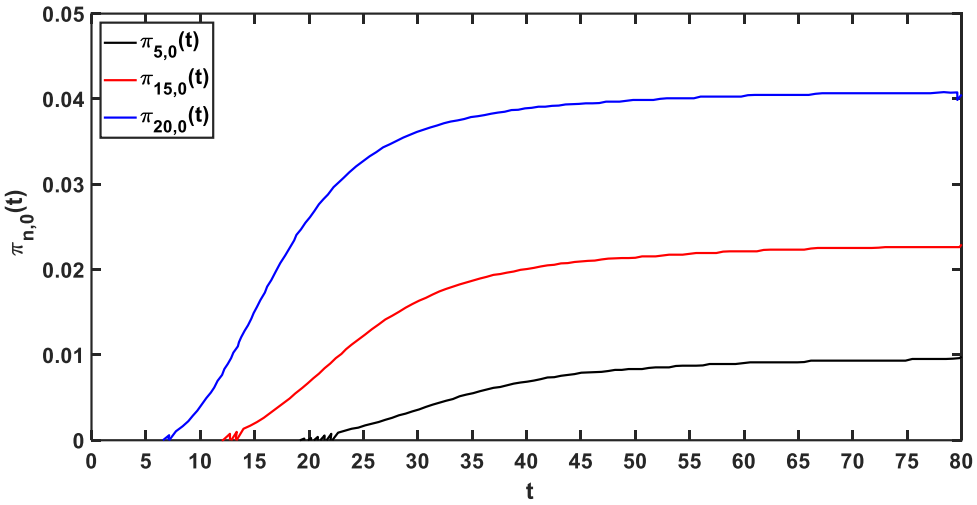


Figure 7.1: The variation of the state probability $\pi_{n,0}(t)$ *wrt* t

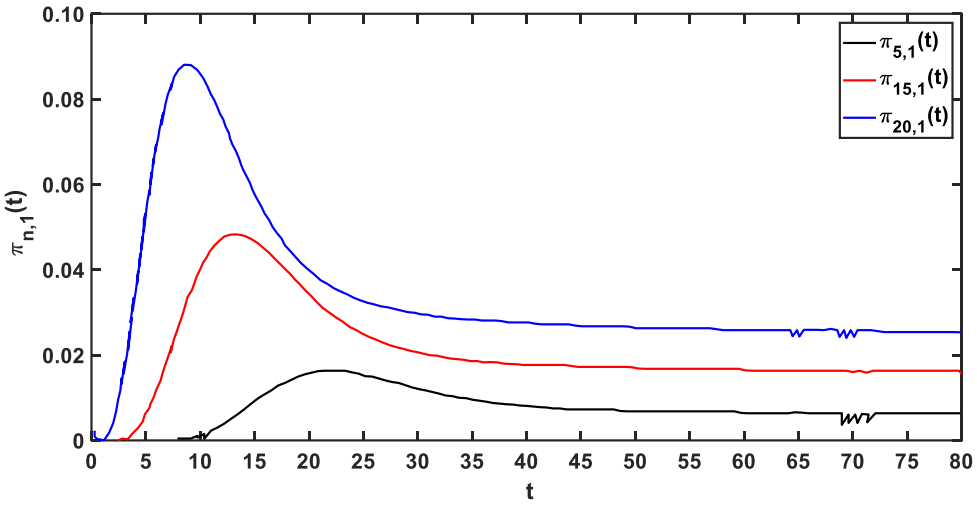


Figure 7.2: The variation of the state probability $\pi_{n,1}(t)$ *wrt* t

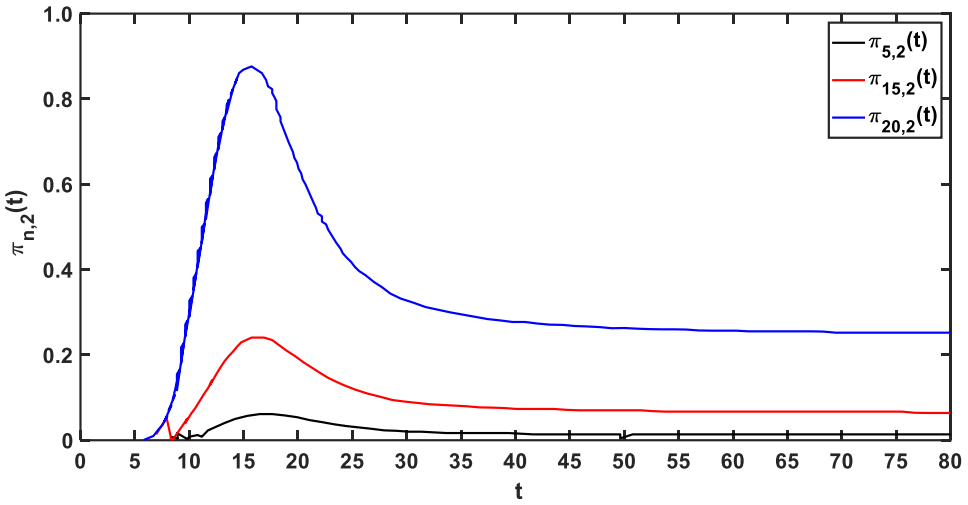


Figure 7.3: The variation of the state probability $\pi_{n,2}(t)$ *wrt* t

For the default value of the involved parameters $\lambda = 0.3$; $\mu = 0.5$, $\nu = 0.1$, $\xi = 0.6$, $\theta_1 = 0.3$ and $\theta_2 = 0.4$, we plot the variation of state-probabilities $\pi_{n,0}$, $\pi_{n,1}$, and $\pi_{n,2}$ in Fig. 7.1, Fig. 7.2, and Fig. 7.3 respectively wherein the deviation is displayed for $n = 5, 15$, and 20. Fig. 7.1-7.3 illustrate that the state probabilities become stable, which prompt the system to tend to steady-state after a long time. Initially, there is much fluctuation in state probabilities which shows the customers are getting service immediately.

7.5 Performance Measures

The acceptance of any queueing model is best evaluated in terms of its system characteristics. Evaluating queueing system performance indices is the most essential and promising method for improving any system. Systematic observation of the state genuinely aids decision-makers in enhancing the performance and efficiency of the queueing system.

7.5.1 Expectation of $N(t)$

Estimating the number of customers in the system $N(t)$ at arbitrary instant t is the primary goal of any queueing modeling. Here, it is expressed as

$$\begin{aligned} m(t) &= E(N(t)) \\ &= \sum_{n=1}^{\infty} n\left(\pi_{n,0}(t) + \pi_{n,1}(t) + \pi_{n,2}(t)\right) \end{aligned}$$

On differentiating both sides *wrt* t, we have

$$m'(t) = \sum_{n=1}^{\infty} n\left(\pi'_{n,0}(t) + \pi'_{n,1}(t) + \pi'_{n,2}(t)\right) \tag{7.37}$$

On substituting the value from $eq^n(7.1)$ to $eq^n(7.8)$ in $eq^n(7.37)$ and using some mathematical manipulation, we get

$$\begin{aligned} m'(t) = &-(\lambda\xi+\mu)\,\pi_{1,0}(t)+(\mu+\nu)\,\pi_{2,0}(t)-\lambda\xi\,(\pi_{1,1}(t)+\pi_{1,2}(t))+\lambda\,(\pi_{0,1}(t)+\pi_{0,2}(t)) \\ &+\sum_{n=2}^{\infty}(\mu-\lambda\xi-\nu)n\pi_{n,0}(t)+\nu\sum_{n=2}^{\infty}n^2\pi_{n,0}(t)+\lambda\xi\sum_{n=2}^{\infty}n\pi_{n-1,0}(t)-\lambda\xi\sum_{n=2}^{\infty}n\pi_{n,2}(t) \\ &+\mu\sum_{n=2}^{\infty}n\pi_{n+1,0}(t)+\nu\sum_{n=2}^{\infty}n^2\nu\pi_{n+1,0}(t)-\lambda\xi\sum_{n=2}^{\infty}n\pi_{n,1}(t)+\lambda\xi\sum_{n=2}^{\infty}n\pi_{n-1,1}(t) \\ &+\lambda\xi\sum_{n=0}^{\infty}n\pi_{n-1,2}(t) \end{aligned} \tag{7.38}$$

$$\begin{aligned} m(t) = &-(\lambda\xi+\mu)\int_0^t\pi_{1,0}(y)dy+(\mu+\nu)\int_0^t\pi_{2,0}(y)dy-\lambda\xi\int_0^t(\pi_{1,1}(y)+\pi_{1,2}(y))\,dy \\ &+\int_0^t\lambda\,(\pi_{0,1}(y)+\pi_{0,2}(y))\,dy+\sum_{n=2}^{\infty}\int_0^t(\mu-\lambda\xi-\nu)n\pi_{n,0}(y)dy \\ &+\nu\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n,0}(y)dy+\lambda\xi\sum_{n=2}^{\infty}\int_0^t n\pi_{n-1,0}(y)dy-\lambda\xi\sum_{n=2}^{\infty}\int_0^t n\pi_{n,2}(y)dy \\ &+\mu\sum_{n=2}^{\infty}\int_0^t n\pi_{n+1,0}(y)dy+\nu\sum_{n=2}^{\infty}\int_0^t n^2\nu\pi_{n+1,0}(y)dy-\lambda\xi\sum_{n=2}^{\infty}\int_0^t n\pi_{n,1}(y)dy \\ &+\lambda\xi\sum_{n=2}^{\infty}\int_0^t n\pi_{n-1,1}(y)dy+\lambda\xi\sum_{n=0}^{\infty}\int_0^t n\pi_{n-1,2}(y)dy \end{aligned} \tag{7.39}$$

7.5.2 The variance of $N(t)$

The variance $V(t)$ of a number of customers in the system $N(t)$ at an arbitrary instant t is calculated as:

$$V(t) = E(N^2(t)) - (E(N(t)))^2 \tag{7.40}$$

where $E(N^2(t))$ represents the 2^{nd} moment of $drv\ N(t)$ at instant t. Therefore,

$$E(N^2(t)) = \sum_{n=1}^{\infty} n^2\,(\pi_{n,0}(t)+\pi_{n,1}(t)+\pi_{n,2}(t))$$

$$E(N(t)) = \sum_{n=1}^{\infty} n\,(\pi_{n,0}(t)+\pi_{n,1}(t)+\pi_{n,2}(t))$$

Differentiating both sides of $eq^n(7.40)$ with respect to t yields

$$V'(t) = E'(N^2(t)) - (E'(N(t)))^2 \tag{7.41}$$

On substituting the values of computed state probabilities, we get

$$\begin{aligned}V'(t) =& -(\mu+\lambda\xi)\pi_{1,0}(t)+(\mu+\nu)\pi_{2,0}(t)-\lambda\xi\pi_{1,1}(t)+\lambda\pi_{0,1}(t)-\lambda\xi\pi_{1,2}(t)+\lambda\pi_{0,2}(t)\\&+(\mu-\lambda\xi)\sum_{n=2}^{\infty}n^2\pi_{n,0}(t)+\nu\sum_{n=2}^{\infty}n^2(n-1)\pi_{n,0}(t)+\theta_2\sum_{n=2}^{\infty}n^2\pi_{n,2}(t)+\mu\sum_{n=2}^{\infty}n^2\pi_{n+1,0}(t)\\&+\nu\sum_{n=2}^{\infty}n^3\pi_{n+1,0}(t)-\lambda\xi\sum_{n=2}^{\infty}n^2\pi_{n,1}(t)+\lambda\xi\sum_{n=2}^{\infty}n^2\pi_{n-1,1}(t)-(\lambda\xi+\theta_2)\sum_{n=2}^{\infty}n^2\pi_{n,2}(t)\\&+\lambda\xi\sum_{n=2}^{\infty}n^2\pi_{n-1,2}(t)-\frac{dm}{dt}\end{aligned} \tag{7.42}$$

Hence, we have

$$\begin{aligned}V(t) =& -(\mu+\lambda\xi)\int_0^t\pi_{1,0}(y)dy+(\mu+\nu)\int_0^t\pi_{2,0}(y)dy-\lambda\xi\int_0^t\pi_{1,1}(y)dy+\lambda\int_0^t\pi_{0,1}(y)dy\\&-\lambda\xi\int_0^t\pi_{1,2}(y)dy+\lambda\int_0^t\pi_{0,2}(y)dy-(\mu+\lambda\xi)\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n,0}(y)dy\\&+\nu\sum_{n=2}^{\infty}\int_0^t n^2(n-1)\pi_{n,0}(y)dy+\theta_2\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n,2}(y)dy+\mu\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n+1,0}(y)dy+\\&\nu\sum_{n=2}^{\infty}\int_0^t n^3\pi_{n+1,0}(y)dy-\lambda\xi\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n,1}(y)dy+\lambda\xi\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n-1,1}(y)dy\\&-(\lambda\xi+\theta_2)\sum_{n=2}^{\infty}\int_0^t n^2\pi_{n,2}(y)dy+\lambda\xi\sum_{n=2}^{\infty}n^2\int_0^t\pi_{n-1,2}(y)dy-m(t)\end{aligned} \tag{7.43}$$

7.6 Numerical Results

The numerical results for different experiments conducted on MAPLE software with a computing system of hardware configuration having processor Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz and RAM 16.0 GB for various involved parameters are summarized in Fig. 7.4-7.6. The depicted results show the effects of various system parameters on the system performance measures, namely, expected customers count in the system ($m(t)$) and time-dependent variance ($V(t)$). Initially we set the default value of system parameters as $\lambda = 0.3$; $\mu = 0.5$, $\nu = 0.1$, $\xi = 0.6$, $\theta_1 = 0.3$ and $\theta_2 = 0.4$.

Fig. 7.4 depicts the deviation in the mean number of customers in the system *wrt* t for different values of λ as 0.2, 0.5, and 0.8. The apparent result is that $m(t)$ is increasing *wrt* λ. As the time t is large, the plot becomes uniform, revealing the system achieves stability after a long time, and the system tends to steady state. Initially, a lot of fluctuation of decreasing and increasing value is observed with customer accumulation before stability.

Fig. 7.5 depicts the deviation of the expected number of customers in the system $m(t)$ *wrt*

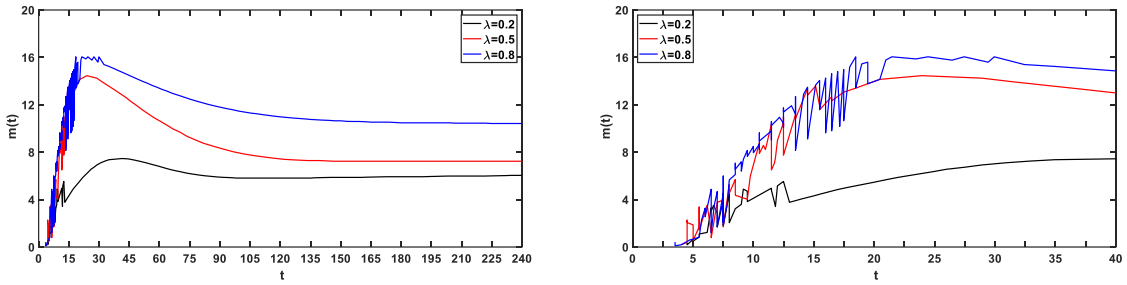


Figure 7.4: The variation of the mean number of the customers in the system $m(t)$ *wrt* t

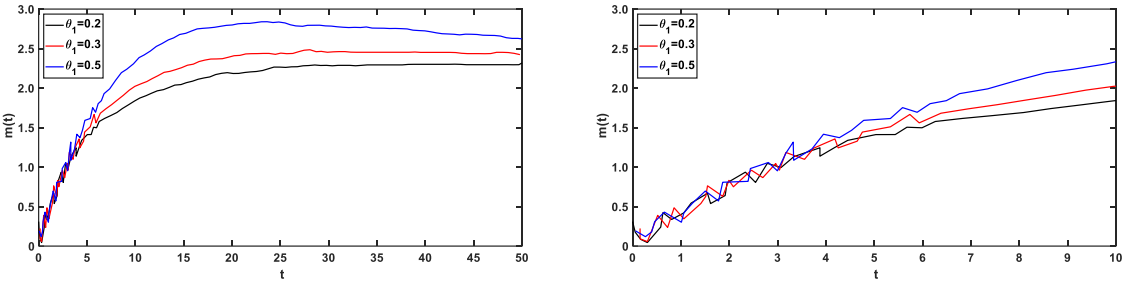


Figure 7.5: The variation of the mean number of the customers in the system $m(t)$ *wrt* t

time t for varying the duration of type-1 vacation as $\theta_1 = 0.2$, 0.3, and 0.5 which denotes the rate at which the server joins the system from type-1 vacation mode. Fig. 7.5 indicates that $m(t)$ increases with time for all values of θ_1 with some fluctuation in the initial time. As the server's vacation time is longer, the system remains with no service provider in this period, and arriving customers either join or show balking behavior. This pattern can be easily inferred from Fig. 7.5 as the customers' count in the system rises for the lesser value of parameter θ_1.

