Prediction of waiting time for entities in independent and merged queues using Markov chain and machine learning techniques

A Thesis Submitted to the

College of Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Mathematics and Statistics

University of Saskatchewan

Saskatoon

By

Mehrdad Dadgar

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College where thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Mathematics and Statistics
176 Mac Lean Building
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

Abstract

Serving customers for any organization is a factor in making profit and extending business. Waiting time has become one of the most critical features in measuring the performance of systems; managers try to predict and minimize this factor. On the other hand, any system can have different customers. Thus, defining priorities can increase profitability and customer loyalty. Two models can emerge in these situations: the independent queue lines and the merge queue line. For the first model, a non-linear programming model is presented by combining resource planning concepts and queueing theory. The non-linear model helps with minimizing the weighted sum of average waiting times by assigning servers to different priority groups. Also, we address a "Memoryless/Memoryless/3 Servers" or $M/M/3$ queue system with two priorities. In other words, the system contains three servers and the distributions of arrival and service times are exponential. A dimension reduction method is used to convert the transition matrix from two-infinite dimensional to a one-infinite dimensional matrix with the help of Phase-type distribution. After solving the Quasi-birth-death process, to compare the results, a simulation method is applied; the average gap between the dimentionality reduction method and simulation results is about 7%.

Moreover, two unsupervised learning methods, K-Means and Gaussian mixture model, are used principal component analysis to cluster customers in an example on the banking industry. In the last step of research, we investigate the $M/M/3$ with three priorities. The high complexity of the queuing systems with more than two priority groups has resulted in having few methods in the literature. Therefore, a novel simulation-driven machine learning algorithm is developed to solve the problem. Some supervised machine learning algorithms are applied to predict the dynamic waiting time for customers. Among them, a random forest regression with mean absolute error of 6.95 minutes has the best performance. This novel method can analyze any merged queueing system with an arbitrary number of priorities and servers.

Acknowledgements

I would like to express my gratitude and appreciation for my supervisor Dr. Hamed Samarghandi whose guidance, support and encouragement have been invaluable throughout this study. The door of his office was always open whenever I ran into a trouble spot or had a question about my research or writing. Undoubtedly, my life has been affected so much by him and I really appreciate it.

Dedication

I dedicate my dissertation work to my family and especially my father that I lost him during my education.

Contents

List of Tables

List of Figures

List of Abbreviations

CMCM	Continuous Markov Chain Modeling
DR	Dimensionality Reduction
ED	Emergency Department
EDA	Exploratory Data Analysis
EM	Expectation Maximization
ESI	Emergency Severity Index
FCFS	First Come First Served
GMM	Gaussian Mixture Model
Lasso	Least absolute shrinkage and selection operator
LCFS	Last Come First Served
MAM	Matrix-Analytic Methods
MAP	Markovian Arrival Process
MAE	Mean Absolute Error
MGM	Matrix-Geometric Method
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MMA	Moment Matching Algorithm
MSE	Mean Square Error
NLP	Non-Linear Programming
OLS	Ordinary Least Square
PCA	Principal Component Analysis
PH	PHase-type
QBD	Quasi-Birth-Death
RF	Random Forest
SVR	Support Vector Regression
WSAWT	Weighted Sum of Average Waiting Times

1. Introduction

Nowadays, with the growing technology and population, managing an organization has become increasingly intriguing. One of the challenging tasks is designing a service system, which consists of decisions such as the number of servers, the service policy and much more. The resources available to the organizations, which are almost always limited, include machines, material, equipment, and human resources, among others; waiting can be influenced by factors such as the capacity of resources, layout, service, and processing policy decisions [7]. On the other hand, the consequences of the influx of customers may result in problems like overcrowding; wait time or waiting time can easily develop into a painful experience, which can impose high psychological, social, financial, or physical costs on the entire service system. Consequently, wait time and its related measures have become essential factors in analyzing the performance of service systems.

A considerable number of studies probe the effects of waiting times, especially behavioral consequences. A striking feature of waiting time is that it can affect both customers and owners of systems. Therefore, long waiting times and consequently, long queues may impact various aspects such as customer experience and loyalty [8, 9], and customer sanctification [10, 11, 12].

1.1 Motivation

One of the challenging decisions that a manager should make is defining some criteria to categorize the customers. Customers categorized in some classes may experience less waiting times. Such clustering of customers will be called prioritization. There are three primary motivations for prioritization. First, customers may have different willingness to pay (or valuations) for the same product. The second reason is that in many situations, a particular product or service is more profitable than the others; so, it is a wise decision to consider the consumer of these products a priority. The third motivation is that having different service levels can impact the profit of an organization in the long term [13]. As Afeche [14] told in his research, the customers receiving excellent service

are more likely to become loyal ones.

In practical situations, there are two different cases in a queueing system regarding prioritization. In the first case, there is some knowledge about the customers that can be employed for classification. For example, in the emergency department, the severity of the illness is a useful factor to separate patients. On the other hand, there are many situations in which a priory knowledge about customers does not exist; hence, separating the customers to similar groups may not be possible. For example, in banks, insurance companies and grocery stores there are several important attributes; considering one or a subset of them as the exclusive separating factor may cause losing information and improper customer segmentation.

1.2 Research Framework

To serve the customers, the system needs resources or "servers" are required. These servers can be machines, rooms, material, doctors, tellers, humans, or even a group consisting of a combination or all of them. In this research, two scenarios are investigated:

1. **Model A**: each group of customers should receive their service on a specific collection of servers.

2. **Model B**: all customers can be served by any of the servers.

1.2.1 Model A- Independent Queueing System

In this model, it is assumed that each group of customers imposes a different penalty for waiting in line. These systems can be hospitals, libraries, amusement parks, call centers, central processing units, banks, restaurants, and other types of service or industrial centers. To deal with this problem, a non-linear programming model to minimize the weighted sum of average waiting times (WSAWT) is proposed.

It is assumed that G groups of customers with different priorities enter the system, in which S servers exist. In this system, the mean arrival time between customers of each group and the mean

service time of servers are exponential. In the queueing theory literature, this system is described as an $M/M/C/\infty$ queueing model.

$M/M/C/\infty$ is a classical model in queueing theory, according to which the arrival and service times have exponential distributions. In this notation, C indicates the number of service providers available, and ∞ means the system can accept infinite number of customers. In **Model A**, it is assumed that each group of customers has its dedicated servers (figure 1.1). In this figure, C_j ($j = 1, ..., G$) shows the number of servers; λ_j and μ_j ($j = 1, ..., G$) indicate the arrival and service rate for each group. Numerous applications of this model will be discussed in the next chapters.

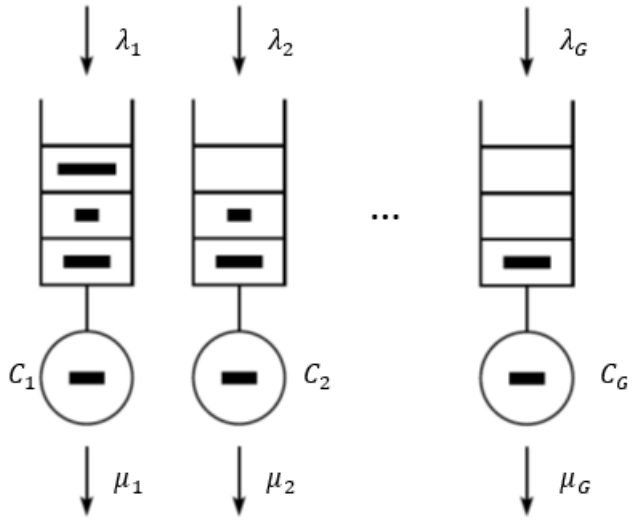

Figure 1.1: Model A: G priority queue system with independent servers
C_j: number of servers; λ_j: arrival rate; μ_j: service rate.

1.2.2 Model B- Merged Queueing System

For the second scenario, there are no dedicated servers for the groups, and the system has shared servers among groups. Therefore, a customer, irrespective of its classification, can receive service from any of the servers called merged queue line. It is assumed that in the merged queues, the customers from the 1^{st} group have the highest priority in the system, customers from the 2^{nd} group have the second highest priority, and so on. So as long as a customer with higher priority is present

in the system, the lower priorities will not receive service.

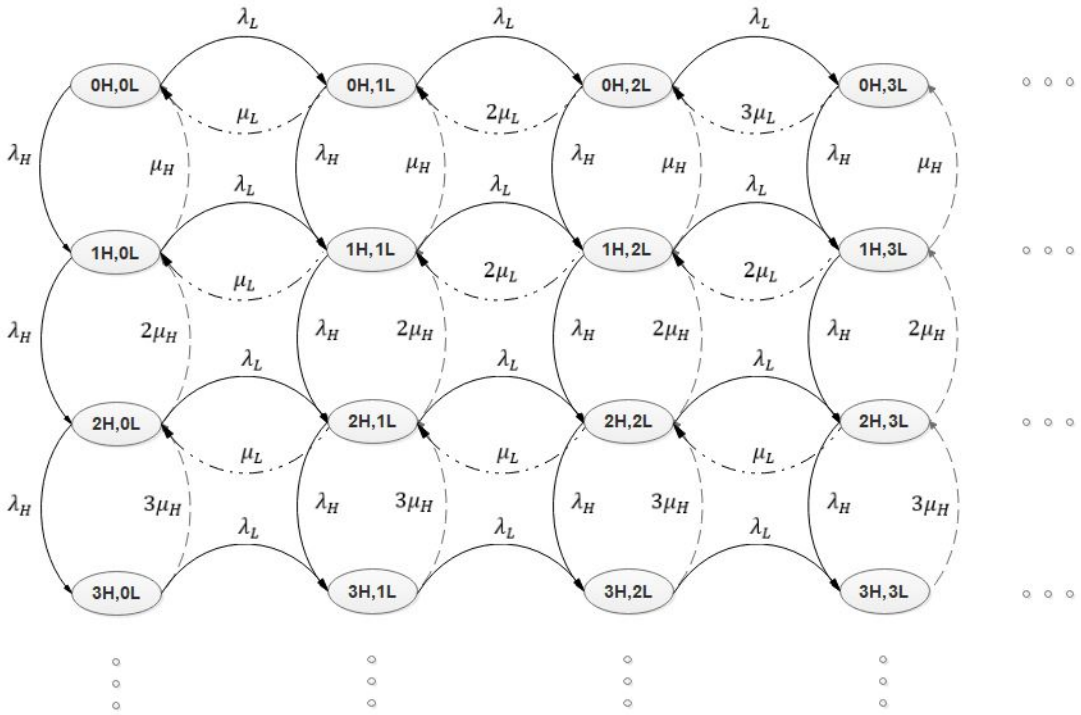

Figure 1.2: Model B: two priority queue system with common servers

In this thesis, **Model B** is considered as a preemptive queueing system and is modeled by the use of Markov chain modeling with infinite states in two dimensions.

Figure 1.2 illustrates the above explanations. In this figure an $M/M/3/\infty$ queue with two priorities (H,L) and three servers is shown. Parameters λ_H and λ_L show the arrival rates for each group. μ_H and μ_L demonstrate the service rates for high and low-priority groups. This chain grows infinitely in two directions. Unfortunately, the transition matrix for this Markov chain cannot be modeled as a Quasi-Birth-Death (QBD) process. Thus the Dimensionality Reduction (DR) approach of Osogami [15], is used to reduce the dimension of the Markov chain. The DR helps in modeling a 2D infinite Markov chain as a 1D infinite Markov chain. After DR, the steady-states for **Model B** are computed with the use of matrix-analytic methods.

Few researchers have addressed the problem described in **Model B**. To the best of writer's

knowledge, **Model B** with more than two priorities has not been studied before. This research presents a new approach called simulation-driven machine learning to solve the problem when more than two priorities exist.

1.3 Thesis Organization

This thesis is divided into seven chapters.

- Chapter 2 gives a brief overview of queueing theory and prioritization. Also, it contains the efforts to implement machine learning techniques and customer segmentation.
- A new methodology is described in chapter 3 to solve the resource planning problem using classical queueing theory approaches. In this chapter, **Model A** is explained in detail, and a novel non-linear programming model is presented.
- Chapter 4 investigates **Model B**. The importance of prioritization is described in this section. Also, a discussion of QBDs and two algorithms for solving them are presented. This section uses the Osogami's technique for dimension reduction, which is an approximation method.
- Chapter 5 provides an introduction to machine learning techniques. A dataset has been used to cluster the customers to similar groups. Customer clustering is performed using two famous unsupervised learning algorithms.
- In chapter 6, a simulation-driven machine learning model is introduced. This model creates proper features and predicts dynamic waiting times. After performing an exploratory data analysis, some machine learning techniques are used to predict dynamic waiting times for an $M/M/n$ system with multiple priorities.
- Finally, chapter 7 summarizes the major findings of the research, and lists potential future research directions.

2. Literature Review

Since this thesis employs various concepts in addition to the queueing theory, the literature review chapter is organized to some sub-sections, each dedicated to a specific subject.

2.1 Waiting Times and Markov Chain

Wait time is known as an important factor in analyzing a queueing system. Nowadays, this factor attracts more researchers to ponder about the relationship between this factor and the other essential criteria. For example, in the health care system [16], waiting has turned into a hot topic for researchers. Other examples will follow.

Robat et al. [17] study the relation between waiting time and tumor growth for cancer patients. Sometimes the researchers wonder to find a threshold for waiting times that is acceptable for patients. Kumazu et al. [18] analyzed the relationship between surgery, survival of patients and waiting times. In this research, the authors stated that for waiting times shorter than 100 days, gastric cancer patients' survival rate does not deteriorate. To understand the effect of waiting times on moral tendencies, Efrat et al. [19] studied the effect of waiting times in health care on the patients' sense of procedural justice and aggressive tendencies.

Zeinal et al. [20] concentrated on congestion and long wait times in Emergency Departments (ED). They considered a resource planning problem when the objective functions indicate the sum of the total average waiting time of patients. The authors were unable to show the objective function analytically; they used a simulation model to evaluate the performance of the system and compute the number of resources. They introduced their model as a tool to manage and improve the flow of patients and relieve congestion by changing the number of resources, including nurses, residents, doctors or even beds.