While Fig. 7.6 illustrates the graph of variance $V(t)$ with time t for varying the duration of type-1 vacation as $\theta_1 = 0.2$, 0.3, and 0.5. Fig. 7.6 reveals that $V(t)$ increase with time for all values of θ_1. As the server's vacation time decreases, arriving customers' reluctance behavior decreases. This observation can be easily incidental from Fig. 7.6 as the customers' count variance in the system upsurges for the lesser parameter θ_1.

7.7 Conclusion

In this chapter, we have analyzed the queueing-based congestion model incorporating strategic, differentiated-multiple vacations and customer impatience attributes like balking and reneging. The studied system has infinite differential-difference equations, which are solved with the help of Laplace transformation, Bessel modified function, and generating function

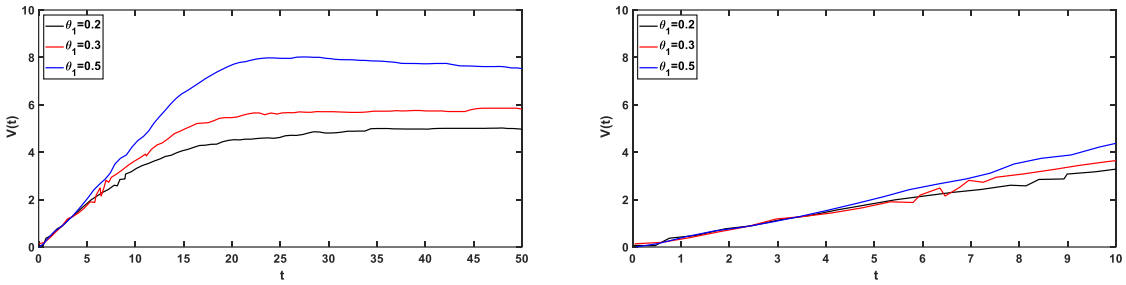


Figure 7.6: The variation of the variance of the number of the customers in the system $V(t)$ *wrt* t

techniques. The transient analysis of the system gives the explicit formula for transient-state probabilities of the proposed queueing-based congestion system. The investigation demonstrates the dynamic congestion behavior in the planning phase. These transient probabilities are helpful in the evaluation of the characteristic measure of the system. The numerical illustrations are performed, which also justify the theoretical results. The present model can be extended for service with general distribution and batch arrivals. The unreliability of the server can also be included in future work.

Chapter 8

Cost Analysis of a Retrial Queueing System with an Unreliable Server Incorporating an Orbital Search Mechanism, Multiple Vacation Policies, and the Balking Phenomenon

This chapter focuses on studying the orbital search concept in Markovian retrial queueing, including multiple vacation policies and server breakdown, which is described by an infinite number of inflow-outflow balanced equations. Processes like arrival, service, search, repair, and vacation are all stochastic in nature. System characteristics are derived using the probability generating function (PGF) technique, and a theoretical background for the PGF technique is established.

8.1 Introduction

Modern businesses, specifically e-commerce and call center service systems, have significantly benefited from advances in technology like artificial intelligence (AI), machine learning (ML), neural networks, etc. in terms of helping servers reduce their idle time and increase their utility via searching for entities that appeared before the system but not in the present. These types of service systems are usually termed "retrial queueing systems" in the queueing theoretic approach. The term "entity" is basically for service seekers in the system, which can be a call, a customer, a machine, etc. A retrial queueing system allows customers who find all servers occupied to join a virtual queue called retrial orbits and retry for service after a random length called retrial time. A retrial queue is comprised of an infinite capacity orbit and a service facility with finite servers. There are many applications for retrial queueing systems, such as telephone switching systems, computer and telecommunication networks [224], which have triggered scientific attention and reinvigorated its study in the last two decades. An analysis of queueing economics in retrial queues was first carried out by Wang and Zhang [194]. An overview of retrial queue theory in real-world call centers and cellular networks systems may be found in [153] by Tuan. References [14], [106], [57], [42], [2] provide a comprehensive survey of retrial queueing systems.

In most studies on retrial queues, it is assumed that servers become idle after service until the arrival of the next primary or retrial customer. This assumption is a hindrance in achieving maximum utility of the server's idle time. Orbital search is a progressive idea introduced by Neuts [145], in which the server search in the orbit for customers during its idle time. A service is followed by another if a search is done; otherwise, it is followed by an idle interval. Therefore, it becomes vital to consider orbital search in retrial queues for maximizing the server's utilization of idle time. Until recently, few research works have been conducted on retrial queues with orbital search (c.f. [15], [39], [46], [64], [62]). Rajadurai et al. [155] implemented the orbital search policy in a Bernoulli vacation schedule $M^{[X]}/G/1$ feedback retrial G-queue with impatient customers.

Besides considering the orbital search policy in our retrial queue model, we will also consider multiple vacation policies and server breakdown. As a result of the multiple vacation policy, when the server has no customer to serve within the system and no customer to find via orbit search, it takes a vacation after a random time period. After returning from this vacation, he keeps on taking vacations until he finds atleast one customer in the orbit. Levy and Yechiali [118] came up with the idea of the vacation queueing paradigm. Various studies on retrial with vacation has expanded and deepened including two-phase service [192], feedback [155], [18], server breakdown [156], [188], and second optional service

with balking [128]. Shin[173] developed a unified model of the level dependent quasi-birth-and-death (LDQBD) process used in Markovian multi-server queues with customers retrials and server's vacations.

Server breakdown is considered as a most common cause of service disruption. It is inevitable that service disruptions will occur in many real-world situations. Studies commonly assume that servers in service stations are always operational and that stations don't malfunction. These presumptions, however, are essentially irrational. A breakdown of a service station in real life is common and needs to be fixed on a frequent basis. In this chapter, the server is subject to breakdown while providing service to the customer for a short time interval. When a server malfunctions, it is sent for repair, which results in a service interruption for customers arriving during that time. After the server breakdown, a customer who had just been served waits for the remaining service to be delivered. Choudhury et al. [32] examined a retrial queueing system with two service failure and repair stages. Ke et al. [100] stressed on a feedback retrial queue with customer balking and unreliable servers. Krishna et al. [110] pointed out a retrial queue with server subject to two types of breakdowns and repairs. Interested readers can see the retrial queue model with server breakdown in [65], [164], [131].

In our model, the server makes strategic searching decisions in the orbit in a manner similar to the aforementioned work on orbital search retrial queueing models. To the best of our knowledge, this is the first effort to study how the realistic consideration of orbital search, vacation, and server breakdown impacts the system's performance and total cost function. This study also looks at the effects of the model's features from three different aspects: (i) We have stressed the need to obtain the probabilities of the server's state analytically using the probability generating functions technique. (ii) In this context, analyzing the system characteristics is highly complex due to their non-linear nature. Therefore, numerical experiments are carried out to point out the impact of system parameters on its performance measures. We evaluate and validate our proposed model using extensive numerical results. (iii) Our key findings are in obtaining optimality of the total cost of the model with regard to critical parameters using newly developed meta-heuristic optimization techniques.

The rest of our chapter proceeds as follows: In the coming Section 8.2, we address the description of the proposed model in more detail. In Section 8.3, we derive the server's state probabilities and various system characteristics analytically by employing the probability generating functions technique. Nextly, the cost function is mathematically formulated in Section 8.4. In Section 8.5, we elaborated on the social group optimization algorithm and its pseudocode for implementation purposes. Numerical simulations are presented in Section

8.6 to describe the dynamics of the system, including its performance measures and optimality of the total cost. Finally, we conclude with key findings and propose future plans for the extension of this model in Section 8.7.

8.2 Model Description

We consider a system that consists of an unreliable server serving in a retrial queueing system with infinite capacity, an orbital search mechanism, multiple vacation policies, and the balking phenomenon of arriving customers. This model considers the common occurrences of a retrial service system in view of its applicability. For the sake of completeness, the main stochastic processes of the model are described below:

Arrival Processes

The inputs or entities arriving in the system are termed "customers" in this chapter. The arrival of customers in the system occurs in a Poisson fashion with the parameter λ. Arrivals initially join the system as "primary customers", but if the server is busy, they tend to join the orbit as "retrial customers". These retrial customers keep on reattempting to get the service in random time periods referred to as "retrial times", following an exponential distribution with parameter Γ.

The Service Process

The outgoing of customers from the system can be in any of the following scenarios: after service completion, balk out of the system, or joining orbit to give a chance for reattempts. If the service is completed, the customer exits the system after receiving it. The service rendered by the server in a given time period is called the "service time", following an exponential distribution with parameter μ. The customers chosen for service are based on first-come-first-serve (FCFS) queue discipline.

Repair and Breakdown Processes

An unreliable server, according to this model, is prone to failures due to a variety of causes, such as hardware or software faults in the case of a machine, an emergency or health issues in the case of a human, or natural calamities that disrupt system functioning. The breakdown process of the server is random and occurs over a time period following an exponential distribution with parameter υ. The breakdown causes the system to lose money, so repairs

are started right away to cut down on the loss. The time to repair also follows an exponential distribution with parameter ϑ.

Orbital Search Process

Once the server gets rid of servicing the customer present in the system, it either starts to search for customers from orbit with probability p or remains idle in the system with probability $q(=1-p)$. The search for customers starts at the head of the orbit. Meanwhile, in the search operation, if there is a request from either the primary or retrial customer, it is taken into consideration and the search stops. Inter-search times follow an exponential distribution with θ as a parameter.

Vacation Process

When the server finds system empty and also found no request from orbit via orbital search, it takes a vacation with rate δ and returns back to the system as soon as he founds a customer in system or vacation time is over. The time duration of server's vacation is exponentially distributed with rate ς. After return from vacation, if server found no customer in the system and orbit, it continues to take vacation until he founds atleast one customer in the system.

Balking Rules

Upon arriving in the system if customer finds server busy he decides to whether join with probabilities q_0, q_1, and q_2 when server is in busy, vacation, and breakdown states, respectively. Otherwise, customers shows impatience attribute and decides to balk away from the system.

The Indepedence

All of the above stochastic processes are mutually independent of each other.
In the model formulation, the necessary parameters used throughout this chapter are as given in Table 8.1.

8.2.1 Practical Justification of the Model

In banks or multinational companies (MNCs), there are a finite number of servicemen who work as customer care agents by providing them information about the company's policies or registering a complaint if the customer is unsatisfied with its services. Customers are present virtually when they contact customer care personnel over the phone. This will arise

Table 8.1: List of parameters used

Variable	Description
λ	arrival rate of primary customer
Γ	retrial rate customers waiting in queue for service
μ	the server's service rate
θ	search rate of the server in the orbit
δ	Rate of return of the server to the vacation state from the busy state
ς	Rate of returning the server from vacation to its normal busy state
υ	breakdown rate of the server
ϑ	repair rate for the server
q_0	joining probability of customers when the server is in busy state
q_1	joining probability of customers when the server is in vacation state
q_2	joining probability of customers when the server is in breakdown state
$P_{n,j}$	The system's stationary probability while in state (n, j)
$\Pi_j(z)$	probability generating function of $P_{n,j}$

in two cases: either the call gets connected or the caller remains in waiting due to the occupancy of the service provider with other customers. These waiting customers keep on retrying for calls to get connected or may leave the system if waiting is beyond their threshold limit. Customer care agents, on the other hand, may search for customers in the virtual queue or orbit using their waiting call history, or they may remain idle in the system. Meanwhile, if there are any technical faults or network issues, referred to as "server breakdowns" in queueing terminology, the service may be suspended, repaired, and then resume taking calls. Due to their limited capacity to work, customer service representatives may repeatedly take a random time period off in the form of vacation and return after the vacation period is over. The proposed model incorporates all the scenarios described in the above example.

8.3 Steady-State Analysis

The above-described queueing system is modelled as a quasi-birth-and-death (QBD) process with system states $\{(N,J); N \geq 0, J = 1,3,4 \ \& \ N \geq 1, J = 0,2\}$, where N represents the number of customers in the system and J represents the state of the server. The joint probability distribution function $P_{n,j}$ represents the long-run fraction of time that the system remains in state $(N = n, J = j)$. Let $\{P_{n,0}, n \geq 1; P_{n,1}, n \geq 0; P_{n,2}, n \geq 1; P_{n,3}, n \geq 0; P_{n,4}, n \geq 0;\}$ be the stationary distribution of the Markov chain $\{N(t), J(t), t \geq 0\}$. Let $\Pi_j(z), j = 0,1,2,3,4$ be

the partial generating functions which are given as follows

$$\Pi_0(z)=\sum_{n=1}^{\infty}z^nP_{n,0};\quad \Pi_1(z)=\sum_{n=0}^{\infty}z^nP_{n,1};\quad \Pi_2(z)=\sum_{n=1}^{\infty}z^nP_{n,2}$$
$$\Pi_3(z)=\sum_{n=0}^{\infty}z^nP_{n,3};\quad \Pi_4(z)=\sum_{n=0}^{\infty}z^nP_{n,4},\quad |z|\leq 1$$

We have the following preliminary result.

Theorem 8.3.1. *In the steady-state for the $M/M/1$ constant retrial queue with multiple vacations, server breakdown and orbital search for the given arrival rates $(\lambda,\lambda_0,\lambda,\lambda_1,\lambda_2)$, the probabilities that the server is idle (P_i), busy (P_b), in search orbit (P_s), on vacation (P_v), and under repair (P_r) respectively, are as follows:*

$$P_i=\Pi_0(1)=\frac{q\mu}{(\lambda+\Gamma)}(B-1)P_{0,1};$$
$$P_b=\Pi_1(1)=BP_{0,1};$$
$$P_s=\Pi_2(1)=\frac{p\mu}{(\lambda+\Gamma+\theta)}(B-1)P_{0,1};$$
$$P_r=\Pi_3(1)=\frac{\upsilon}{\vartheta}BP_{0,1};$$
$$P_v=\Pi_4(1)=\frac{\delta}{\varsigma}BP_{0,1};$$

where,

$$B=\frac{q\mu\Gamma(\lambda+\Gamma+\theta)+p\mu(\Gamma+\theta)(\lambda+\Gamma)}{\begin{array}{r}-\lambda_0(\lambda+\Gamma)(\lambda+\Gamma+\theta)\vartheta\varsigma+(\Gamma+\theta)p\mu(\lambda+\Gamma)\vartheta\varsigma\\ -\upsilon\lambda_2(\lambda+\Gamma)(\lambda+\Gamma+\theta)\varsigma-\delta\lambda_1\vartheta(\lambda+\Gamma)(\lambda+\Gamma+\theta)\end{array}}$$

$$P_{0,1}=\frac{(\lambda+\Gamma)(\lambda+\Gamma+\theta)\vartheta\varsigma}{\begin{array}{r}q\mu(B-1)(\lambda+\Gamma+\theta)\vartheta\varsigma+B(\lambda+\Gamma)(\lambda+\Gamma+\theta)\vartheta\varsigma\\ +p\mu(\lambda+\Gamma)\vartheta\varsigma(B-1)+\upsilon B(\lambda+\Gamma)\end{array}}$$

Proof. We have the following equations using the birth and death process and relating the system's state to a steady state.

$$(\lambda_1+\varsigma)P_{n,4}=\lambda_1P_{n-1,4}+\delta P_{n,1};\quad n\geq 0 \tag{8.1}$$
$$(\lambda+\Gamma)P_{n,0}=q\mu P_{n,1};\quad n\geq 1 \tag{8.2}$$
$$(\lambda_0+\mu+\upsilon+\delta)P_{n,1}=\lambda_0P_{n-1,1}+\lambda P_{n,0}+\lambda P_{n,2}+\Gamma P_{n+1,0}+(\Gamma+\theta)P_{n+1,2}+\vartheta P_{n,3};\quad n\geq 1 \tag{8.3}$$
$$(\lambda_0+\upsilon+\delta)P_{0,1}=\Gamma P_{1,0}+(\Gamma+\theta)P_{1,2}+\vartheta P_{0,3}+\varsigma P_{0,4}; \tag{8.4}$$
$$(\lambda+\Gamma+\theta)P_{n,2}=p\mu\ P_{n,1};\quad n\geq 1 \tag{8.5}$$
$$(\lambda_2+\vartheta)P_{n,3}=\lambda_2P_{n-1,3}+\upsilon P_{n,1};\quad n\geq 0 \tag{8.6}$$

where, $P_{-1,j}=0,\quad j=1,3,4.$

Multiplying Eqns. 8.1 and 8.6 by z^n and summing it from $n=0$ to $n=\infty$ and then using