The importance of stuying waiting time is not limited to healthcare. Kim et al. [21] considered a multi-server retrial queueing system with N independent and identical servers, Batch Markovian

arrival, and Phase-type service distribution. They studied the behavior of the system as a level-dependent multi-dimensional Markov chain. According to this study, when a batch of primary customers arrives and several servers are idle, the primary customers occupy the corresponding number of servers. Whenever the number of empty servers is not enough, the batch will go to the so-called orbit. These customers are known as repeated customers. Repeated customers will try entering the system and getting their services later. The contribution of authors helps with calculating the performance measures even for a large number of servers.

Rejita and Jose [22] evaluated the Markov chain process for an inventory problem. They analyzed a special kind of inventory problem with a real service time and retrial customers. These customers arrived based on a Markovian Arrival Process (MAP), while service time has a Ph-type (PH) distribution. They used the matrix-analytic method to compute the state probabilities.

Farkas and Telek [23] investigated an electric car charging station and modeled this application with the use of queueing theory, with the goal of determining the number of chargers to meet the defined service level. Cars enter the system according to a Markovian arrival process in which cars are controlled by a background continuous-time Markov chain. They used moment matching algorithm to fit the observed arrival data and service time with a continuous-time MAP of order two. MAP is a general form of mathematical model for arrival time of customers to the system; in its simplest shape MAP can have Poisson distribution, but in [23] it consisted of two Poisson distributions.

Ye and Liu [24] analyzed an $MAP/M/1$ queue system with break down. They used the matrix-geometric method to compute the steady-state performance criteria, such as the number of customers in the system. In their problem, the server failure occurs according to a Poisson process at a constant rate. Once the server breaks down, it will be sent for repair with an exponential distribution. During the repair time, a standby server may continue the service instead of stopping the service but with a lower service rate. They modeled this system as a Markov chain process and used the Quasi-Birth-Death (QBD) method to solve the stationary probabilities.

Phung and Kawanishi [25] used a three-dimensional continuous-time Markov chain to obtain

the stability condition of their problem. They analyzed a multi-server retrial queue with setup time, which can be applied in data centers. Whenever a customer arrives at the system, if there is an off-server available, the customer will get its service; otherwise, the customer will leave and re-enter the system at a random time. Additionally, returning to operational level for the idle servers requires a setup time. Numerical examples show this policy is effective for power saving and waiting time reduction.

2.2 Prioritization

Not all the customers in a system have the same priority for the firm. Therefore, the service and wait times are design in accordance to their priorities. Many researchers have focused on determining the waiting time for different priorities. For example, Mazzini et al. [26] investigated the behavior of a two-priority queue, when the entities arrive with a Bernoulli distribution; they calculated the length of low- and high-priority queues as a closed-form formula.

Walraevens et al. [27] investigated the asymptotic queue length of the $M/G/1$ retrial queue in which priority customers wait in line, while non-priority customers join an orbit and retry later. This assumptions can be seen in restaurants and call centers. The studied model was an $M/G/1$ queue, where two groups of customers enter the system with Poisson rate and service times follow a general distribution.

Wong et al. [13] considered an $M/M/C$ queue with two priority jobs. They described the model as a two infinite-dimensional Markov chains. They employed a technique based on generating functions to reduce the dimension. Also, They proposed an exact algorithm to calculate the moments of the number of low-priority jobs. In a two priority environment, analyzing the number of low-priority jobs is difficult. In the new dimension, with the help of proposed technique, they computed different moments of the number of "Class 2" jobs in the system.

Sarhangian et al. [28] used a queueing model to solve a perishable inventory management problem. They modeled the Red Blood Cells (RBC) inventory as an $M/M/1+D$ system in which RBC

units arrive at a hospital with a Poisson rate; then, RBC units are independently requested based on another Poisson rate. Hence, the consumption process of RBCs can be modeled as a queue. The crucial assumption is that RBCs have a deterministic threshold until the end of the service, and if their sojourn time exceeds the threshold, they have to leave the system. They assumed units arrive and are consumed one by one, despite the fact that, in reality, hospitals may receive blood shipments consisting of multiple units, and one patient may need to receive more than one unit.

Aniyeri and Ratnam [29] studied a multi-server queueing model with two servers in an international airport. After entering the terminal, the passengers are divided into two groups. The first group is the batch queue passengers. They join the queue when the servers are doing batch service, and other passengers wait in the individual queue ($M/M/2$). The problem was modeled as a QBD process; the matrix-geometric method is employed to analyze the system. The authors reached a formula for probability distributions of the number of passengers in the queues and some other related features. Since the QBD process has a particular tri-diagonal structure in its transition matrix, it can be adequately solved through the matrix-geometric method. The matrix-geometric method can be categorized as a matrix-analytic approach.

Phase-type or Ph-type (PH) distribution is a probability distribution constructed by a convolution or mixture of exponential distributions, that has attracted many scholars in the field of queueing theory. Ph-type (PH) distributions have been used in a massive spectrum of stochastic modelings such as telecommunications, finance, bio-statistics, queueing theory, drug kinetics, and survival analysis. Fackrell [30] focused on Phase-type distributions and their applications in the healthcare industry. They found out that Phase-type distributions can be used to fit any length of stay or inter-arrival data because of its flexibility.

Call centers are considered as an integral part of a large number of businesses. Since the staff members in the call center are directly in contact with customers, they have a crucial impact on customer satisfaction and retention. Afche[14] focused on the profit maximization problem in such a call center. The author considered two types of impatient customers in the system. New customers are assumed to be first-time callers; base customers repeatedly interact with the call center. Finally,

they studied the model in a steady-state, where every server has a cost per time unit and a policy prioritizes new or base customer calls.

Valakevicius [31] introduced a method for evaluating queue models and their approximations in which a numerical queueing model with priorities is used to explore the behavior of exponential Phase-type approximation of service-time distribution. They analyzed a queueing system for quality control lines; and used a moment matching method to approximate the model and calculate the performance criteria.

Some researchers have addressed the priority concept in systems with one server. Takagi et al. [32] analyzed the distribution of sojourn times for a single-server queue with priorities, such as $M/G/1$. Miller [33] presented an efficient algorithm to derive the steady-state probability distribution of an $M/M/1$ queue with priorities using the matrix-analytic method.

Brandi and Brandt [34] studied a queueing system with s servers and protected and unprotected queues, in which the arrival and service rate depend on the number of customers in the system. They also considered the case of serving impatient customers; such customers will leave the queue once they wait longer than a threshold.

In some studies, scholars have tried to model special phenomena such as failure of servers to investigate its effect on the performance of system. Some preliminary work was carried out in the early 1990s. As an example, Twosley and Tripathi [35] investigated a single server with two classes of customers. This server is prone to failures, and at the time of failure, all customers are flushed out of the system. With consideration of general repair times and exponential time between failures, they computed the performance metrics for every class of customers.

Gail and co-authors [36] considered a queueing system with multiple servers and two groups of customers being served based on a preemptive resume priority rule, in serving a member of one group stops when a customer from the other group joins the line. The service for the interrupted customer resumes once the system does not contain any higher priority customer. They computed the equilibrium probabilities of the number of customers in the system using generating function.

Alfa [37] focused on a particular type of queueing system in which the arrival pattern has a

MAP queue model, and the servicing distribution for customers has a Ph-type (PH) distribution. This study considers two classes of priority customers. The author used a matrix-geometric method to analyze the performance of the system.

Brouns and Wal [38] considered a two-class queueing system with a preemptive priority which requires two decisions: whether the new jobs should be accepted in the system; and whether a job should be removed from the system. Sleptchenko and co-authors [39] analyzed an $M/M/1$ queueing system with multiple customer classes in which the service time has exponential distribution, and there is preemptive priority between classes. They used this method in the logistics industry. matrix-analytic techniques and probabilistic arguments were applied to compute the equilibrium joint queue length distribution. Furthermore, Sleptchenko et al. [40] developed an exact solution for the state probabilities of a multi-class queueing system with preemptive priorities.

Sharif et al. [41] developed a queueing system for the healthcare industry. They investigated the Accumulating Priority Queue (APQ), where customer priorities are a function of their waiting times. They presented their results with expressions for the waiting time distributions of an APQ system with multi-server in which arrival pattern and service times have Poisson and exponential distributions, respectively.

2.3 Customer Segmentation

Segmentation is the act of dividing individuals or objects into meaningful sub-groups [42]. Primarily, it intends to reduce the number of entities into a manageable number of clusters that are exclusive with specific characteristics [43]. The importance of segmentation for organizations and their marketing strategies is widely accepted [44, 45]. It enables an organization to prioritize the customers and service them with better performance.

In many organizations, the easiest way to classify customers is using the 80/20 rule to divide customers into two groups. The 80/20 rule means 80% of the profits are made by the top 20% of customers, and 80% of the costs are produced by the top 20% of unprofitable customers [46, 47].

Kim et al. [2] stated that with the evolving marketing paradigm, the relationship with customers gains more importance. Therefore, the organization tries to pursue long-term relationships with profitable customers. A starting point is to manage and understand the real value of customers. Measuring and identifying the value of customers can help build and maintain loyal and valued customer relationships. They proposed a calculation model to compute customer value in a telecommunication company. They divided the customers into two groups: customers in the low-potential loyalty and customers in the high-potential loyalty; see figure 2.1.

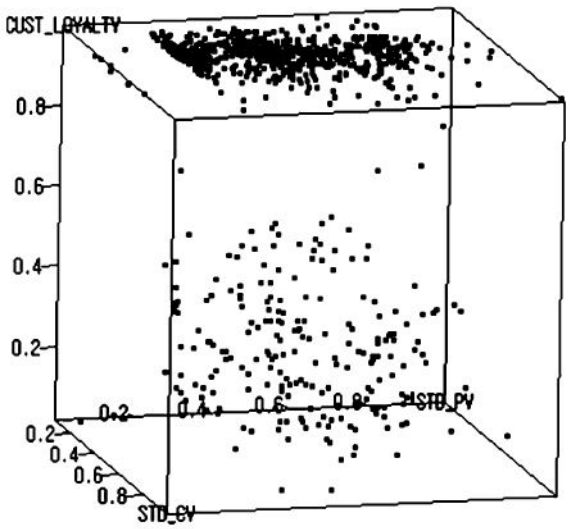

Fig. 6. Result of customer segmentation.

Figure 2.1: Result of customer segmentation [2]

Several approaches for dividing customers into groups exist in the literature. These methods can be categorized into three branches [48]:

1. **A-priori segmentation methods:** segments are defined before the data is collected. Thus, the segments may be identified by the use of customer characteristics or product-specific information.

2. **Post-Hoc methods:** identify the segments empirically with the help of data analysis or by using multivariate data analysis approaches.

3. **Hybrid approaches:** use a combination of the two previous methods [49].

One of the best method for segmentation is using Machine Learning (ML) techniques. ML is attracting widespread interest due to its flexibility and using real data to analyze the problems and find solutions. Since, in the theoretical world, the best model is the model that has more similarities to reality, using data and datasets on top of theoretical methods will convert methods to a qualified approach to solve real problems. Therefore, using machine learning techniques for customer segmentation give the opportunity of using real data besides classical formulas to have meaningful groups among customers. Customer segmentation has been applied to a large number of applications. Some examples will follow.

[50] worked on customer segmentation in the airline industry. Based on their investigation, the simplicity of segmentation logic in the airline industry no longer matches the ever more difficult choices made by customers. Therefore, the airlines relying on flight class as a dividing rule of separating customers will not be able to customize their product and marketing policies. They applied latent class modeling to data from 5800 airline passengers to find some rules for segmentation of airline customers.

Chen et al. [51] performed a case study that demonstrates a technique for making customer-centric business intelligence for online retailers using data mining. To better understand the customers in the online retail industry, they used K-Means and decision tree induction algorithms.

Bi et al. [52] found that businesses pay more attention to the existing customers to avoid customer churn. Customer churn refers to the loss of customers switching from one company to another competitor within a given period [53]. Thus, customer churn management is vital, especially in the service industry. Also, Bi et al. [52] focused on customer churn management with big data analysis. They employed Axiomatic Fuzzy Sets (AFS) with use of K-means to cluster the customers and mitigate the risk of customer churn.

Tsai et al. [54] found that companies use customer segmentation to identify customers with similar characteristics and developed strategies based on these groups. They investigated a case study of an automobile dealer in Taiwan. The authors considered two clustering techniques: K-

means and expectation maximization, and categorized the customers into four groups.

Sekaran et al. [55] implemented machine learning algorithms to predict how much a customer may spend in a mall. To create clusters, they used two clustering approaches: K-means and hierarchical clustering algorithm.

Bayrak [56] reviewed several approaches and findings available in the literature, and classified the terms used to describe analytics. Specifically, he defines what businesses are doing today when they use metrics to measure the performance of their systems. A significant section of their job consists of describing the different types of analytics that can be used for prediction and heavy computations. Some of the terminology of their job follows:

1. **Descriptive analytics:** interpreting the data to understand the changes and make comparisons over a range of time.

2. **Predictive analytics:** predict future events by use of analyzing the historical data.

3. **Prescriptive analytics:** it is a combination of descriptive and Predictive analytics. It employs mathematical and computational models to take advantage of those two other fields.

Based on these definitions, results of this thesis has presented as predictive analytics in chapters 5 and 6.

2.4 Research Gaps

The research gaps that motivated this research are divided into three categories:

1. Many researchers have addressed queueing systems with an $M/M/c$ structure; however, some queueing systems consist of several $M/M/c$ structures within them. In other words, sometimes a queueing system can be divided into some sub-systems, each with $M/M/c$ structure. Such problems may be analyzed by mathematical programming.

2. Sometimes servers can be shared among customers with different priorities; therefore, instead of forming multiple queues of customers, a single waiting line is formed. Due to its complexity, most of the research in the literature has addressed an $M/M/2$ system with two priorities. This research deals with the problem with three servers and two priorities.

3. To the best of the author's knowledge, there are few researches available in the literature that studies queueing systems identified as **Model B** in this thesis with more than three priorities. This thesis presents a new approach to deal with the mentioned problem. The mentioned technique can be applied to any queueing system with any arrival and service time distribution.