$\Pi_1(z)$, $\Pi_3(z)$, and $\Pi_4(z)$ defined above, we have

$$(\lambda_1(1-z)+\varsigma)\Pi_4(z) = \delta\Pi_1(z) \tag{8.7}$$

$$(\lambda_2(1-z)+\vartheta)\Pi_3(z) = \upsilon\Pi_1(z) \tag{8.8}$$

Multiplying Eqns. 8.2 and 8.5 by z^n and summing it from $n=1$ to $n=\infty$ and then using $\Pi_0(z)$, $\Pi_1(z)$, and $\Pi_2(z)$ defined above, we have

$$(\lambda+\Gamma)\Pi_0(z) = q\mu(\Pi_1(z)-P_{0,1}) \tag{8.9}$$

$$(\lambda+\Gamma+\theta)\Pi_2(z) = p\mu(\Pi_1(z)-P_{0,1}) \tag{8.10}$$

Similarly, multiplying Eqn. 8.3 by z^n and summing it from $n=0$ to $n=\infty$, we have

$$\begin{aligned}(\lambda_0(1-z)+\mu+\upsilon+\delta)\Pi_1(z) =& \mu P_{0,1} + (\frac{\Gamma}{z}+\lambda)\Pi_0(z) + (\frac{\Gamma+\theta}{z}+\lambda)\Pi_2(z) \\ &+ \vartheta\Pi_3(z) + \varsigma\Pi_4(z)\end{aligned} \tag{8.11}$$

Solving Eqns. 8.7-8.11, and after some algebraic manipulations we obtain

$$\Pi_1(z) = \frac{\left[\mu - (\frac{\Gamma}{z}+\lambda)\frac{q\mu}{\lambda+\Gamma} - \frac{p\mu}{\lambda+\Gamma+\theta}(\frac{\Gamma+\theta}{z}+\lambda)\right]P_{0,1}}{\lambda_0(1-z)+\mu+\upsilon+\delta-(\frac{\Gamma}{z}+\lambda)\frac{q\mu}{\lambda+\Gamma}-(\frac{\Gamma+\theta}{z}+\lambda)\frac{p\mu}{\lambda+\Gamma+\theta}-\frac{\vartheta\upsilon}{\lambda_2(1-z)+\vartheta}-\frac{\varsigma\delta}{\lambda_1(1-z)+\varsigma}} \tag{8.12}$$

Substituting $z=1$ in Eqn.8.12 and using theory of calculus for indeterminate forms, we obatin

$$\Pi_1(1) = BP_{0,1} \tag{8.13}$$

where

$$\begin{aligned}B = &\frac{q\mu\Gamma(\lambda+\Gamma+\theta)+p\mu(\Gamma+\theta)(\lambda+\Gamma)}{(\lambda+\Gamma)(\lambda+\Gamma+\theta)[-\lambda_0\vartheta\varsigma-\upsilon\lambda_2\varsigma-\delta\lambda_1\vartheta]} \\ &+\mu\vartheta\varsigma[\Gamma q(\lambda+\Gamma+\theta)+(\Gamma+\theta)p(\lambda+\Gamma)]\end{aligned} \tag{8.14}$$

Solving Eqns. 8.7-8.10 by substituting $z=1$ and Eqn. 8.13, we obtain the following:

$$\Pi_0(1) = \frac{q\mu(B-1)}{\lambda+\Gamma}P_{0,1} \tag{8.15}$$

$$\Pi_2(1) = \frac{p\mu(B-1)}{\lambda+\Gamma+\theta}P_{0,1} \tag{8.16}$$

$$\Pi_3(1) = \frac{\upsilon B}{\vartheta}P_{0,1} \tag{8.17}$$

$$\Pi_4(1) = \frac{\delta B}{\varsigma}P_{0,1} \tag{8.18}$$

To solve $P_{0,1}$ explicitly, we use the normalizing condition of probability as

$$\sum_{j=0}^{4}\Pi_j(1) = 1$$

which gives

$$P_{0,1} = \frac{(\lambda+\Gamma)(\lambda+\Gamma+\theta)\vartheta\varsigma}{(B-1)\mu\vartheta\varsigma[q(\lambda+\Gamma+\theta)+p(\lambda+\Gamma)]+(\lambda+\Gamma+\theta)(\lambda+\Gamma)B[\vartheta\varsigma+\upsilon\varsigma+\delta\vartheta]}$$

□

Lemma 8.3.2. *The system is stable if and only if $B>1$, where B is given by Eqn.8.14.*

Proof. The non-negativity axiom of probability gives that Eqns. 8.15-8.18 are valid for $B > 1$.

On the other hand, the inequality $B > 1$ is also necessary for the system to be stable, which can be guaranteed by

$$P_{0,1} = \frac{(\lambda+\Gamma)(\lambda+\Gamma+\theta)\vartheta\varsigma}{(B-1)\mu\vartheta\varsigma[q(\lambda+\Gamma+\theta)+p(\lambda+\Gamma)]+(\lambda+\Gamma+\theta)(\lambda+\Gamma)B[\vartheta\varsigma+\upsilon\varsigma+\delta\vartheta]} > 0$$

Thus, $B > 1$ is a necessary and sufficient condition for the system to be stable. □

After that, we want to use the server's different system states to figure out orbit size distributions.

Theorem 8.3.3. *The mean orbit sizes when the server is in system states such as busy, idle, in search period, under repair, and on vacation for the Markovian retrial queue with multiple vacations, an unreliable server, customer balking, and an orbital search mechanism are respectively given by*

$$\begin{aligned}
N_1 &= CP_{0,1} \\
N_0 &= \frac{q\mu}{\lambda+\Gamma}N_1 \\
N_2 &= \frac{p\mu}{\lambda+\Gamma+\theta}N_1 \\
N_3 &= \frac{\lambda_2\upsilon BP_{0,1}+\upsilon\vartheta N_1}{\vartheta^2} \\
N_4 &= \frac{\lambda_1\delta BP_{0,1}+\delta\varsigma N_1}{\varsigma^2}
\end{aligned}$$

where,

$$C = \frac{\left(\frac{\Gamma q\mu}{\lambda+\Gamma}+\frac{(\Gamma+\theta)p\mu}{\lambda+\Gamma+\theta}\right)\left(\lambda_0+\frac{\lambda_2\upsilon}{\vartheta}(1+\frac{\lambda_2}{\vartheta})+\frac{\lambda_1\delta}{\varsigma}(1+\frac{\lambda_1}{\varsigma})\right)}{\left(\lambda_0-\frac{\Gamma q\mu}{\lambda+\Gamma}-\frac{(\Gamma+\theta)p\mu}{\lambda+\Gamma+\theta}+\frac{\lambda_2\upsilon}{\vartheta}+\frac{\lambda_1\delta}{\varsigma}\right)^2}$$

Proof. By differentiating Eqns. 8.7-8.11 with respect to z, denoted by $N_j = \Pi_j'(1), j = 0,1,2,3,4$ and using 8.12, after some algebraic calculations, we have the following results. □

8.3.1 Mean Waiting Time

Let W be the waiting time of a customer in system given by

$$W = W_o + W_s$$

where, W_o and W_s denote waiting times in orbit and in service completion, respectively. We know that, the time required for service completion $W_s = \frac{1}{\mu}$, we must obtain W_0.

Theorem 8.3.4. *For the Markovian retrial queue with multiple vacations, an unreliable server, customer balking, and an orbital search mechanism, the mean sojourn time of the customer in orbit is given by*

$$W_o = \frac{N}{\lambda_{ret}} \tag{8.19}$$

where, $N = N_0 + N_1 + N_2 + N_3 + N_4$ *and* $\lambda_{ret} = \lambda_0\Pi_1(1) + \lambda_1\Pi_3(1) + \lambda_2\Pi_4(1)$

Proof. In idle and vacation states, the server immediately starts the service upon customer arrival. So the wait in orbit is required when the server is unavailable upon its arrival according to the PASTA property, which can be possible in three cases as follows:

- The server is in busy state: $\Pi_1(1)$
- The server is in breakdown state: $\Pi_3(1)$
- The server is in vacation state: $\Pi_4(1)$

Thus, the total arrival rate in the retrial orbit is $\lambda_{ret} = \lambda_0\Pi_1(1) + \lambda_1\Pi_3(1) + \lambda_2\Pi_4(1)$.
In addition, from Theorem 8.3.3, the mean number of customers in the orbit is $N = N_0 + N_1 + N_2 + N_3 + N_4$.
From the Little's formula, we can obtain 8.19. □

8.4 Cost Analysis

Economic agglomeration effects of expenses incurred in the system have promoted progress in the study of cost minimization problems using an effective optimization technique. In this regard, state-of-the-art metaheuristic optimization techniques have greatly accelerated the development of the cost minimization and profit maximization problems of complex queueing systems.
Now, we consider the proposed total cost optimization problem which can be mathematically formulated as:

$$TC^* = \underset{\mu^*,\theta^*}{\text{minimize}}\, TC(\mu,\theta)$$

where, the total cost function $TC(\mu,\theta)$ is given by

$$TC(\mu,\theta) = C_hN + C_vP_v + C_iP_i + C_bP_b + C_sP_s + C_rP_r + C_wW + C_m\mu + C_\theta\theta \tag{8.20}$$

The costing attributable to different aspects of the system are as follows:
$C_h \equiv$ Cost for sustaining per customer in the retrial space during unit time.
$C_v \equiv$ Cost/unit time of the server while in vacation period.
$C_i \equiv$ The cost per unit time when the server is sitting idle in the system.
$C_b \equiv$ Cost for the server in per unit time in the normal busy mode.

$C_s \equiv$ Cost of the server in the orbital search state per unit of time.
$C_r \equiv$ The cost spent per customer per unit time when the server is in breakdown state.
$C_w \equiv$ The cost agreed upon per unit time spent by the customer waiting for service.
$C_m \equiv$ Cost/unit time of rendering service by the server with rate μ while in busy state.
$C_\theta \equiv$ Cost/unit time of the server while searching for customers in the orbit with rate θ.
The terms N, P_v, P_i, P_b, P_s, P_r, and W in Eq. 8.20 are highly non-linear in decision parameters (μ, θ) and multiplied by different cost rates which makes $TC(\mu, \theta)$ tedious to minimize analytically. Hence, it is challenging to execute analytical methods for such optimization problems, especially regarding convexity, as well as to determine the optimal values. The goal of the system administrators is to decide how to trade off between cost minimization and performance maximization.

The overall objective of this study is to implement an efficient and state-of-the-art metaheuristic optimization technique to the formulated total cost minimization problem and consequently obtain the optimal cost TC^* of the system along with optimal values of service rate μ^* and searching rate θ^*. In this regard, we opt for the metaheuristic optimization technique (social group optimization algorithm) discussed in the coming Section 8.5 to evaluate the optimality.

8.5 Social Group Optimization technique

Inspiration

The inspiration for the population-based Social group optimization(SGO) algorithm developed by Satapathy and Naik [163] in 2016, comes from the concept of the social behavior of human beings toward solving complex tasks in life. There are a number of behavioral traits that humans possess to solve their problems in life. Individuals sometimes find these problems too complex to solve alone and form groups to solve them with the influence of one another's traits. On the basis of the idea that solving a given complex problem in a group comes out to be more effective and efficient than individuals in exploiting and exploring their different traits. Also, it has been observed that living entities imitate or follow their surroundings and so human beings as well mimic the knowledge sharing concepts in solving any task by observing others who are better than them. A person's fitness value corresponds to their ability to solve a problem in SGO. Consequently, the person with the best fitness value enhances the knowledge of the entire group.

Mathematical Formulation of SGO

Let $P_i, i = 1,2,3,...,N$ be the persons of a social group defined by $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$, where D is the number of variables as a person's behavioral traits which determines the dimensions of a person and $f_i, i = 1,2,...,N$ are their corresponding fitness values. The knowledge of each individual in a group is mapped by their fitness. SGO is divided into mainly two phases: improving and acquiring phase ([139], [138]).

Improving phase: In this phase, the knowledge level of each person in the group is improved with the impact of the best person in the group, who is one with the highest level of knowledge and capacity to solve the problem. The knowledge updation of every individual in the group is according to the relation:

$$Pnew_{i,j} = \zeta * Pold_{i,j} + r * (gbest_j - Pold_{i,j}) \tag{8.21}$$

where, $Pnew_{i,j}$ and $Pold_{i,j}$ are new and old knowledge levels, respectively, ζ represents the self-introspection parameter lies between 0 and 1, r is a random numeral $[0,1]$, and $gbest$ is knowledge level of best person in group defined as $gbest_j = max\{f_i, i = 1,2,...,N\}$ at jth iteration for solving maximization problems.

Acquiring phase: In the acquiring phase, each person increases his/her knowledge with the mutual interaction among other people in the group by randomly select one person from the group P_r based on $i \neq r$, and the best person with knowledge level $gbest$ in the group at that point in time. Once the fitness value becomes $f_i > f_r$, the knowledge updating procedure is executed as:

$$Pnew_{i,j} = Pold_{i,j} + r_1 * (P_{i,j} - P_{r,j}) + r_2 * (gbest_j - P_{i,j}) \tag{8.22}$$

otherwise,

$$Pnew_{i,j} = Pold_{i,j} + r_1 * (P_{r,j} - P_{i,j}) + r_2 * (gbest_j - P_{i,j}) \tag{8.23}$$

where, r_1 and r_2 are two independent random numbers in the range $[0,1]$, $P_{r,j}$ is the knowledge value of the randomly chosen individual ([94], [123]).

8.6 Computational Analysis

In order to evaluate the influence of each parameter on the system characteristics obtained in Sections 8.3 and 8.4, we performed a numerical study. As a baseline, we used the following parameter values given in Table 8.2. The purpose of this numerical section is twofold. First, it is to demonstrate the long-run performance of the system analyzed in Section 8.3. Second, it is being shown how the optimization techniques employed in obtaining the optimal values of critical parameters and minimum total cost.

Fig. 8.1 corresponds to variations in probabilities of the server's state with regard to system

Algorithm 8 Pseudo code for SGO

Parameter Initialization: population size N, search dimension D, self-introspection factor c, the objective function f, and total iteration t_{max}.
Randomly initialize the population and evaluate fitness value of each individual.
while iteration $< t_{max}$ or convergence criterion **do**
    Find the best person with fitness value *gbest* and perform the improving phase.
    for i=1:N **do**
        for j=1:D **do**
            Update $Pnew_{ij}$ according to Eqn. 8.21
        end for
    end for
    If *Pnew* provides better fitness than *Pold*, accept *Pnew*.
    Initiate the acquiring phase to update knowledge level based on attained *gbest*.
    for i=1:N **do**
        Randomly select one person P_r, $i \neq r$
        if f_i is better than f_r **then**
            for j=1:D **do**
                update $Pnew_{ij}$ according to Eqn. 8.22
            end for
        else
            for j=1:D **do**
                update $Pnew_{ij}$ according to Eqn. 8.23
            end for
        end if
        If *Pnew* provides better fitness than *Pold*, accept *Pnew*.
    end for
end while
itration= itration+1
Output: Return the best optimum solution.

Table 8.2: The data set of parameters involved in the outlined model, along with their sources

System Parameters	Numeric value	Source(s)
λ	1	[61], [230]
Γ	7.5	[64]
μ	5.25	Assumed
θ	0.75	Assumed
δ	0.1	[61]
ς	0.8	[201]
υ	1.3	Assumed
ϑ	1.5	Assumed
p	0.55	[64], [63]
q	0.45	[64], [63]
q_0	0.9	Assumed
q_1	0.8	Assumed
q_2	0.7	Assumed
C_h	250	[170]
C_v	30	Assumed
C_i	300	Assumed
C_b	150	[169]
C_s	200	Assumed
C_r	100	Assumed
C_θ	4.5	Assumed
C_m	60	[90]
C_w	250	Assumed

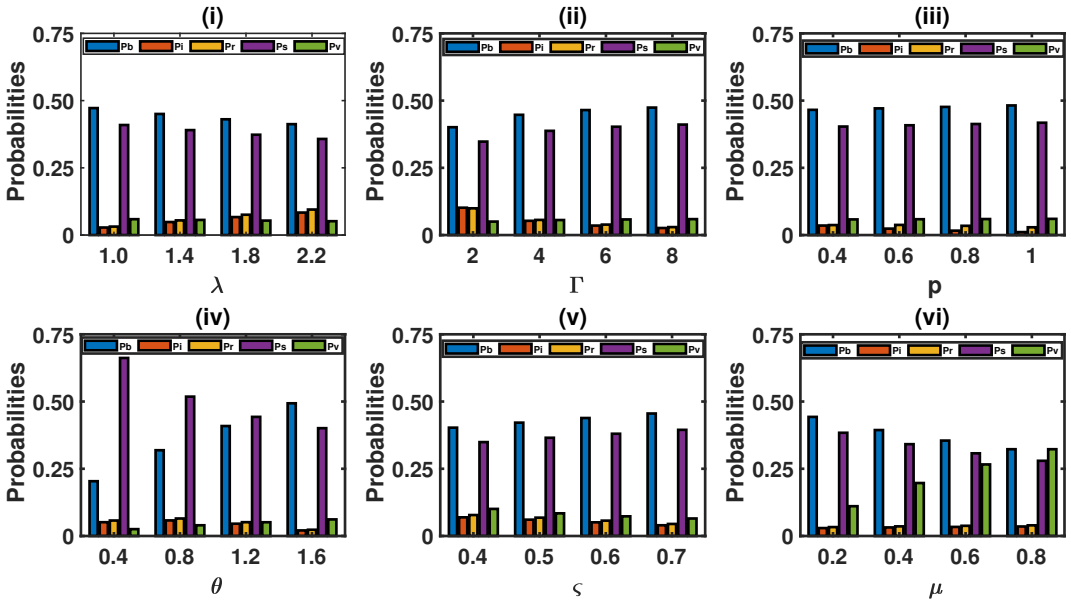


Figure 8.1: Bar graphs for distribution of the server's state probabilities wrt system parameters. The default set of paramters values are given in Table 8.2.

parameters. The orbital search quest and increased retrials are designed to keep the server busy in either serving (P_b) or searching (P_s). On the contrary, the idleness (P_i) of the server is reduced. These observations can be verified from Fig. 8.1 (ii, iii). The improvement in repair and vacation rates of the server is additive in terms of server utilization, and busy state probabilities tend to increase with it, while the proportional change in search probabilities is negative(Fig. 8.1(iv, v)). The vacation rate matches very well with the server being in vacation state (P_v), as illustrated in Fig. 8.1(vi).