3. Systems with Independent Queues

3.1 Introduction

Besides increasing the population and progressing technology throughout the world, organizations tend to improve the quality of their services. Each organization tries to improve its performance and increase its profits. The customers search for places with better quality and shorter service time. As an example, any person may experience the annoyance of having to wait in a long line for service in a bank. Managers want to decrease the waiting times for customers because it costs time, money, and even poor quality of service for a business and may affect the loyalty of customers.

3.2 Characteristics of the Queuing Process

There are six characteristics of a queuing system that play a crucial role in the structure and performance of the systems and servers. In the following they will be elaborated [1]:

3.2.1 Arrival Patterns

In usual queuing situations, the process of arrival in the system is stochastic; therefore, it is necessary to know the distribution that describes the time between customer arrivals. If the arrivals to a queue occur according to a particular frequency distribution, the intervals between the arrivals will also follow a probability distribution. These intervals may be independently or dependently distributed. Also, it is necessary to know the probability distribution describing the number of customers who arrive in the system because, sometimes, they enter the system as individuals or in groups.

For arrival patterns, there is a particular type of customer arrival known as impatient customers. A customer may decide to join the queue and wait in the line, or, on the other hand, if the number of

customers in the queue is too many, they may choose not to enter the queue. Usually the customer is said to have balked if s/he refuses to join the queue. With the second type of impatient customer, the customer may join the line, but after a time decides to leave the queue because s/he has lost his/her patience; in this case, the customer is said to have reneged. The third kind of impatient behavior involves a queue with more than one line of service; in this case, the customer joins one of the lines and may switch from one line to another with the hope of getting service sooner.

Sometimes, a stationary probability distribution exits for the arrivals, meaning that the mean and variance of arrivals do not change over time. In some cases, the system may experience customer arrival patterns which are non-stationary or time-independent.

3.2.2 Service Patterns

Most of the previous discussions concerning arrival patterns are applicable to the service pattern as well. Hence, there is a probability distribution to describe the duration of customers receiving service. In many situations, servers give service to customers one-by-one. On the other hand, in numerous systems, the servers can simultaneously serve several customers; examples include taxis, planes, or industries in which machines serve multiple products at the same time.

Also, the server can work faster as the number of customers served increases because the server can learn to work more quickly. On the contrary, when the number of customers in the system goes up, the server may become less efficient, which increases the service time. The problems in which the service time depends on the number of customers waiting in the line is referred to as state-dependent service. Also, the service pattern can be stationary or non-stationary, or a combination of state-dependent and stationary.

3.2.3 Queue Discipline

Queue discipline refers to how customers are selected for service when they enter the system and wait in the queue. The most common discipline is First Come, First Served (FCFS); it means the first customer entering the system is the first one to receive service. The other type is Last

Come, First Served (LCFS), which is appropriate for inventory management. Sometimes analysts consider a different priority between the customers based on their importance for the system, so this priority assures that the system serves the customers based on a defined order.

For queuing systems with priority, there are two policies. In the first policy, as soon as a high-priority customer enters the line, serving the lower priority customers is preempted until after service to the high-priority entity is finished. In the non-preemptive case, the highest priority goes to the front of the queue and receives service after the completion of the service of the lower priority customer.

3.2.4 System Capacity

In many real situations, the queuing system has a waiting room or space because the queuing space can only accept a limited number of customers. So, when the queue line reaches a certain length, no more customers are allowed to enter the system. On the other hand, the system may not have any thresholds for admitting customers; capacity of such systems are considered to be infinite.

3.2.5 Number of Service Channels

The number of service channels refers to the number of parallel service stations that can serve the customers simultaneously. It is generally assumed that the service mechanisms of parallel channels operate independently of each other.

3.2.6 Service Stages

A queuing system may have a single stage for services such as in airports and amusement parks, or it may have several stages to deliver a complete service, such as a production line in a factory or medical operations. In some multistage queuing systems, recycling or feedback may occur. Recycling or reworking is a common phenomenon in the manufacturing process, where quality centers define the products that should receive the rework.

3.2.7 Queuing Systems

Queuing systems generally mean line systems used by users or customers receiving services. In a queuing system, one or more service stations or servers provide service to the customers. A schema of queuing system is presented in 3.1. There is a general assumption that a server cannot serve two or more customers at the same time. If the server is busy, customers have to wait in a queue to receive service. When the server becomes free, a customer moves from the queue to the server according to a defined discipline.

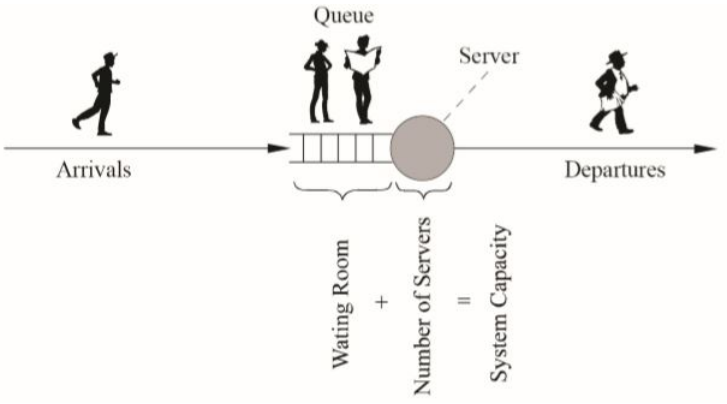

Figure 3.1: Scheme of queuing system [3]

The customer arrival pattern is a stochastic process defined by an adequate probability distribution. In many cases, the arrival process can be modeled by a Poisson distribution. The parameter that characterizes the arrival process is arrival rate, $\lambda = \frac{1}{E(t_{arrival})}$, which represents the average number of arrivals per unit of time. Similarly, the service time is defined by a statistical distribution. The service rate, $\mu = \frac{1}{E(t_{Service})}$, represents the average number of customers being served per unit of time.

Besides the above parameters, there is another important parameter in queuing systems, which is called "*loading factor of the service server*" or "*utilization factor*," ρ. ρ is defined as the portion of time the service station is busy and cannot serve other customers. Since the system cannot perform more work than its capacity allows for, the upper bound of the utilization factor is restricted by one

[3]:

$$\rho = min\{\frac{\lambda}{m\mu}, 1\}, \tag{3.1}$$

where m is the number of service providers. When the arrival rate, λ, is greater than the service rate, μ, the system is completely busy.

Definition 1. Unstable systems

$\rho > 1$ means that more customers arrive to the system than exit, i.e., the length of queue tends to increase to infinity. Such systems are called **non-stationery** or **unstable systems**. The system is stable only if the arrival rate is less than the service rate ($\rho < 1$). Usually researchers try to investigate and analyze the performance of stable systems.

3.3 Queuing Networks

A queuing network is a directed graph that represents a queue. The nodes in the queuing networks are often servers or sometimes states, and the branches represent links among particular servers, as shown in figure 3.2. The pathways of customers between servers are designated by the probabilities or rates of movement among servers.

A queue, in the sense used here, can be described as a group of entities, such as machines waiting for repair [57, 58, 59], customers waiting for service [60], vehicles waiting at petrol pumps [61], products in a production line, kids in amusement parks [62], passengers in an airport [63], or even a task in Central Processing Units (CPUs). There exists some server or service facility, such as a cashier, repair person, or green traffic light. In queuing theory, the term queue is normally reserved for the number of elements waiting, excluding the elements being served. The word line, on the other hand, is used to refer to all elements in the system, including the ones currently being served.

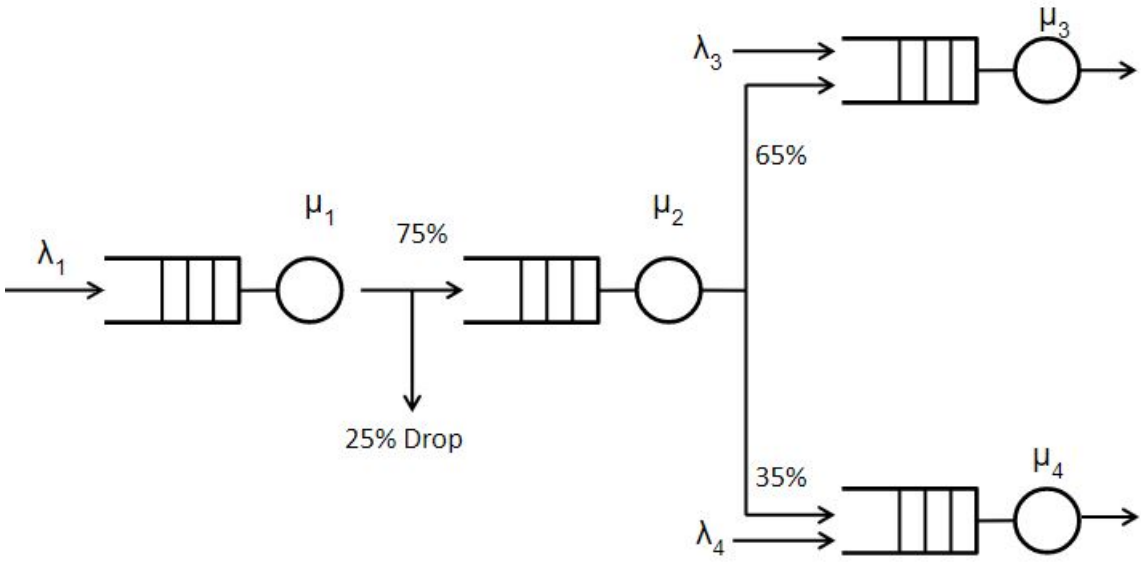

Figure 3.2: A queuing network diagram

3.4 Birth-Death Process

A birth-death (birth and death) process is a particular type of Markov process. As the name implies, birth-death processes were initially used to describe populations increased by births and decreased by deaths. This name is taken from the science of biology since biologists, in particular, study the development of populations of animals, plants and other organisms, using the birth-death processes. These types of operations can be seen in many situations. The birth-death process can be defined as *a group of entities whose numbers increase by births or arrivals, and decrease by deaths or departures* [64]. Figure 3.3 shows a state diagram for a birth-death process.

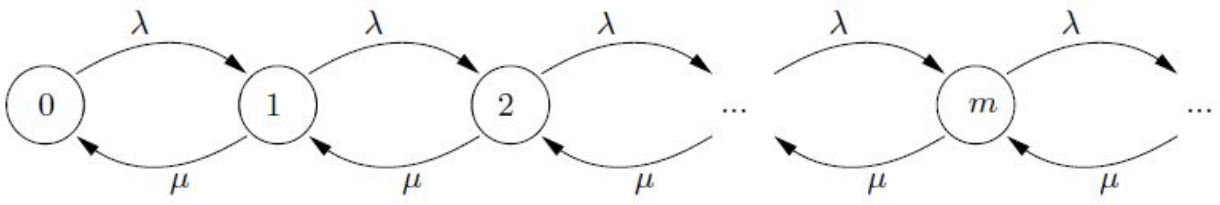

Figure 3.3: State diagram of a birth-death queuing system

Assume a birth-death process with one state variable, S, representing the number of elements in the population or in the line. Both birth and death rates depend on S. If $S = i$, the birth rate is λ_i and the death rate is μ_i. Since a birth increases the population S by 1, the rate at which S increases from i to $i + 1$ is λ_i. Also, when the state decreases from i to $i - 1$, the rate of the changing state is μ_i. Because no other transition between distinct states exists, the transition matrix of a birth-death process assumes the form shown in figure 3.4. It has the states $0, 1, 2, ...$, and the only non-zero transition rates are $a_{i,i+1} = \lambda_i$ and $a_{i,i-1} = \mu_i$.

$$
\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ \\ \\ \\ \end{array}
\begin{array}{c}
\begin{array}{ccccccc} 0 & 1 & 2 & 3 & & & \end{array} \\
\begin{bmatrix}
-\lambda_0 & \lambda_0 & 0 & 0 & . & . & . \\
\mu_1 & -(\mu_1 + \lambda_1) & \lambda_1 & 0 & . & . & . \\
0 & \mu_2 & -(\mu_2 + \lambda_2) & -\lambda_2 & . & . & . \\
0 & 0 & \mu_3 & -(\mu_3 + \lambda_3) & . & . & . \\
. & . & . & . & . & . & . \\
. & . & . & . & . & . & . \\
. & . & . & . & . & . & .
\end{bmatrix}
\end{array}
$$

Figure 3.4: Transition matrix of a birth-death process

Computing the steady-state equations is straightforward. One must first find π_i (the elements of π denote the probabilities of being in phase i). All the π_i values are expressed in terms of π_0, using steady-state equations; and then π_0 is determined such that the sum of all probabilities equals 1. This can be done using the matrix given in figure 3.4.

$$0 = -\lambda_0\pi_0 + \mu_1\pi_1$$

$$0 = \lambda_0\pi_0 - (\mu_1 + \lambda_1)\pi_1 + \mu_2\pi_2$$

$$0 = \lambda_1\pi_1 - (\mu_2 + \lambda_2)\pi_2 + \mu_3\pi_3$$

.

.

.

$$0 = \lambda_{i-1}\pi_{i-1} - (\mu_i + \lambda_i)\pi_i + \mu_{i+1}\pi_{i+1}$$

.

.

.

These equations hold for $i = 0, 1, 2, ...$; solving the equations for π_{i+1} results in:

$$\pi_{i+1} = \frac{\lambda_i}{\mu_{i+1}}\pi_i \tag{3.2}$$

This is known as the matrix-geometric solution due to the geometric relation between the stationary probabilities.

3.5 Kendall's Notation

Commonly investigated types of queuing systems make some scholars suggest notations to describe the queuing systems. The Kendall's notation was defined in the mid-twentieth century by D. G. Kendall [65], and is among the most widely used approaches to summarize queuing systems. Kendall's notation describes the type of queues and their characteristics. Consider a system is denoted by:

$$A/B/m/K/n/D,$$

with parameters defined as in table 3.1.

Table 3.1: Kendall's notation description

Parameter	Description
A	Distribution function of the inter arrival times
B	Distribution function of the service times
m	Number of servers
K	Capacity of the system; the maximum number of customers in the system including the ones being served
n	Population size; number of sources of customers
D	Service discipline

Exponentially distributed random variables are notated by M, meaning *Markovain* or *memory-less*. Furthermore, if the population size or the capacity are infinite, and the service discipline is FCFS, then their notations are omitted.