Fig. 8.2 exhibits graphs for orbit size (N) versus service rate (μ) with regard to various system parameters. The sudden fluctuation in orbital size for arrivals is the impact of retrial and the orbit search mechanism when λ is quite small. Eventually, as the arrival of primary customers arises, the effect is suppressed and N behaves linearly in decreasing proportion, as can be seen in Fig. 8.2(i). Customers retry for service from orbit in retrials and are treated as if they are waiting if the server is already busy. In such scenarios, the count of orbital sizes reduces when retrials are more frequent (Fig. 8.2(ii)). Repairing servers faster and quick returns from vacation are regarded as a benefit to system efficiency, and thus N decreases as the repair rate ϑ and vacation return rate to busy state ς increases (Fig. 8.2(iii, v)). When the server is unavailable, either in a breakdown state or during a vacation period, the orbit size graph exhibits unusual behavior. Normally, the orbit size increases, but in this case, as the server breakdown rate (υ) increases further, the orbit size decreases until a limit

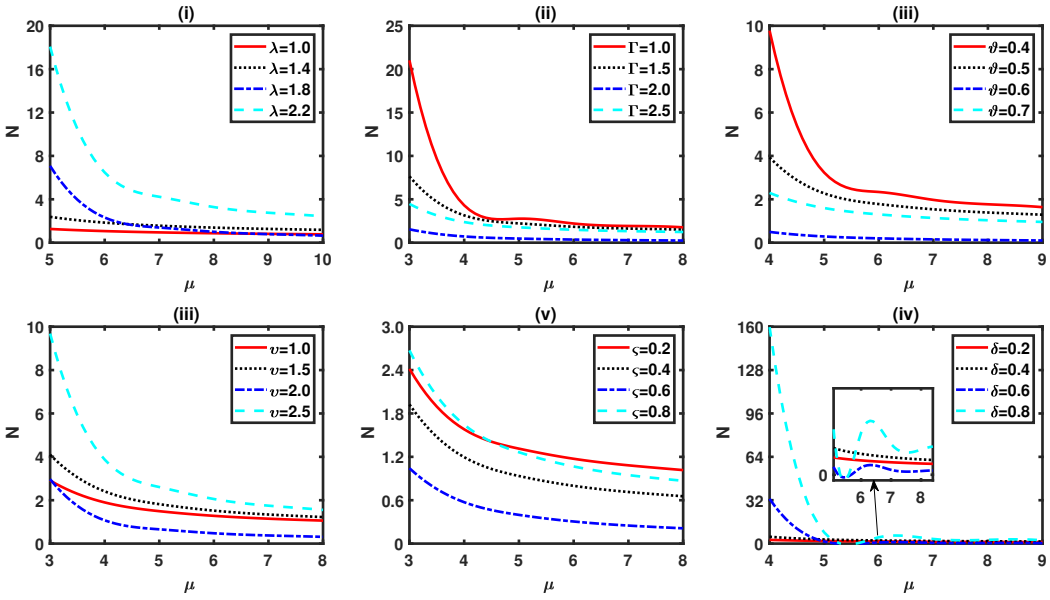


Figure 8.2: Line graphs of the mean orbit size (N) and the service rate of the server (μ) for different system parameters. The default set of paramters values are given in Table 8.2.

is reached, at which point the behaviour reverses. The reason behind this is that as the server becomes unavailable due to the balking effect, there is a tendency for customers not to join the system when the server is in a breakdown state or during a vacation period. This balking factor prevents further increases in υ, causing N to naturally increase (Fig. 8.2 (iv, vi)).

The sojourn time W is in line with orbit size N and inverses with total arrival rate λ_{ret}. So, the dominance of N or λ_{ret} determines variations in the graphs of W versus μ, as shown in Fig. 8.3, for different parameters.

Fig. 8.4(i, ii) resemble convex graphs, indicating that for the parameters μ and θ, it is possible to obtain a convex surface graph with a closed contour, as shown in Fig. 8.4(iii, iv). According to Fig. 8.4(i,ii), the total cost TC decreases as the server's service rate and searching rate improve until a point in both cases and then increases. This trait illustrates that an increase in service and search rates makes the system less congested and more profitable in terms of the number of services rendered, but a further increase in μ and θ makes it more expensive and less profitable from an economic perspective. Thus, to overcome this issue, there is a trade-off between cost minimization and server utilization maximization through simultaneous variation in mu and theta for the TC function. Fig. 8.4 (iii, iv) gives the results.

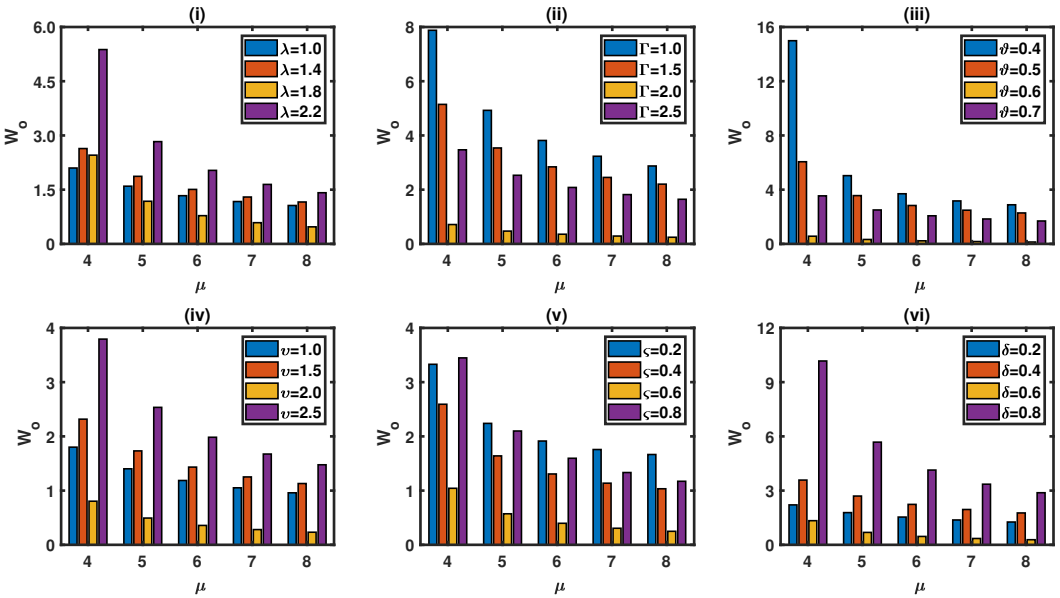


Figure 8.3: Bar graphs of the mean waiting time in orbit (W_o) and the service rate of the server (μ) for different system parameters. The default set of paramters values are given in Table 8.2.

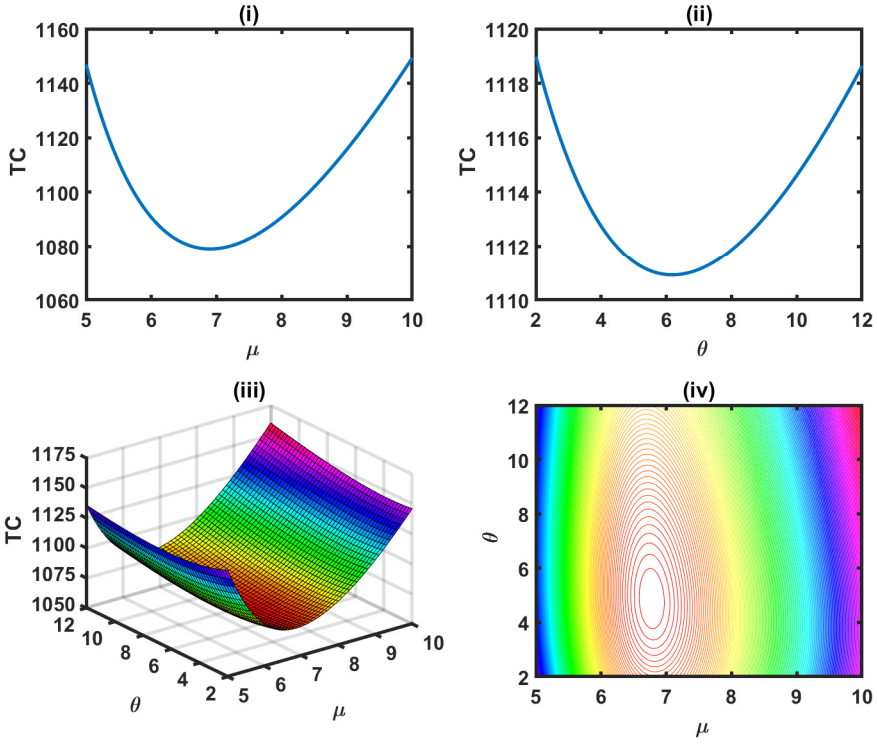


Figure 8.4: Plot of total cost (TC) wrt system's decision parameters (i) μ, (ii) θ as well their cumulative effect in (iii) surface plot and (iv) contour plot of TC for parameters values set.

8.6.1 Sensitivity Analysis

The SGO algorithm is coded in MATLAB R2020b, License Number 925317, and is run on a computer with the following specifications: Intel(R) Xeon(R) CPU E3-1231 v3 @ 3.40 GHz, 3401 MHz, with 4 Core(s), 8 Logical Processor(s), and 32 GB RAM. It takes around 3 seconds to reach the optimized solution after 30 iterations. The convergence result is demonstrated in Fig. 8.5. We did a sensitivity analysis on the parameter values to find out how each parameter affected the optimal solution. Fig. 8.4 shows that the function is unimodal over the set of parameters given in Table 8.2. The optimal service rate and searching rate are $\mu^* = 6.794699$ and $\theta^* = 4.814927$, respectively, and the minimal total cost rate is $TC^* = 1070.607677$.

Tables 8.3 and 8.4 show how the total cost TC^* and the critical parameters μ^* and θ^* change depending on the system parameters and the cost elements in the total cost function. Here, variations in parameter values are considered by altering them one at a time to test the effects on optimization results. The optimal parameters (μ^*, θ^*) that lead to the lowest system cost (TC^*) are summed up for different parameter sets (Table 8.3) and cost sets (Table 8.4). The computational times reported in Tables 8.3 and 8.4 are for an average of 20 runs each for different scenarios by perturbing different parameters in the model. Fig. 8.6 shows the convergence of the objective function TC during the minimization efforts of the SGO algorithm. The figure shows that after 10 iterations, convergence to the optimal solution (μ^*, θ^*, TC^*) is achieved.

8.7 Conclusion

We addressed the orbital search mechanism in the retrial queue problem in this chapter, incorporating multiple vacation policy and server breakdown. The consideration of all such phenomena in the outlined model makes it more realistic and applicable. We emphasize in analytically determining the probabilities of the server's state using the probability generating functions technique, so that it can be deduced that the server is in a specific state and in what proportion, and how parameters play a critical role in the change of its proportion. We further use these probabilities to find several system characteristics and the total cost function formulation. These performance measures of the system are enriched with numerical experiments against several parameters. We have employed a newly developed meta-heuristic optimization technique, SGO algorithm, to obtain the optimality of the total cost function and critical parameters of the system. The evaluation of the optimum system cost and optimal design parameters allows decision makers to solve their techno-economic

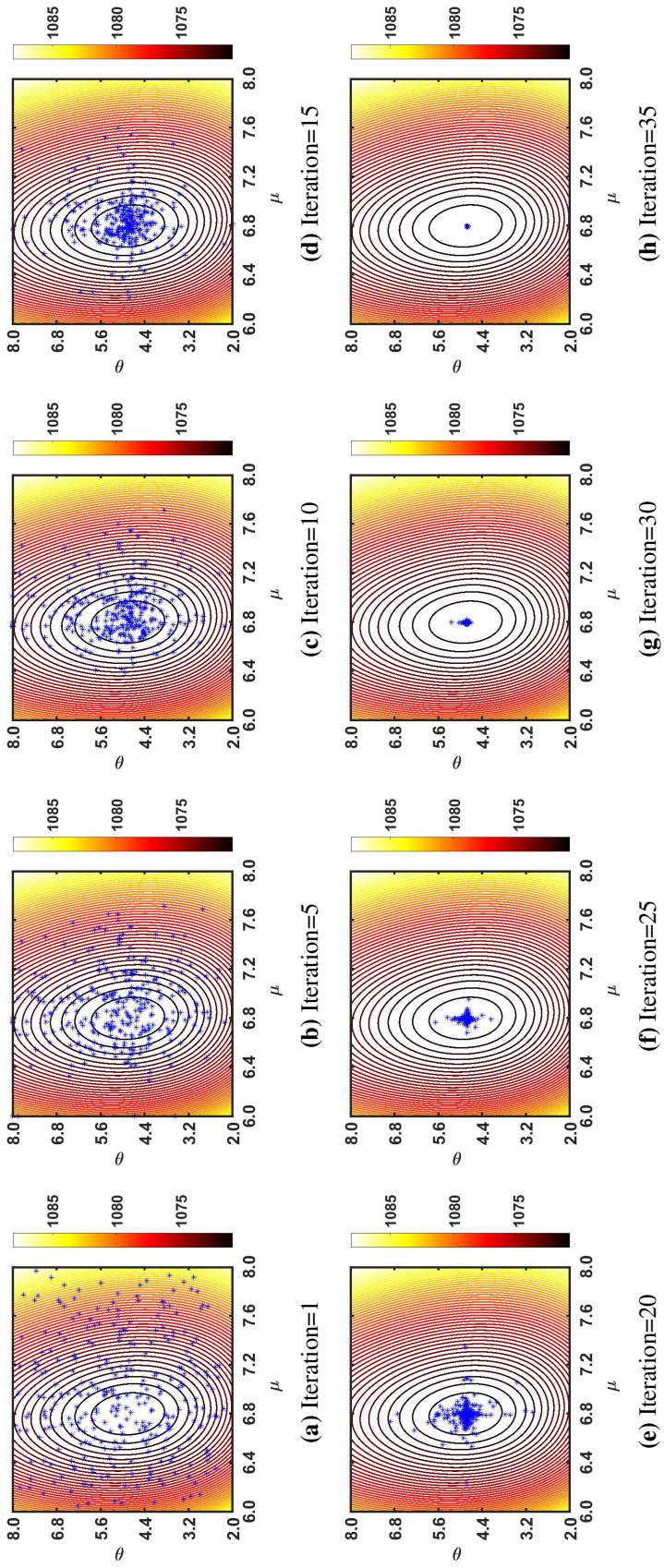


Figure 8.5: Several generations of SGO algorithm on the contour of $TC(\mu, \theta)$

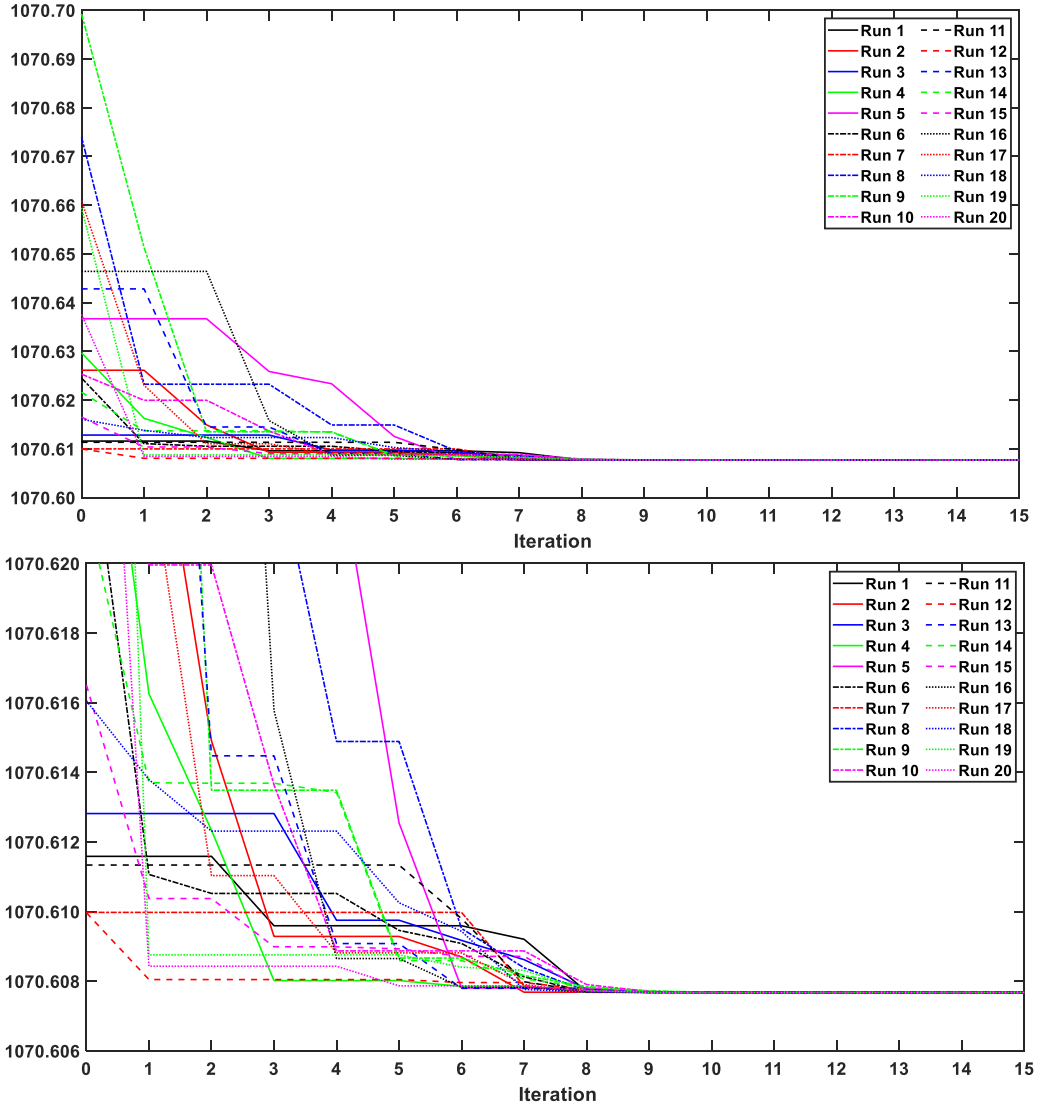


Figure 8.6: Convergence of iteration of SGO algorithm

Table 8.3: Optimal expected total cost of the system TC^* for different parameters via SGO algorithm.

$(\lambda, \Gamma, \varsigma, \vartheta, \delta, \upsilon, p, q_0, q_1, q_2)$	μ^*	θ^*	TC^*	**Mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Time elapsed**
0.8,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.054598	3.608207	977.947464	4.862363	2.826474	0.126503
1.0,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.794699	4.814927	1070.607677	8.384675	6.876443	0.117832
1.2,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.7	7.541015	6.004055	1163.185867	3.863732	5.896533	0.1119232
1,6.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.83664	5.903697	1085.679122	1.983753	4.826183	0.1455267
1,8.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.761881	3.745661	1057.859047	3.297321	5.872642	0.125207
1,7.5,0.6,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.300503	3.165671	1006.487588	7.983276	5.864224	0.106893
1,7.5,1.0,1.5,0.1,1.3,0.55,0.9,0.8,0.7	6.915105	5.257191	1088.788359	6.368224	2.624324	0.128373
1,7.5,0.8,1.0,0.1,1.3,0.55,0.9,0.8,0.7	6.875686	2.931765	1112.417724	5.876242	1.983652	0.108706
1,7.5,0.8,2.0,0.1,1.3,0.55,0.9,0.8,0.7	6.805977	5.022871	1071.540459	3.983465	2.986432	0.125511
1,7.5,0.8,1.5,0.05,1.3,0.55,0.9,0.8,0.7	6.608612	4.662149	1039.114425	4.872642	5.286421	0.111441
1,7.5,0.8,1.5,0.15,1.3,0.55,0.9,0.8,0.7	7.157941	5.098816	1130.544147	7.765324	9.876542	0.113357
1,7.5,0.8,1.5,0.1,1.0,0.55,0.9,0.8,0.7	6.373244	4.646756	1018.008588	4.564983	3.543211	0.107314
1,7.5,0.8,1.5,0.1,1.6,0.55,0.9,0.8,0.7	7.202564	4.972857	1119.625729	1.986342	2.863598	0.117979
1,7.5,0.8,1.5,0.1,1.3,0.45,0.9,0.8,0.7	6.842897	3.633838	1076.757777	5.764251	4.876423	0.115931
1,7.5,0.8,1.5,0.1,1.3,0.65,0.9,0.8,0.7	6.745445	5.864409	1063.635299	2.837621	1.973525	0.111685
1,7.5,0.8,1.5,0.1,1.3,0.55,0.85,0.8,0.7	6.716718	4.747265	1065.986567	3.983675	5.895367	0.108005
1,7.5,0.8,1.5,0.1,1.3,0.55,0.95,0.8,0.7	6.872896	4.882296	1075.399085	1.903875	2.983275	0.104378
1,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.75,0.7	6.801201	4.842687	1072.022748	1.976532	5.982548	0.118072
1,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.85,0.7	6.818757	3.592338	1074.780668	9.986354	4.938763	0.122646
1,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.6	6.506182	4.392036	1019.527885	2.826411	3.932865	0.136959
1,7.5,0.8,1.5,0.1,1.3,0.55,0.9,0.8,0.8	7.08789	5.244814	1122.412837	1.937658	4.876322	0.132411

Table 8.4: Optimal expected total cost of the system TC^* for different parameters via SGO algorithm.

$(C_h,C_v,C_m,C_b,C_r,C_s,C_w,C_i,C_\theta)$	μ^*	θ^*	TC^*	**Mean** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Max** $\frac{TC_i}{TC^*}$ $(*10^{-10}+1)$	**Time elapsed**
200,30,60,150,100,200,250,300,4.5	6.584186	4.436318	1023.697680	1.839765	1.435765	0.114952
250,30,60,150,100,200,250,300,4.5	6.794699	4.814927	1070.607677	8.384675	6.876443	0.117832
300,30,60,150,100,200,250,300,4.5	6.995489	5.188646	1116.345152	1.984356	4.873643	0.107139
250,20,60,150,100,200,250,300,4.5	6.794911	4.816071	1070.002117	2.843908	5.876343	0.113397
250,40,60,150,100,200,250,300,4.5	6.794487	4.813785	1071.213236	1.635479	8.276543	0.115614
250,30,40,150,100,200,250,300,4.5	7.942098	4.269025	924.403441	5.872454	6.873265	0.126009
250,30,80,150,100,200,250,300,4.5	6.111857	5.292065	1199.181856	7.398653	1.986353	0.121966
250,30,60,100,100,200,250,300,4.5	6.803204	4.860492	1046.383546	2.836573	3.863289	0.117981
250,30,60,200,100,200,250,300,4.5	6.786234	4.769014	1094.828281	4.782564	5.863494	0.111444
250,30,60,150,50,200,250,300,4.5	6.802068	4.854437	1049.613633	1.975487	3.872542	0.111135
250,30,60,150,150,200,250,300,4.5	6.787361	4.775157	1091.599072	9.276438	1.837629	0.115411
250,30,60,150,100,100,250,300,4.5	6.781481	4.640743	1069.053685	5.892633	4.218641	0.122857
250,30,60,150,100,300,250,300,4.5	6.807816	4.985422	1072.134526	2.974532	8.746502	0.122444
250,30,60,150,100,200,200,300,4.5	6.514314	4.223871	1012.620848	4.986421	3.826392	0.109678
250,30,60,150,100,200,300,300,4.5	7.059621	5.365571	1126.461494	5.389652	1.987522	0.114902
250,30,60,150,100,200,250,200,4.5	6.774203	4.805069	1068.628506	6.896422	4.782654	0.115293
250,30,60,150,100,200,250,400,4.5	6.815471	4.824827	1072.576482	7.863243	6.864892	0.107526
250,30,60,150,100,200,250,300,3.5	6.767742	6.529641	1064.989188	2.893652	1.986583	0.109398
250,30,60,150,100,200,250,300,5.5	6.818757	3.592338	1074.780668	3.743651	5.753248	0.122645

problems with the system in an efficient manner. The optimal results are summarized in tabular form for various sets of parameters and costs. Future research directions are to include the following considerations: (i) The concept of hybrid vacation, which combines working vacation and complete vacation, can be used to develop and integrate new features for possible extension in the outlined model. (ii) In order to model an actual service system as accurately as possible, one can consider the case where the service, retrial, search, repair, and vacation times follow a general (non-Markovian) distribution. (iii) Moreover, retrial queues with many servers are more realistic in practice.

Chapter 9

Conclusions and Future Work

In this chapter, the main outcomes of the thesis are encapsulated. Further, few research directions are also provided that may be studied in the future course of research work.

9.1 Summary and Conclusions

The main objective of the thesis was framed as the study of queue-based service systems with distinct waiting variants. This thesis contributes to the descriptive modeling of some queueing systems with various phenomena like customer impatience behavior, single and multiple vacation policies, arrival control policies, homogenous and heterogenous servers, and some others. Queueing is a mechanism used to handle congestion, and congestion is a natural phenomenon in everyday life. The queueing models can approximate realistic situations with accurate predictions of performance measures and are useful for evaluation, control, and monitoring of the systems. This thesis is divided into nine chapters:

Chapter 1 of this thesis deals with brief introduction of the service models and motivation of the present work. The review of customer-oriented, server-oriented, and system-oriented queueing models is presented in order to highlight the present work in its right perspective. The modeling of these systems is highly influenced by the characterization of the constituent processes, namely, arrival and service processes. The chapter-wise outline of the thesis is also presented.

Chapter 2 deals with the analysis and development of a multi-server queueing model incorporating three types of impatience attributes of customers: balking, reneging, and jockeying. Upon arrival, strategic customers initially either balk or join one of the multi-queues selectively and decide at subsequent arrival and departure epochs whether to renege or jockey in a probabilistic manner with the aim of early service or reducing expected waiting time. The study of systems integrating customers' attributes is motivated by observing real service

systems where these queueing occurrences interact. The proposed model contemplates the existence of impatient customers within a classical queueing system.

Chapter 3 investigated a finite-capacity service system with heterogeneous servers of two types as subordinate servers and a chief server arranged in tandem. Subordinate servers contribute in the initial phase of service, and the chief server completes the remaining service in the final phase. The service provided by the subordinate-chief server may be unreliable, which means the service may be repeatedly unsuccessful before it is successful. The current service strategy studied is applicable for various managerial systems like application form approval systems, amateur-expert systems, and call centers that operateunder a policy according to which customers are not allowed to approach the chief server of the system directly.

Chapter 4 deals with the analysis and development of a two-phase service system. The arriving customers in the first phase can either join the queue and wait for their turn or directly seek service through the online application. In the next phase, the customers, through both modes, must be physically present in the system. The controllable online booking is conceptualized for the online application users as, after a specific threshold limit, the online application customers will not be able to book online due to capacity constraints to benefit the customers waiting physically. There is a general tendency among the waiting customers to abandon the long queue in the first phase. The numerical simulation results provide important insights into the complex interactions between the parameters and the critical performance measures of the system. The findings confirmed that the balking strategies of customers negatively impact the system's throughput, and the admission control policy, i.e., the F-policy, helps service providers reduce the congestion level.

Chapter 5 focused on optimal policies for a highly efficient service system since the congestion of customers more often originates from degraded policies than faulty arrangements. In this chapter, we presented a notion of unreliable service and an F-policy for stochastic modeling of a finite-capacity customer service system. This study captures diversified service characteristics, customer behavior, and performability measures. The results indicate that preventive and corrective actions are crucial for a better service system. Our inferences also demonstrate that the optimization approach and stochastic modeling are practical ways to ensure efficient policies and optimize performance for the studied service model.

Chapter 6 analyzed a finite capacity service system with several realistic customer-server phenomena: customer impatience, server's partial breakdown, and threshold recovery policy. When the number of customers is more, the server is under pressure to increase the service rate to reduce the service system's load. Motivating from this fact, the concept of service pressure condition is also incorporated. This study is mainly based on efficient

resource utilization of real-life queueing-based service systems. This research provides essential theoretical and practical contributions to service systems that can be replicated in an organization with limited resources facing the challenge of queues. As a practical aspect, the insights derived from this study can help decision-makers take the necessary actions to reduce the overall cost of the service systems.

Chapter 7 deals with the critical issue of the single-server congestion problem withpr ominent customer impatience attributes and server strategic differentiated vacation. Despite their apparent practical relevance, the proposed congestion problem has yet to be studied from a service/production perspective with transient analysis.The queue-theoretic approach is used for mathematical modeling. The transient queue-size distribution has been derived using a modified Bessel function and generating function technique. A time-dependent solution is advantageous for queueing systems' dynamic behavior over a planning phase and is predominantly valuable within the real-time design process for the the-state-of-the-art strategic system.

Chapter 8 presents the orbital search concept in Markovian retrial queueing model, including multiple vacation policies, and server breakdown. The consideration of all such phenomena in the outlined model makes it more realistic and applicable. We emphasize in analytically determining the probabilities of the server's state using the probability generating functions technique, so that it can be deduced that the server is in a specific state and in what proportion, and how parameters play a critical role in the change of its proportion.

9.2 Contributions Through this Research

The major findings of the present study are highlighted below.

- The present study is mainly based on efficient resource utilization of real-life queueing-based service systems. This research provides essential theoretical and practical contributions to service systems that can be replicated in an organization with limited resources facing the challenge of queues. As a practical aspect, the insights derived from this study can help decision-makers take the necessary actions to reduce the overall cost of the service systems.

- This investigation demonstrated the dynamic congestion behavior of the customer in the planning phase.

- The steady-state and transient-state analytical results developed in this thesis would be useful to managers and system analysts in optimally allocating resources for system cost reduction. These transient and steady state probabilities are helpful in the evaluation of the characteristic measure of the system.

- We computed the stationary distribution of various models using the repeated substitution approach and derived various system performances in vector form.

- We also formulated a cost function and defined the problem of cost minimization constraint in each model. The state-of-the-art analysis of the service system is optimal expected cost. We used several efficient meta-heuristic optimization algorithms to analyze the optimal values of decision parameters of the system with the optimal stability condition and a global minimum of the cost function. Finally, several numerical experiments have been included to demonstrate and attain optimal results. The cost analysis clearly communicates the validity and profitability of the established model. Minimizing the cost of service, a widely sought attribute of any firm, will benefit system designers and decision-makers.

9.3 Future Scope of the Present Research Work

A number of future researches that can be developed and/or integrated as an advancement of the present thesis work are listed below.

- We can extend the present study for the random processes like service times or arrival times follows a distribution of general nature despite exponential to fit practical systems better.

- An extension for a more realistic queueing system can be done by increasing the number of phases for service for more than two phases. In such a case, servers will render services in phases orderly as servers of the previous phase will act as a customer for the next phase of service. In such a service system, there will be several levels of servers before reaching the final phase of service which contributes to economic savings through maximum utilization of service providers and queue management of system.

- In addition to the substantial insights in this research, there are many other queueing notions, such as machine repair problems, working vacation, etc., that present future research opportunities for academics, managers, and policymakers.

Bibliography

[1] Y. M. Abdelradi, A. A. El-Sherif, and L. H. Afify, "A queueing theory approach to traffic offloading in heterogeneous cellular networks," *AEU-International Journal of Electronics and Communications*, vol. 139, p. 153 910, 2021.

[2] M. A. Abidini, O. Boxma, and J. Resing, "Analysis and optimization of vacation and polling models with retrials," *Performance Evaluation*, vol. 98, pp. 52–69, 2016.

[3] M. O. Abou El Ata and A. M. A. Hariri, "The $M/M/c/N$ queue with balking and reneging," *Computers & Operations Research*, vol. 19, no. 8, pp. 713–716, 1992.

[4] M. O. Abou El Ata and A. I. Shawky, "The single-server Markovian overflow queue with balking, reneging and an additional server for longer queues," *Microelectronics Reliability*, vol. 32, no. 10, pp. 1389–1394, 1992.

[5] I. J. Adan, J. Wessels, and W. H. M. Zijm, "Analysis of the asymmetric shortest queue problem," *Queueing Systems*, vol. 8, pp. 1–58, 1991.