In many resources n is removed; researchers usually show the system by five items: $A/B/m/K/D$. Various types of queues based on these elements are presented in table 3.2.

3.6 Measures of System Performance

Usually, in a queuing system customers arrive, wait for service, receive service, and then leave the system. On the other hand, the managers need to select the best configuration of the queuing system. Therefore, they need some performance criteria to compare the queuing systems against each other and choose the best one. Generally, there are three vital measures for managers to analyze:

1. the customer's waiting time;

2. the number of customers accumulated in the queue;

3. the idle time of the servers

Table 3.2: Types of queuing systems based on Kendall's notation [1]

Characteristic	Symbol	Explanation
Arrival and service patterns (A, B)	M	Exponential
	D	Deterministic
	m	Erlang type of $k = 1, 2, ...$
	H_k	Mixture of k exponentials
	PH	Phase-type
	G	General distribution
Number of parallel servers (m)	$1, 2, ..., \infty$	
Maximum system capacity (K)	$1, 2, ..., \infty$	
Service discipline (D)	FCFS	First come, first served
	LCFS	Last come, first served
	RSS	Random selection for service
	PR	Priority
	GD	General discipline

Measures of waiting time

Since randomness is an inevitable attribute of queuing systems, the above measures can be stated as random variables. In other words, for every measure a researcher can find a probability distribution or at least the expected value of their respective random variables.

Another measure, which is related to processing and waiting time is the time a customer stays in the system, i.e., the sum of waiting time and processing time. Depending on the system under study, one of these measures (wait time or stay time) may be more important than the other. For example, in a healthcare setting stay time analysis is very interesting. When analyzing a bank, one desires to keep the wait time as short as possible [1]. Throughout this research, the average waiting time of a typical customer in a queue is W_q, and the average waiting time in the system is W. Evidently, $W \geq W_q$.

Measures of customer accumulation and idle time

There are two customer accumulation measures: 1) the number of customers in the queue; and, 2) the number of customers in the system. These two factors can be very helpful in designing a waiting room or space for customers or entities entering the system. L_q indicates the average number of customers in the queue; L denotes the average number of customers in the system.

Finally, idle time measures show the percentage of time that the entire system is empty of customers [1].

Queueing system analysis

The responsibility of a queuing analyst is computing a performance measure of the system and selecting the best design based on the defined criteria. Therefore, the analyst may determine waiting time and queue length. Also, s/he might want to balance the average customer waiting time and the idle time of the servers considering the involved costs; it may be possible to calculate the optimum number of servers by comparing the cost of waiting for customers, and the cost of idling for servers such that a trade-off between wait time costs and server costs is reached.

For designing the waiting facility, it is crucial to collect information regarding the possible length of the queue. In this case, the system encounters two costs; the cost of purchasing or renting space versus the cost of refusing to accept customers.

3.7 Little's Law

Assume $\alpha(t)$ denote the number of arrivals to the system up to time t. Also, the notation $\delta(t)$ is used for the number of departures from the system before time t. If the system starts empty, then $N(t)$ is the number of customers in the system for all times t (see figure 3.5).

$$N(t) = \alpha(t) - \delta(t)$$

Thus:

$$L = \frac{\int_0^t N(t)dt}{t} = \frac{\int_0^t (\alpha(t) - \delta(t))\, dt}{t}$$

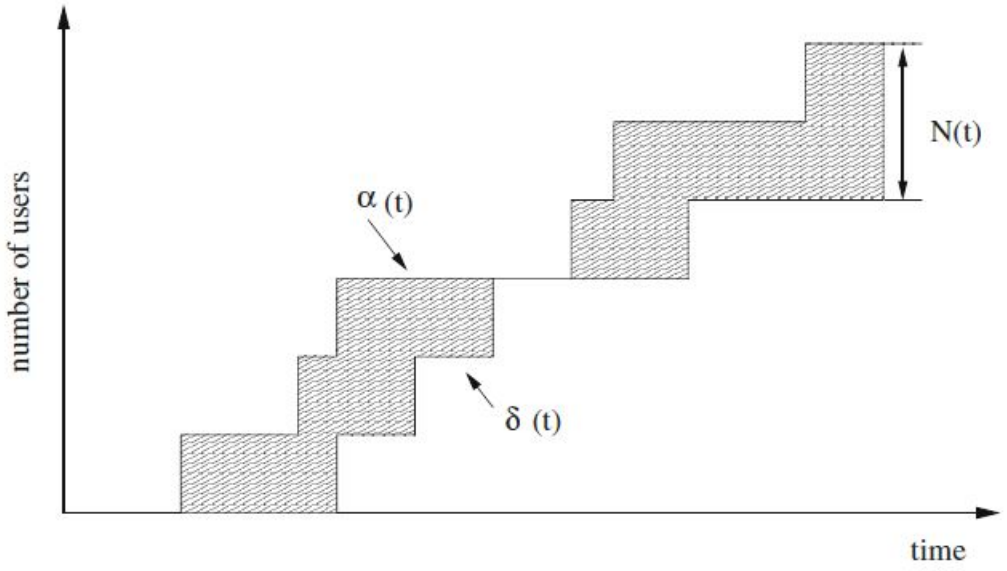

Figure 3.5: Number of customers in the system at time t [1]

Little's law is a fundamental relationship used extensively in queueing theory. Little's law provides a relationship between three fundamental quantities: 1) λ, the average rate of customer arrival; 2) W, the average time a customer spends in the system; and, 3) L, the average number of customers in the system:

$$L = W\lambda$$

If two of the three quantities are given, one can infer the third. For example, if one can observe customers leaving a store (yielding an estimate for λ), and one can ask each customer how long s/he has been in the store (estimating W), then one can estimate L, the average number of customers in the store. Little's law can be applied to a wide variety of systems, even ones that might not be considered queues.

3.8 Steps of Analyzing Queues

Analytical methods, simulation and Markov chains have been employed to analyze complicated queueing systems. In particular, Markov chains can be used as a tool to create numerical models for complex systems based on the following steps:

1. developing the state-space of the system;

2. creating equations that describe the Markov chain;

3. computing the steady-state probabilities of the Markov chain;

4. measuring system performance.

The most difficult steps are obtaining the set of all the possible states of a system and the transition matrix between them [31].

For example, assume that one desires to analyze a queue system with two types of customers. First step is defining the state-space. If n_1 is the number of customers in the queue for customer type 1, and n_2 indicates the number of customers in group 2, the state-space is shown by (n_1, n_2). Using Little's law one can calculate the average number of customers in the system (L_i), the average waiting time in the system (W_i), and the average waiting time in the queue for group 1 as below:

$$L_1 = \sum_{n_1} n_1 \times \pi(n_1, n_2) \tag{3.3}$$

$$W_1 = \frac{L_1}{\lambda_1} \tag{3.4}$$

$$W_{q_1} = \frac{L_1}{\lambda_1} - \frac{1}{\mu_1} \tag{3.5}$$

where $\pi(n_1, n_2)$ is the steady-state probability of the system state.