[6] J. Aguila-Leon, C. Vargas-Salgado, C. Chiñas-Palacios, and D. Díaz-Bello, "Solar photovoltaic maximum power point tracking controller optimization using grey wolf optimizer: A performance comparison between bio-inspired and traditional algorithms," *Expert Systems with Applications*, vol. 211, p. 118 700, 2023.

[7] A. Ahuja, A. Jain, and M. Jain, "Finite population multi-server retrial queueing system with an optional service and balking," *International Journal of Computers and Applications*, vol. 41, no. 1, pp. 54–61, 2019.

[8] A. Ahuja, A. Jain, and M. Jain, "Transient analysis and anfis computing of unreliable single server queueing model with multiple stage service and functioning vacation," *Mathematics and Computers in Simulation*, vol. 192, pp. 464–490, 2022.

[9] R. O. Al-Seedy, A. El-Sherbiny, S. El-Shehawy, and S. Ammar, "Transient solution of the $M/M/c$ queue with balking and reneging," *Computers & Mathematics with Applications*, vol. 57, no. 8, pp. 1280–1285, 2009.

[10] F. Al Thobiani, S. Khatir, B. Benaissa, E. Ghandourah, S. Mirjalili, and M. A. Wahab, "A hybrid pso and grey wolf optimization algorithm for static and dynamic crack identification," *Theoretical and applied fracture mechanics*, vol. 118, p. 103 213, 2022.

[11] S. B. Alaoui, E. H. Tissir, and N. Chaibi, "Analysis and design of robust guaranteed cost active queue management," *Computer Communications*, vol. 159, pp. 124–132, 2020.

[12] S. I. Ammar, M. M. Helan, and F. T. Al Amri, "The busy period of an $M/M/1$ queue with balking and reneging," *Applied mathematical modelling*, vol. 37, no. 22, pp. 9223–9229, 2013.

[13] I. Arizono and Y. Takemoto, "Statistical mechanics approach for steady-state analysis in $M/M/s$ queueing system with balking," *Journal of Industrial & Management Optimization*, vol. 18, no. 1, p. 25, 2022.

[14] J. R. Artalejo and A. Gómez-Corral, "Retrial queueing systems," *Mathematical and Computer Modelling*, vol. 30, no. 3-4, pp. xiii–xv, 1999.

[15] J. Artalejo, V. Joshua, and A Krishnamoorthy, "An $M/G/1$ retrial queue with orbital search by the server," *Advances in stochastic modelling*, pp. 41–54, 2002.

[16] G Ayyappan and S Karpagam, "Analysis of a bulk service queue with unreliable server, multiple vacation, overloading and stand-by server," *International Journal of Mathematics in Operational Research*, vol. 16, no. 3, pp. 291–315, 2020.

[17] G Ayyappan and M Nirmala, "An $M^{[X]}/G(a,b)/1$ queue with unreliable server, second optional service, closedown, setup with n-policy and multiple vacation," *International Journal of Mathematics in Operational Research*, vol. 16, no. 1, pp. 53–81, 2020.

[18] S. P. Bala Murugan and R Keerthana, "An $M/G/1$ feedback retrial queue with working vacation and a waiting server.," *Journal of Computational Analysis & Applications*, vol. 31, no. 1, 2023.

[19] S. Barile and F. Polese, "Smart service systems and viable service systems: Applying systems theory to service science," *Service Science*, vol. 2, no. 1-2, pp. 21–40, 2010.

[20] R. J. Batt and C. Terwiesch, "Waiting patiently: An empirical study of queue abandonment in an emergency department," *Management Science*, vol. 61, no. 1, pp. 39–59, 2015.

[21] E. Bolandifar, N. DeHoratius, T. Lennon Olsen, and J. L. Wiler, "Modeling the behavior of patients who leave the emergency department without being seen," *Chicago Booth Research Paper*, no. 14-12, pp. 430–446, 2016.

[22] M. Boualem, A. Bareche, and M. Cherfaoui, "Approximate controllability of stochastic bounds of stationary distribution of an m/g/1 queue with repeated attempts and two-phase service," *International Journal of Management Science and Engineering Management*, vol. 14, no. 2, pp. 79–85, 2019.

[23] A. A. Bouchentouf, M. Cherfaoui, and M. Boualem, "Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers," *Opsearch*, vol. 56, no. 1, pp. 300–323, 2019.

[24] A. A. Bouchentouf and A. Guendouzi, "The $M^X/M/c$ bernoulli feedback queue with variant multiple working vacations and impatient customers: Performance and economic analysis," *Arabian Journal of Mathematics*, vol. 9, no. 2, pp. 309–327, 2020.

[25] O. Boudali and A. Economou, "Optimal and equilibrium balking strategies in the single server markovian queue with catastrophes," *European Journal of Operational Research*, vol. 218, no. 3, pp. 708–715, 2012.

[26] M. Bourreau and F. M. Manenti, "Selling cross-border in online markets: The impact of the ban on geoblocking strategies," *International Journal of Industrial Organization*, p. 102 892, 2022.

[27] A. Burnetas and A. Economou, "Equilibrium customer strategies in a single server markovian queue with setup times," *Queueing Systems*, vol. 56, no. 3, pp. 213–228, 2007.

[28] A. Burnetas, A. Economou, and G. Vasiliadis, "Strategic customer behavior in a queueing system with delayed observations," *Queueing Systems*, vol. 86, no. 3, pp. 389–418, 2017.

[29] A. E. Chaleshtori, H. Jahani, and A. Aghaie, "Bi-objective optimization approach to a multi-layer location–allocation problem with jockeying," *Computers & Industrial Engineering*, vol. 149, p. 106 740, 2020.

[30] X. Chen, B. Xu, K. Yu, and W. Du, "Teaching-learning-based optimization with learning enthusiasm mechanism and its application in chemical engineering," *Journal of Applied Mathematics*, vol. 2018, 2018.

[31] A Chouar, S Tetouani, A Soulhi, and J Elalami, "Performance improvement in physical internet supply chain network using hybrid framework," *IFAC-PapersOnLine*, vol. 54, no. 13, pp. 593–598, 2021.

[32] G. Choudhury and K. Deka, "An $M/G/1$ retrial queueing system with two phases of service subject to the server breakdown and repair," *Performance Evaluation*, vol. 65, no. 10, pp. 714–724, 2008.

[33] G. Choudhury and M. Deka, "A single server queueing system with two phases of service subject to server breakdown and bernoulli vacation," *Applied Mathematical Modelling*, vol. 36, no. 12, pp. 6050–6060, 2012.

[34] G. Choudhury and M. Deka, "A batch arrival unreliable server delaying repair queue with two phases of service and bernoulli vacation under multiple vacation policy," *Quality Technology & Quantitative Management*, vol. 15, no. 2, pp. 157–186, 2018.

[35] G. Choudhury, L. Tadj, and M. Paul, "Steady state analysis of an mx/g/1 queue with two phase service and bernoulli vacation schedule under multiple vacation policy," *Applied Mathematical Modelling*, vol. 31, no. 6, pp. 1079–1091, 2007.

[36] E. Cinlar, *Introduction to stochastic processes*. Courier Corporation, 2013.

[37] J. W. Cohen, *The single server queue*. Elsevier, 2012.

[38] A. E. Conway and N. D. Georganas, *Queueing networks—exact computational algorithms: a unified theory based on decomposition and aggregation*. Mit Press, 1989.

[39] T. Deepak, A. Dudin, V. Joshua, and A Krishnamoorthy, "On an $M^{(X)}/G/1$ retrial system with two types of search of customers from the orbit," *Stochastic Analysis and Applications*, vol. 31, no. 1, pp. 92–107, 2013.

[40] A. Dehghanian, J. P. Kharoufeh, and M. Modarres, "Strategic dynamic jockeying between two parallel queues," *Probability in the Engineering and Informational Sciences*, vol. 30, no. 1, pp. 41–60, 2016.

[41] A. Delavarkhalafi, "On optimal stochastic jumps in multi server queue with impatient customers via stochastic control," *Numerical Algebra, Control and Optimization*, vol. 12, no. 4, pp. 693–703, 2022.

[42] I. Dimitriou, "A queueing model with two classes of retrial customers and paired services," *Annals of Operations Research*, vol. 238, pp. 123–143, 2016.

[43] B. Doshi, "Analysis of a two phase queueing system with general service times," *Operations research letters*, vol. 10, no. 5, pp. 265–272, 1991.

[44] B. Doshi, *Queuing systems with vacations: A survey queuing system*, 1986.

[45] S. Drekic and D. G. Woolford, "A preemptive priority queue with balking," *European Journal of Operational Research*, vol. 164, no. 2, pp. 387–401, 2005.

[46] A. N. Dudin, A. Krishnamoorthy, V. Joshua, and G. V. Tsarenkov, "Analysis of the $BMAP/G/1$ retrial system with search of customers from the orbit," *European Journal of Operational Research*, vol. 157, no. 1, pp. 169–179, 2004.

[47] A. Economou, "How much information should be given to the strategic customers of a queueing system?" *Queueing Systems*, pp. 1–3, 2022.

[48] A. Economou, A. Gómez-Corral, and S. Kanta, "Optimal balking strategies in single-server queues with general service and vacation times," *Performance Evaluation*, vol. 68, no. 10, pp. 967–982, 2011.

[49] A. Economou and S. Kanta, "Equilibrium balking strategies in the observable single-server queue with breakdowns and repairs," *Operations Research Letters*, vol. 36, no. 6, pp. 696–699, 2008.

[50] A. Economou and S. Kanta, "Optimal balking strategies and pricing for the single server markovian queue with compartmented waiting space," *Queueing Systems*, vol. 59, no. 3, pp. 237–269, 2008.

[51] A. Economou and S. Kanta, "Equilibrium customer strategies and social–profit maximization in the single-server constant retrial queue," *Naval Research Logistics (NRL)*, vol. 58, no. 2, pp. 107–122, 2011.

[52] A. Economou, D. Logothetis, and A. Manou, "The value of reneging for strategic customers in queueing systems with server vacations/failures," *European Journal of Operational Research*, 2022.

[53] D. V. Efrosinin and O. V. Semenova, "An $M/M/1$ system with an unreliable device and a threshold recovery policy," *Journal of Communications Technology and Electronics*, vol. 55, no. 12, pp. 1526–1531, 2010.

[54] T ENGSET, *The probability theory for computing the number of switching equipments in automatic telephone exchange, etz, 31, 304-305*, 1918.

[55] A. K. Erlang, "Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges," *Post Office Electrical Engineer's Journal*, vol. 10, pp. 189–197, 1917.

[56] M. A. Esfeh, S. Saidi, S. Wirasinghe, and L. Kattan, "Waiting time and headway modeling considering unreliability in transit service," *Transportation Research Part A: Policy and Practice*, vol. 155, pp. 219–233, 2022.

[57] G. Falin, "A survey of retrial queues," *Queueing systems*, vol. 7, pp. 127–167, 1990.

[58] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural computing and applications*, vol. 30, pp. 413–435, 2018.

[59] J. Ford and I. Moghrabi, "Multi-step quasi-newton methods for optimization," *Journal of Computational and Applied Mathematics*, vol. 50, no. 1-3, pp. 305–323, 1994.

[60] A. Furnham, L. Treglown, and G. Horne, "The psychology of queuing," 2020.

[61] S. Gao, H. Dong, and X. Wang, "Equilibrium and pricing analysis for an unreliable retrial queue with limited idle period and single vacation," *Operational Research*, vol. 21, pp. 621–643, 2021.

[62] S. Gao and J. Wang, "Performance and reliability analysis of an $M/G/1-G$ retrial queue with orbital search and non-persistent customers," *European Journal of Operational Research*, vol. 236, no. 2, pp. 561–572, 2014.

[63] S. Gao and J. Wang, "Stochastic analysis of a preemptive retrial queue with orbital search and multiple vacations," *RAIRO-Operations Research*, vol. 54, no. 1, pp. 231–249, 2020.

[64] S. Gao and J. Zhang, "Strategic joining and pricing policies in a retrial queue with orbital search and its application to call centers," *IEEE Access*, vol. 7, pp. 129 317–129 326, 2019.

[65] S. Gao, J. Zhang, and X. Wang, "Analysis of a retrial queue with two-type breakdowns and delayed repairs," *IEEE Access*, vol. 8, pp. 172 428–172 442, 2020.

[66] C. Gautam, K. Priyanka, and S. Dharmaraja, "Analysis of a model of batch arrival single server queue with random vacation policy," *Communications in Statistics-Theory and Methods*, pp. 1–44, 2020.

[67] M. Ghalambaz, R. J. Yengejeh, and A. H. Davami, "Building energy optimization using grey wolf optimizer (gwo)," *Case Studies in Thermal Engineering*, vol. 27, p. 101 250, 2021.

[68] N. Gore, S. Arkatkar, G. Joshi, and C. Antoniou, "Developing modified congestion index and congestion-based level of service," *Transport policy*, vol. 131, pp. 97–119, 2023.

[69] D. Gross, *Fundamentals of queueing theory*. John Wiley & Sons, 2008.

[70] D. Guha, V. Goswami, and A. Banik, "Algorithmic computation of steady-state probabilities in an almost observable $GI/M/c$ queue with or without vacations under state dependent balking and reneging," *Applied Mathematical Modelling*, vol. 40, no. 5-6, pp. 4199–4219, 2016.

[71] S. M. Gupta, "Interrelationship between controlling arrival and service in queueing systems," *Computers & operations research*, vol. 22, no. 10, pp. 1005–1014, 1995.

[72] F. A. Haight, "Queueing with balking," *Biometrika*, vol. 44, no. 3/4, pp. 360–369, 1957.

[73] F. A. Haight, "Two queues in parallel," *Biometrika*, vol. 45, no. 3-4, pp. 401–410, 1958.

[74] F. A. Haight, "Queueing with reneging," *Metrika*, vol. 2, no. 1, pp. 186–197, 1959.

[75] G. Hanukov, "A service system where junior servers approach a senior server on behalf of customers," *International Journal of Production Economics*, vol. 244, p. 108 351, 2022.

[76] G. Hanukov and U. Yechiali, "Individual and social customers' joining strategies in a two-stage service system when discount is offered to users of smartphone application," *Applied Mathematical Modelling*, vol. 105, pp. 355–374, 2022.

[77] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, pp. 1531–1551, 2021.

[78] B. A. Hassan and I. A. Moghrabi, "A modified secant equation quasi-newton method for unconstrained optimization," *Journal of Applied Mathematics and Computing*, vol. 69, no. 1, pp. 451–464, 2023.

[79] R. Hassin, *Rational queueing*. CRC press, 2016.

[80] R. Hassin and M. Haviv, "Equilibrium strategies for queues with impatient customers," *Operations Research Letters*, vol. 17, no. 1, pp. 41–45, 1995.

[81] R. Hassin and M. Haviv, *To queue or not to queue: Equilibrium behavior in queueing systems*. Springer Science & Business Media, 2003, vol. 59.

[82] F. S. Hillier, *Introduction to Operations Research*. Tata McGraw-Hill Education, 2012.

[83] P. Hoseinpour, "Improving service quality in a congested network with random breakdowns," *Computers & Industrial Engineering*, vol. 157, p. 107 226, 2021.

[84] P. Hoseinpour, "Modeling and solving an economies-of-scale service system design problem," *International Transactions in Operational Research*, 2022.

[85] L.-C. Hwang, "M-green: An active queue management mechanism for multi-qos classes," *Computer Standards & Interfaces*, vol. 36, no. 1, pp. 122–131, 2013.

[86] E. Indramaya and S. Suyanto, "Comparative study of recent swarm algorithms for continuous optimization," *Procedia Computer Science*, vol. 179, pp. 685–695, 2021.

[87] O. A. Isijola-Adakeja and O. C. Ibe, "$M/M/1$ Multiple vacation queueing systems with differentiated vacations and vacation interruptions," *IEEE Access*, vol. 2, pp. 1384–1395, 2014.

[88] J. R. Jackson, "Networks of waiting lines," *Operations research*, vol. 5, no. 4, pp. 518–521, 1957.

[89] M. Jain and A. Bhagat, "Finite population retrial queueing model with threshold recovery, geometric arrivals and impatient customers," *Journal of Information and Operations Management*, vol. 3, no. 1, p. 162, 2012.

[90] M. Jain, A. Bhagat, and C. Shekhar, "Double orbit finite retrial queues with priority customers and service interruptions," *Applied Mathematics and Computation*, vol. 253, pp. 324–344, 2015.

[91] M. Jain, S. Dhibar, and S. S. Sanga, "Markovian working vacation queue with imperfect service, balking and retrial," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 4, pp. 1907–1923, 2022.

[92] M. Jain and S. S. Sanga, "Admission control for finite capacity queueing model with general retrial times and state-dependent rates," *Journal of Industrial and Management Optimization*, vol. 16, no. 6, pp. 2625–2649, 2020.

[93] M. Jain and S. S. Sanga, "State dependent queueing models under admission control $F-$policy: A survey," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 9, pp. 3873–3891, 2020.

[94] J. J. Jena and S. C. Satapathy, "A new adaptive tuned social group optimization (SGO) algorithm with sigmoid-adaptive inertia weight for solving engineering design problems," *Multimedia Tools and Applications*, pp. 1–35, 2021.

[95] T. Jorgensen Jr, "On probability generating functions," *American Journal of Physics*, vol. 16, no. 5, pp. 285–289, 1948.

[96] K. Kalidass and R. Kasturi, "A queue with working breakdowns," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 779–783, 2012.

[97] C. Kao, W. T. Song, and S.-P. Chen, "A modified quasi-newton method for optimization in simulation," *International Transactions in Operational Research*, vol. 4, no. 3, pp. 223–233, 1997.

[98] U. P. Karupothu and P. Kumar, "Perceptionization of fm/fd/1 queuing model under various fuzzy numbers," *Croatian Operational Research Review*, pp. 135–144, 2020.

[99] J.-C. Ke, C.-J. Chang, and F.-M. Chang, "Controlling arrivals for a markovian queueing system with a second optional service," *International Journal of Industrial Engineering*, vol. 17, no. 1, pp. 48–57, 2010.

[100] J.-C. Ke, T.-H. Liu, S. Su, and Z.-G. Zhang, "On retrial queue with customer balking and feedback subject to server breakdowns," *Communications in Statistics-Theory and Methods*, vol. 51, no. 17, pp. 6049–6063, 2022.

[101] J.-C. Ke, C.-H. Wu, and W. L. Pearn, "Analysis of an infinite multi-server queue with an optional service," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 216–225, 2013.

[102] W. M. Kempa and R. Marjasz, "Distribution of the time to buffer overflow in the $M/G/1/N$-type queueing model with batch arrivals and multiple vacation policy," *Journal of the Operational Research Society*, vol. 71, no. 3, pp. 447–455, 2020.

[103] D. G. Kendall, "Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain," *The Annals of Mathematical Statistics*, pp. 338–354, 1953.

[104] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, IEEE, vol. 4, 1995, pp. 1942–1948.

[105] Y. Kerner, "Equilibrium joining probabilities for an $M/G/1$ queue," *Games and Economic Behavior*, vol. 71, no. 2, pp. 521–526, 2011.

[106] J. Kim and B. Kim, "A survey of retrial queueing systems," *Annals of operations research*, vol. 247, pp. 3–36, 2016.

[107] L. Kleinrock, *Theory, volume 1, queueing systems*, 1975.

[108] K. Kotb and H. A. El-Ashkar, "Quality control for feedback $M/M/1/N$ queue with balking and retention of reneged customers," *Filomat*, vol. 34, no. 1, pp. 167–174, 2020.

[109] C. Krishna and Y.-H. Lee, "A study of two-phase service," *Operations Research Letters*, vol. 9, no. 2, pp. 91–97, 1990.

[110] B Krishna Kumar, R Rukmani, A Thanikachalam, and V Kanakasabapathi, "Performance analysis of retrial queue with server subject to two types of breakdowns and repairs," *Operational research*, vol. 18, pp. 521–559, 2018.

[111] A. Krishnamoorthy, P. K. Pramod, and S. R. Chakravarthy, "Queues with interruptions: A survey," *Top*, vol. 22, no. 1, pp. 290–320, 2014.

[112] A. Kumar, "Single server multiple vacation queue with discouragement solve by confluent hypergeometric function," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.

[113] A. Kumar and M. Jain, "Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy," *Mathematics and Computers in Simulation*, vol. 204, pp. 259–281, 2023.

[114] R. Kumar, "Economic analysis of an $M/M/c/N$ queuing model with balking, reneging and retention of reneged customers," *Opsearch*, vol. 50, no. 3, pp. 383–403, 2013.

[115] R. Kumar, S. Sharma, and V. Rykov, "Transient solution of a heterogeneous queuing system with balking and retention of reneging customers," in *International Conference on Computer Networks*, Springer, 2019, pp. 330–346.

[116] P. V. Laxmi and A. A. George, "Transient and steady state analysis of m/mˆ([b])/1 queue with second optional service," *Journal of Industrial and Production Engineering*, pp. 1–11, 2021.

[117] P. V. Laxmi and T. W. Kassahun, "Transient analysis of multi-server markovian queueing system with synchronous multiple working vacations and impatience of customers," *International Journal of Mathematics in Operational Research*, vol. 16, no. 2, pp. 217–237, 2020.

[118] Y. Levy and U. Yechiali, "Utilization of idle time in an $M/G/1$ queueing system," *Management Science*, vol. 22, no. 2, pp. 202–211, 1975.

[119] L. Li, J. Wang, and F. Zhang, "Equilibrium customer strategies in Markovian queues with partial breakdowns," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 751–757, 2013.

[120] L. Li, Y. He, H. Zhang, J. C. Fung, and A. K. Lau, "Enhancing iaq, thermal comfort, and energy efficiency through an adaptive multi-objective particle swarm optimizer-grey wolf optimization algorithm for smart environmental control," *Building and Environment*, vol. 235, p. 110 235, 2023.

[121] G. Lindfield and J. Penny, *Introduction to Nature-Inspired Optimization*. Academic Press, 2017.

[122] C. D. Liou, "Markovian queue optimisation analysis with an unreliable server subject to working breakdowns and impatient customers," *International Journal of Systems Science*, vol. 46, no. 12, pp. 2165–2182, 2015.

[123] Y. Liu, D. Wu, W. Zhou, K. Fan, and Z. Zhou, "EACP: An effective automatic channel pruning for neural networks," *Neurocomputing*, 2023.

[124] Z. Liu and Y. Song, "The $M^X/M/1$ queue with working breakdown," *RAIRO-Operations Research*, vol. 48, no. 3, pp. 399–413, 2014.

[125] M. Lozano and P. Moreno, "A discrete time single-server queue with balking: Economic applications," *Applied Economics*, vol. 40, no. 6, pp. 735–748, 2008.

[126] V. R. Lumb and I. Rani, "Analytically simple solution to discrete-time queue with catastrophes, balking and state-dependent service," *International Journal of System Assurance Engineering and Management*, vol. 13, no. 2, pp. 783–817, 2022.

[127] K. C. Madan, "On a single server queue with two-stage heterogeneous service and deterministic server vacations," *International Journal of Systems Science*, vol. 32, no. 7, pp. 837–844, 2001.

[128] S. P. Madheswari, B. K. Kumar, and P. Suganthi, "Analysis of $M/G/1$ retrial queues with second optional service and customer balking under two types of bernoulli vacation schedule," *RAIRO-Operations Research*, vol. 53, no. 2, pp. 415–443, 2019.

[129] P. P. Maglio and J. Spohrer, "Fundamentals of service science," *Journal of the academy of marketing science*, vol. 36, pp. 18–20, 2008.

[130] A. Mandelbaum and N. Shimkin, "A model for rational abandonments from invisible queues," *Queueing Systems*, vol. 36, no. 1, pp. 141–173, 2000.

[131] P Manoharan and S Subathra, "Non markovian retrial queue, balking, disaster under working breakdown and working vacation.," *Journal of Computational Analysis & Applications*, vol. 31, no. 1, 2023.

[132] E. B. McBride, *Obtaining Generating Functions*. Springer Science & Business Media, 2012, vol. 21.

[133] R. K. Meena and P. Kumar, "Performance analysis of markov retrial queueing model under admission control $F-$policy," in *Mathematical Modeling and Computation of Real-Time Problems*, CRC Press, 2021, pp. 65–78.

[134] C. Mele, T. Tuominen, B. Edvardsson, and J. Reynoso, "Smart sensing technology and self-adjustment in service systems through value co-creation routine dynamics," *Journal of Business Research*, vol. 159, p. 113 737, 2023.

[135] H. Mendelson and U. Yechiali, "Controlling the $GI/M/1$ queue by conditional acceptance of customers," *European Journal of Operational Research*, vol. 7, no. 1, pp. 77–85, 1981.

[136] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[137] P. Mishra and A. Moustafa, "Reinforcement learning based monotonic policy for online resource allocation," *Future Generation Computer Systems*, vol. 138, pp. 313–327, 2023.

[138] A. Naik, S. C. Satapathy, and A. Abraham, "Modified social group optimization—a meta-heuristic algorithm to solve short-term hydrothermal scheduling," *Applied Soft Computing*, vol. 95, p. 106 524, 2020.

[139] A. Naik, S. C. Satapathy, A. S. Ashour, and N. Dey, "Social group optimization for global optimization of multimodal functions and data clustering problems," *Neural Computing and Applications*, vol. 30, pp. 271–287, 2018.

[140] P. Naor, "The regulation of queue size by levying tolls," *Econometrica: journal of the Econometric Society*, pp. 15–24, 1969.

[141] A. Negahban, "Estimating the true arrival, balking, and reneging processes from censored transactional data: A simulation-based approach," *SIMULATION*, vol. 98, no. 7, pp. 597–614, 2022.

[142] M. F. Neuts, "Matrix geometric solutions in stochastic models: An algorithmic approach," *Baltimore, MD, USA*, 1981.

[143] M. F. Neuts, "Matrix-analytic methods in queuing theory," *European Journal of Operational Research*, vol. 15, no. 1, pp. 2–12, 1984.

[144] M. F. Neuts, *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation, 1994.

[145] M. F. Neuts and M. Ramalhoto, "A service model in which the server is required to search for customers," *Journal of Applied Probability*, vol. 21, no. 1, pp. 157–166, 1984.

[146] M. Ozen and A. Krishnamurthy, "G-network models to support planning for disaster relief distribution," *International Journal of Production Research*, vol. 60, no. 5, pp. 1621–1632, 2022.

[147] C Palm, "Research on telephone traffic carried by full availability groups, tele, 1 (107) pp," *English translation of results first published in*, vol. 7, 1946.

[148] J. Patterson and A. Korzeniowski, "M/m/1 model with unreliable service," *International Journal of Statistics and Probability*, vol. 7, no. 1, 2018.

[149] J. Patterson and A. Korzeniowski, "M/m/1 model with unreliable service and a working vacation," *International Journal of Statistics and Probability*, vol. 8, no. 2, 2019.

[150] J. Patterson, A. Korzeniowski, *et al.*, "Decomposition of m/m/1 with unreliable service and a working vacation," *International Journal of Statistics and Probability*, vol. 9, no. 1, pp. 1–63, 2020.

[151] E. Perel and U. Yechiali, "Queues where customers of one queue act as servers of the other queue," *Queueing Systems*, vol. 60, no. 3, pp. 271–288, 2008.

[152] E. Perel and U. Yechiali, "On customers acting as servers," *Asia-Pacific Journal of Operational Research*, vol. 30, no. 05, p. 1 350 019, 2013.

[153] T. Phung-Duc, "Retrial queueing models: A survey on theory and applications," *arXiv preprint arXiv:1906.09560*, 2019.

[154] K. U. Prameela and P. Kumar, "Conceptualization of finite capacity single-server queuing model with triangular, trapezoidal and hexagonal fuzzy numbers using α-cuts," in *Numerical Optimization in Engineering and Sciences: Select Proceedings of NOIEAS 2019*, Springer, 2020, pp. 201–212.

[155] P Rajadurai, V. Chandrasekaran, and M. Saravanarajan, "Analysis of an $M^{[X]}/G/1$ unreliable retrial G-queue with orbital search and feedback under bernoulli vacation schedule," *Opsearch*, vol. 53, pp. 197–223, 2016.

[156] P Rajadurai, M. Saravanarajan, and V. Chandrasekaran, "A study on $M/G/1$ feedback retrial queue with subject to server breakdown and repair under multiple working vacation policy," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 947–962, 2018.

[157] P. Rajadurai, "Sensitivity analysis of an $M/G/1$ retrial queueing system with disaster under working vacations and working breakdowns," *RAIRO-Operations Research*, vol. 52, no. 1, pp. 35–54, 2018.

[158] S. Rani, M. Jain, and R. K. Meena, "Queueing modeling and optimization of a fault-tolerant system with reboot, recovery, and vacationing server operating under admission control policy," *Mathematics and Computers in Simulation*, vol. 209, pp. 408–425, 2023.

[159] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-aided design*, vol. 43, no. 3, pp. 303–315, 2011.

[160] R. Ravid, "A new look on the shortest queue system with jockeying," *Probability in the Engineering and Informational Sciences*, vol. 35, no. 3, pp. 557–564, 2021.

[161] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Advances in engineering software*, vol. 105, pp. 30–47, 2017.

[162] K. Sasanuma, R. Hampshire, and A. Scheller-Wolf, "Controlling arrival and service rates to reduce sensitivity of queueing systems with customer abandonment," *Results in Control and Optimization*, vol. 6, p. 100 089, 2022.

[163] S. Satapathy and A. Naik, "Social group optimization (SGO): A new population evolutionary optimization technique," *Complex & Intelligent Systems*, vol. 2, no. 3, pp. 173–203, 2016.

[164] M Seenivasan and J Epciya, "$M/M/1$ retrial queueing model with server breakdown and feedback.," *Journal of Computational Analysis & Applications*, vol. 31, no. 1, 2023.

[165] W. P. Sendfeld, "Two-dimensional overflow queueing systems," 2009.

[166] C. Shekhar, M. Jain, and A. A. Raina, "Transient analysis of machining system with spare provisioning and geometric reneging," *International Journal of Mathematics in Operational Research*, vol. 11, no. 3, pp. 396–421, 2017.

[167] C. Shekhar, N. Kumar, A. Gupta, A. Kumar, and S. Varshney, "Warm-spare provisioning computing network with switching failure, common cause failure, vacation interruption, and synchronized reneging," *Reliability Engineering & System Safety*, vol. 199, p. 106 910, 2020.

[168] C. Shekhar, S. Varshney, and A. Kumar, "Optimal and sensitivity analysis of vacation queueing system with F-policy and vacation interruption," *Arabian Journal for Science and Engineering*, vol. 45, no. 8, pp. 7091–7107, 2020.

[169] C. Shekhar, S. Varshney, and A. Kumar, "Optimal control of a service system with emergency vacation using bat algorithm," *Journal of computational and applied mathematics*, vol. 364, p. 112 332, 2020.

[170] C. Shekhar, S. Varshney, and A. Kumar, "Matrix-geometric solution of multi-server queueing systems with bernoulli scheduled modified vacation and retention of reneged customers: A meta-heuristic approach," *Quality Technology & Quantitative Management*, vol. 18, no. 1, pp. 39–66, 2021.

[171] C. Shekhar, S. Varshney, and A. Kumar, "Standbys provisioning in machine repair problem with unreliable service and vacation interruption," in *The Handbook of Reliability, Maintenance, and System Safety through Mathematical Modeling*, Elsevier, 2021, pp. 101–133.

[172] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, IEEE, 1998, pp. 69–73.

[173] Y. W. Shin, "Algorithmic approach to markovian multi-server retrial queues with vacations," *Applied Mathematics and Computation*, vol. 250, pp. 287–297, 2015.

[174] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of queueing theory*. John Wiley & Sons, 2018, vol. 399.

[175] C. J. Singh, S. Kaur, and M. Jain, "Analysis of bulk queue with additional optional service, vacation and unreliable server," *International Journal of Mathematics in Operational Research*, vol. 14, no. 4, pp. 517–540, 2019.

[176] V. P. Singh, "Two-server markovian queues with balking: Heterogeneous vs. homogeneous servers," *Operations Research*, vol. 18, no. 1, pp. 145–159, 1970.

[177] B. K. Som, V. K. Sharma, and S. Seth, "An $M/M/2$ heterogeneous service markovian feedback queuing model with reverse balking, reneging and retention of reneged customers," in *Advances in Computing and Intelligent Systems*, Springer, 2020, pp. 291–296.

[178] V. Sridharan and P. J. Jayashree, "Some characteristics on a finite queue with normal partial and total failures," *Microelectronics Reliability*, vol. 36, no. 2, pp. 265–267, 1996.

[179] W. Sun, S. Li, and N. Tian, "Equilibrium and optimal balking strategies of customers in unobservable queues with double adaptive working vacations," *Quality Technology & Quantitative Management*, vol. 14, no. 1, pp. 94–113, 2017.

[180] H. Takagi and L. B. Boguslavsky, "A supplementary bibliography of books on queueing analysis and performance evaluation," *Queueing systems*, vol. 8, no. 1, pp. 313–322, 1991.

[181] Y. Takahashi, *Queueing analysis: A foundation of performance evaluation, volume 1: Vacation and priority systems, part 1: By h. takagi. elsevier science publishers, amsterdam, the netherlands, april 1991. isbn: 0-444-88910-8*, 1993.

[182] G. Tamrakar and A Banerjee, "On steady-state joint distribution of an infinite buffer batch service poisson queue with single and multiple vacation," *OPSEARCH*, pp. 1–37, 2020.

[183] A. M. Tarabia, "Analysis of two queues in parallel with jockeying and restricted capacities," *Applied Mathematical Modelling*, vol. 32, no. 5, pp. 802–810, 2008.

[184] N. Tian and Z. G. Zhang, *Vacation queueing models: theory and applications*. Springer Science & Business Media, 2006, vol. 93.

[185] H. C. Tijms, *Stochastic modelling and analysis: a computational approach*. John Wiley & Sons, Inc., 1986.

[186] H. C. Tijms, *Stochastic models: an algorithmic approach*. John Wiley & Sons Incorporated, 1994, vol. 303.

[187] Q. Tong, G. Liang, X. Cai, C. Zhu, and J. Bi, "Asynchronous parallel stochastic quasi-newton methods," *Parallel computing*, vol. 101, p. 102 721, 2021.

[188] S. Upadhyaya and C. Kushwaha, "Performance prediction and ANFIS computing for unreliable retrial queue with delayed repair under modified vacation policy," *International Journal of Mathematics in Operational Research*, vol. 17, no. 4, pp. 437–466, 2020.

[189] V. Varadharajan and K Ruth, "Two heterogenous parellel queues with customers act as servers and random set up time," *Indian Journal of Pure and Applied Mathematics*, vol. 117, pp. 21–27, May 2020.

[190] V Vijayalakshmi, K Kalidass, and B Deepa, "Cost analysis of m/m/1/n queue with working breakdowns and a two-phase services," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1850, 2021, p. 012 026.

[191] K. Vijayashree and B Janani, "Transient analysis of an $M/M/1$ queueing system subject to differentiated vacations," *Quality Technology & Quantitative Management*, vol. 15, no. 6, pp. 730–748, 2018.

[192] J. Wang and J. Li, "A repairable $M/G/1$ retrial queue with bernoulli vacation and two-phase service," *Quality Technology & Quantitative Management*, vol. 5, no. 2, pp. 179–192, 2008.

[193] J. Wang, F. Wang, J. Sztrik, and A. Kuki, "Finite source retrial queues with two phase service," *International Journal of Operational Research*, vol. 30, no. 4, pp. 421–440, 2017.

[194] J. Wang and F. Zhang, "Strategic joining in $M/M/1$ retrial queues," *European Journal of Operational Research*, vol. 230, no. 1, pp. 76–87, 2013.

[195] K.-H. Wang, C.-C. Kuo, and W. Pearn, "Optimal control of an $M/G/1/K$ queueing system with combined F policy and startup time," *Journal of Optimization Theory and Applications*, vol. 135, no. 2, pp. 285–299, 2007.

[196] K.-H. Wang, C.-C. Kuo, and W. Pearn, "A recursive method for the $F-$policy $G/M/1/K$ queueing system with an exponential startup time," *Applied mathematical modelling*, vol. 32, no. 6, pp. 958–970, 2008.

[197] K. H. Wang and Y. H. Lin, "Profit analysis of a repairable system with imperfect coverage and service pressure coefficient," May 2011, pp. 127 –131. DOI: 10.1109/CSO.2011.210.

[198] K. H. Wang, C. D. Liou, and Y. H. Lin, "Comparative analysis of the machine repair problem with imperfect coverage and service pressure condition," *Applied Mathematical Modelling*, vol. 37, no. 5, pp. 2870–2880, 2013.

[199] K.-H. Wang and D.-Y. Yang, "Controlling arrivals for a queueing system with an unreliable server: Newton-quasi method," *Applied Mathematics and Computation*, vol. 213, no. 1, pp. 92–101, 2009.

[200] Q. Wang and B. Zhang, "Analysis of a busy period queuing system with balking, reneging and motivating," *Applied Mathematical Modelling*, vol. 64, pp. 480–488, 2018.

[201] Y. Wang, L. Hu, B. Zhao, and R. Tian, "Stochastic modeling and cost-benefit evaluation of consecutive k/n: F repairable retrial systems with two-phase repair and vacation," *Computers & Industrial Engineering*, vol. 175, p. 108 851, 2023.

[202] A. Wills, T. B. Schön, and C. Jidling, "A fast quasi-newton-type method for large-scale stochastic optimisation," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1249–1254, 2020.

[203] A. G. Wills and T. B. Schön, "Stochastic quasi-newton with line-search regularisation," *Automatica*, vol. 127, p. 109 503, 2021.

[204] C.-H. Wu and D.-Y. Yang, "Bi-objective optimization of a queueing model with two-phase heterogeneous service," *Computers & Operations Research*, vol. 130, p. 105 230, 2021.

[205] C.-H. Wu and D.-Y. Yang, "Control charts for the expected system size of markovian queues under $F-$policy," *Quality Technology & Quantitative Management*, vol. 18, no. 5, pp. 576–596, 2021.

[206] C.-H. Wu, D.-Y. Yang, and C.-R. Yong, "Performance evaluation and bi-objective optimization for F-policy queue with alternating service rates," *Journal of Industrial and Management Optimization*, 2022.

[207] C.-H. Wu, D.-Y. Yang, and C.-R. Yong, "Performance evaluation and bi-objective optimization for F-policy queue with alternating service rates," *Journal of Industrial and Management Optimization*, vol. 19, no. 5, pp. 3819–3839, 2023.

[208] S. H. Xu and Y. Q. Zhao, "Dynamic routing and jockeying controls in a two-station queueing system," *Advances in Applied Probability*, vol. 28, no. 4, pp. 1201–1226, 1996.

[209] D.-Y. Yang, P.-K. Chang, and Y.-C. Cho, "Optimal control of arrivals in a g/g/c/k queue with general startup times via simulation," *International Journal of Management Science and Engineering Management*, vol. 16, no. 1, pp. 27–33, 2021.

[210] D.-Y. Yang and Y.-H. Chen, "Computation and optimization of a working breakdown queue with second optional service," *Journal of Industrial and Production Engineering*, vol. 35, no. 3, pp. 181–188, 2018.

[211] D. Y. Yang and Y. C. Chiang, "An evolutionary algorithm for optimizing the machine repair problem under a threshold recovery policy," *Journal of the Chinese Institute of Engineers*, vol. 37, no. 2, pp. 224–231, 2014.

[212] D. Y. Yang, Y. C. Chiang, and C. S. Tsou, "Cost analysis of a finite capacity queue with server breakdowns and threshold-based recovery policy," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 174–179, 2013.

[213] D.-Y. Yang, J.-C. Ke, and C.-H. Wu, "Randomized control of arrivals in a finite-buffer gi/m/1 system with starting failures," *RAIRO-Operations Research*, vol. 54, no. 2, pp. 351–367, 2020.

[214] D.-Y. Yang, K.-H. Wang, and Y.-T. Kuo, "Economic application in a finite capacity multi-channel queue with second optional channel," *Applied Mathematics and Computation*, vol. 217, no. 18, pp. 7412–7419, 2011.

[215] D.-Y. Yang, K.-H. Wang, and C.-H. Wu, "Optimization and sensitivity analysis of controlling arrivals in the queueing system with single working vacation," *Journal of Computational and Applied Mathematics*, vol. 234, no. 2, pp. 545–556, 2010.

[216] D.-Y. Yang and C.-H. Wu, "Cost-minimization analysis of a working vacation queue with n-policy and server breakdowns," *Computers & Industrial Engineering*, vol. 82, pp. 151–158, 2015.

[217] D.-Y. Yang and N.-C. Yang, "Performance and cost analysis of a finite capacity queue with two heterogeneous servers under $F-$policy," *International Journal of Services Operations and Informatics*, vol. 9, no. 2, pp. 101–115, 2018.

[218] X. S. Yang, *Nature-Inspired Optimization Algorithms*. Academic Press, 2020.

[219] X.-S. Yang and S. Deb, "Cuckoo search via lévy flights," in *2009 World congress on nature & biologically inspired computing (NaBIC)*, Ieee, 2009, pp. 210–214.

[220] D. Ye, Q. He, Y. Wang, and Y. Yang, "Detection of transmissible service failure in distributed service-based systems," *Journal of Parallel and Distributed Computing*, vol. 119, pp. 36–49, 2018.

[221] U. Yechiali, "On optimal balking rules and toll charges in the $GI/M/1$ queuing process," *Operations Research*, vol. 19, no. 2, pp. 349–370, 1971.

[222] C. Yeh, Y.-T. Lee, C.-J. Chang, and F.-M. Chang, "Analysis of a two-phase queue system with $< p,F >$ $-$policy," *Quality Technology & Quantitative Management*, vol. 14, no. 2, pp. 178–194, 2017.

[223] T.-C. Yen, C.-H. Wu, K.-H. Wang, and W.-P. Lai, "Optimisation analysis of the f-policy retrial machine repair problem with working breakdowns," *International Journal of Industrial and Systems Engineering*, vol. 40, no. 2, pp. 200–227, 2022.

[224] N. Yiming and B.-Z. Guo, "Asymptotic behavior of a retrial queueing system with server breakdowns," *Journal of Mathematical Analysis and Applications*, vol. 520, no. 1, p. 126 867, 2023.

[225] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, and E. M. Petriu, "Policy iteration reinforcement learning-based control using a grey wolf optimizer algorithm," *Information Sciences*, vol. 585, pp. 162–175, 2022.

[226] H. Zhang and Q. Ni, "A new regularized quasi-newton algorithm for unconstrained optimization," *Applied Mathematics and Computation*, vol. 259, pp. 460–469, 2015.

[227] X. Zhang, L. Chen, G. Sheng, X. Lu, and X. Ming, "An innovation service system and personalized recommendation for customer-product interaction life cycle in smart product service system," *Journal of Cleaner Production*, vol. 398, p. 136 470, 2023.

[228] F.-J. Zhao, X. Du, Y.-H. Ma, X.-M. Sun, and K. Wang, "Optimization and control for variable cycle engine based on grey wolf algorithm," *IFAC-PapersOnLine*, vol. 54, no. 10, pp. 465–470, 2021.

[229] Y. Zhao and W. Grassmann, "The shortest queue model with jockeying," *Naval Research Logistics (NRL)*, vol. 37, no. 5, pp. 773–787, 1990.

[230] M. Zhou, L. Liu, X. Chai, and Z. Wang, "Equilibrium strategies in a constant retrial queue with setup time and the *N*-policy," *Communications in Statistics-Theory and Methods*, vol. 49, no. 7, pp. 1695–1711, 2020.

[231] W. Zhou, "A modified bfgs type quasi-newton method with line search for symmetric nonlinear equations problems," *Journal of Computational and Applied Mathematics*, vol. 367, p. 112 454, 2020.

[232] E. Zohar, A. Mandelbaum, and N. Shimkin, "Adaptive behavior of impatient customers in tele-queues: Theory and empirical support," *Management Science*, vol. 48, no. 4, pp. 566–583, 2002.

[233] F. Zou, L. Wang, X. Hei, and D. Chen, "Teaching–learning-based optimization with learning experience of other learners and its application," *Applied Soft Computing*, vol. 37, pp. 725–736, 2015.

List of Publications

The following works included in this thesis as a chapter have been published/under revision/communicated in the following journals

1. Kumar, A., **Kaswan, S.**, Devanda, M., and Shekhar, C. (2023): "Transient Analysis of Queueing-Based Congestion with Differentiated Vacations and Customer's Impatience Attributes", *Arabian Journal for Science and Engineering* (SCIE), (https://doi.org/10.1007/s13369-023-08020-3).

2. Devanda, M., **Kaswan, S.**, and Shekhar, C.: "Quasi and Metaheuristic Optimization Approach for Service System with Strategic Policy and Unreliable Service", *Journal of Ambient Intelligence and Humanized Computing*, (**Revision Submitted**).

3. **Kaswan, S.**, Devanda, M., and Shekhar, C.: "Economic analysis of a service system with unreliable service of two types of servers", (**Under Review**).

4. **Kaswan, S.**, Devanda, M., and Shekhar, C.: "Admission Control Policy on Online and Impatience Attributes of Offline Customers in Multi-phase Queueing Systems," (**Under Review**).

5. **Kaswan, S.**, Devanda, M., and Shekhar, C.: "Cost Analysis of Customer's Impatience Attributes in the Service System", (**Communicated**).

6. Varshney, S., **Kaswan, S.**, Devanda, M., and Shekhar, C.: "Finite Capacity Service System with Partial Server Breakdown and Recovery Policy: An Economic Perspective", (**Communicated**).

7. **Kaswan, S.**, Devanda, M., and Shekhar, C.: "Cost optimization of a retrial queueing system with an unreliable server incorporating an orbital search mechanism, multiple vacation policies, and the balking phenomenon", (**Under preparation**).

In addition, some research works credit to my profile have been published/communicated in the following journals:

1. Devanda, M., Shekhar, C., and **Kaswan, S.** (2023): "Fuzzifed imperfect repair redundant machine repair problem", *International Journal of System Assurance Engineering and Management*, pp. 1-20.
(https://doi.org/10.1007/s13198-023-01922-3).

2. Devanda, M., **Kaswan, S.**, and Shekhar, C.: "Quasi and Metaheuristic Optimization Approach for Service System with Strategic Policy and Unreliable Service", *Quality and Reliability Engineering International* (SCIE), (https://doi.10.1002/qre.3421).

3. Devanda, M., **Kaswan, S.**, and Shekhar, C.: "The state-of-the-art methodologies for reliability analysis of imperfect repair and threshold-based measures," (**Communicated**).

List of Attended Conferences/Workshops

1. Presented paper entitled "Economic Analysis of a Retrial Queueing System with Balking, Orbit Search, Multiple Vacation, and Unreliable Server" in *International Conference on Mathematical and Statistical Sciences (ICMSS-1)* organized by the Department of Mathematics, Statistics and Actuarial Science, Namibia University of Science and Technology on July 03-04, 2023.

2. Presented paper entitled "Controllable Online Joining and Balking Strategies of Customers in a Two-stage Service Systems" in *International Conference on Dynamical Systems, Control and their applications (ICDSCA-2022)* organized by the Department of Mathematics, IIT Roorkee on July 01-03, 2022.

3. Presented paper entitled "Economic analysis of service system with junior-senior servers and unreliable service" in *International Conference on Advances in Mechanics, Modelling, Computing and Statistics (ICAMMCS-2022)* organized by the Department of Mathematics, BITS Pilani, Pilani Campus on March 19-21, 2022.

4. Presented paper entitled "Cost analysis of $M/M/R+1/K$ service system with junior-senior servers and unreliable service" in *International Conference on Advance Trends in Computational Mathematics, Statistics and Operations Research (ICCMSO-2022)* organized by the Department of Applied Sciences, The Northcap University, Gurugram, Haryana, India on April 2-3, 2022.

5. Attended "Workshop on Data Analysis using Statistical Packages", organized by the Department of Mathematics, Saranathan College of Engineering, Chennai, during June 29-30, 2020.

6. Attended "National Workshop on Queuing with retention of impatient customers", organized by the Department of Science and Humanities Engineering, Hindustan College of Engineering and Technology, Chennai, during 12th June, 2020.

Brief Biography of the Candidate

Ms. Suman Kaswan received her Bachelor of Science, B.Sc. (Hons.) in Mathematics, from the University of Delhi in 2017. In 2019, she received her M.Sc. in Mathematics from the Indian Institute of Technology, Patna. Currently, she is working towards a Ph.D. degree from the Birla Institute of Technology and Science, Pilani. Her research interests lie primarily in the areas of development of queueing models incorporating several features like retrials, vacations, impatience, service regimes, arrival control policies, etc.; implementation of optimization techniques on the cost minimization problem of the system; a matrix-analytic method for solving stationary distributions; and probability generating functions techniques. She has published several research article in peer-reviewed journal to her credit.

Brief Biography of the Supervisor

Prof. Chandra Shekhar, the faculty, and Ex. Head in Department of Mathematics, BITS Pilani, India, is actively involved in teaching and research in the area of Engineering Mathematics, Operations Research, Optimization, Probability & Statistics, Fuzzy Logic, and Differential Equations and having much research interest in the area of Queueing theory, Stochastic processes, Computer and communication system, Reliability and maintainability, Manufacturing and machine repair problem, Reliability engineering, Fuzzy set and fuzzy logic, Inventory theory, Statistical inference and analysis, Evolutionary and nature-inspired optimization techniques, etc.

Besides attending, presenting scientific papers, and delivering invited talk in conferences, he is organizing committee member of many conferences and workshops. He has a number of published articles in journals, proceedings, and edited books of repute along with association as a member of the editorial board and reviewer of reputed journals. He is also a member of many scholastic societies.

In addition to a member of the Board of Studies and Doctoral Research Committee of many universities, he has delivered invited talk in universities and has visited many research organizations, government organizations, industry. He has association with CBSE-AIPMT, National service scheme (NSS), BITSAT, UKPSC, and many more.

Two Ph.D. students have graduated from his research group in 2020, while four other Ph.D. scholars are working on various problems in the areas of queueing theory, reliability and maintainability, and inventory modeling.