3.9 Markov Chain Process

Sefozo [66] described that a sequence of random variables $X_0, X_1, ...$ with values in a countable set S is a Markov chain if at any time n, the future states (or values) $X_{n+1}, X_{n+2}, ...$ depend on the history $X_0, ..., X_n$ only through the present state X_n. Markov chains are stochastic processes with diverse applications.

3.9.1 The Exponential Distribution

The exponential distribution has the following Probability Distribution Function (PDF):

$$f(x) = \lambda e^{-\lambda x} \qquad x \geq 0 \tag{3.6}$$

This distribution is well known in applied probability for several reasons. First and foremost, it has the memoryless property:

$$P(X > t + s | X > t) = P(X > s) \qquad \forall t, s \geq 0 \tag{3.7}$$

This property means if the "past" and "present" are known, then the "future" depends only on the "present" but not on the "past"; hence, many of the random processes can be modeled using this property to analyze the performance criteria [3]. The memoryless attribute of the exponential distribution can also be used in Markov process modeling [67].

3.10 $M/M/C$ Queue Model

As Grassman [64] states, a single queue with more than one server is referred to as a multi-server queue. Multi-server queues are frequently used to analyze many situations in service industries. Here, the queue is formed by the customers waiting for a service.

Commonly, it is assumed that multi-server queues have a constant arrival rate, λ, and each

server has a constant service rate of μ. So, the departure rate is μ times the number of busy servers.

Lemma 1. Suppose that $X_1, X_2, ..., X_k$ are independent, identical, exponential random variables with mean μ^{-1}, and consider the corresponding order statistics:

$$X_{(1)} \leq X_{(2)} \leq ... \leq X_{(k)}$$

Then $X_{(1)}$ is exponentially distributed with mean $\frac{1}{k}$ times the mean of the original random variables [31].

Proof: note that:

$$X_{(1)} = \min\{X_1, X_2, ..., X_k\}$$

$X_{(1)} > x$ if and only if $\forall i \in \{1, 2, ..., k\} \rightarrow X_i > x$. Hence:

$$\begin{aligned} P\{X_{(1)} > x\} &= P\{X_1 > x\}P\{X_2 > x\} \dots P\{X_k > x\} \\ &= \left(e^{-\mu x}\right)^k \\ &= e^{-k\mu x}. \end{aligned}$$

Figure 3.6 immediately follows Lemma 1. According to this figure, the service rate for states smaller than c is equal to $i\mu$.

$$\mu = \begin{cases} i\mu & i < c \\ c\mu & i \geqslant c \end{cases} \tag{3.8}$$

where c is the number of available servers. For multi-server queues, π_i can be calculated as follows:

$$\pi_i = \frac{\lambda_0}{\mu_1} \times \frac{\lambda_1}{\mu_2} \times \ldots \times \frac{\lambda_i - 1}{\mu_i} \times \pi_0 = \frac{\lambda}{\mu} \times \frac{\lambda}{2\mu} \times \ldots \times \frac{\lambda}{i\mu} \times \pi_0 = \frac{\lambda^i}{i!\mu^i}\pi_0 \qquad i \leqslant c \tag{3.9}$$

$$\pi_i = \frac{\lambda_0}{\mu_1} \times \frac{\lambda_1}{\mu_2} \times \ldots \times \frac{\lambda_i - 1}{\mu_i} \times \pi_0 = \frac{\lambda^c}{c!\mu^c}\left(\frac{\lambda}{\mu}\right)^{i-c} \pi_0 \qquad i \geqslant c \tag{3.10}$$

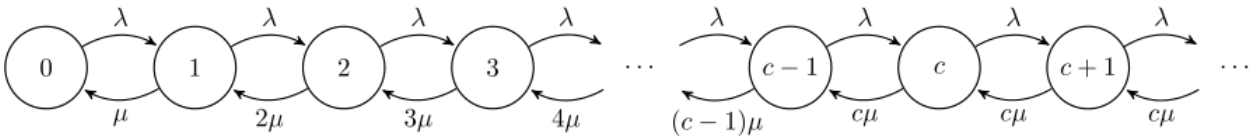

Figure 3.6: State diagram for an $M/M/C$ queue model

Note that if $i = c$, either equation can be used. To find π_0, one can use the fact that the sum of all transition probabilities equals 1. Thus:

$$1 = \sum_{i=0}^{\infty} \pi_i = \sum_{i=0}^{c-1} \frac{\lambda^i}{i!\mu^i}\pi_0 + \sum_{i=c}^{\infty} \frac{\lambda^c}{c!\mu^c}\left(\frac{\lambda}{\mu}\right)^{i-c}\pi_0 \tag{3.11}$$

in other words:

$$1 = \pi_0\left[\sum_{i=0}^{c-1} \frac{\lambda^i}{i!\mu^i} + \frac{\lambda^c}{c!\mu^c}\frac{1}{1-\frac{\lambda}{c\mu}}\right] \tag{3.12}$$

to simplify, define:

$$C_i = \frac{\lambda^i}{i!\mu^i} \tag{3.13}$$

therefore:

$$\pi_0 = \frac{1}{\left[\sum_{i=0}^{c-1} C_i + \frac{C_c}{1-\frac{\lambda}{c\mu}}\right]} \tag{3.14}$$

Frequently, researchers are interested in analyzing the waiting times or how long the customers or entities in a queue have to wait until they receive their services. This application is observed in many situations such as telephone reservation systems, amusement parks, check-in counters, tollgates, airports, restaurants and so on. Sometimes, factors such as the total time that an element spends in the system are interested; this can be seen in the health industry such as Length Of Stay (LOS) [68]. Thus, there are two terms when analyzing the time in a queue system [64]:

1. **Waiting time:** the time spent in the queue, excluding the service time.

2. **Sojourn time:** the time spent in the system, including the service time.

For calculation purposes, formulas exist for L and L_q, i.e., the expected number of elements in

the system and queue, and W and W_q, i.e., the average time spent in the system or the queue [64]. To write these formulas more concisely, define B or the probability that all servers are busy:

$$B = \sum_{i=c}^{\infty} \pi_i = \sum_{i=c}^{\infty} C_c \pi_0 \left(\frac{\lambda}{c\mu}\right)^{i-c} \tag{3.15}$$

then:

$$W_q = \frac{B}{c\mu - \lambda} \tag{3.16}$$

$$L_q = \frac{B\lambda}{c\mu - \lambda} \tag{3.17}$$

$$W = W_q + \frac{1}{\mu} \tag{3.18}$$

$$L = L_q + \frac{\lambda}{\mu} \tag{3.19}$$

3.11 The Proposed Mixed-Integer Non-Linear Program

Assume that there are several customers in a number of priority groups, which are served only by a specific subset of the servers. In other words, every group forms a separate and independent queue and only specific servers process the customers of a particular group. Figure 3.7 shows the structure of this system. The number of servers dedicated to each group is determined based on characteristic of the customers of that group.

3.11.1 Assumptions

1. Queue discipline for all groups is first come, first served (FCFS).
2. Customers of group j arrive according to a Poisson process with rate λ_j.
3. Service time of group j is exponentially distributed with mean μ_j.
4. All the queues are non-preemptive.

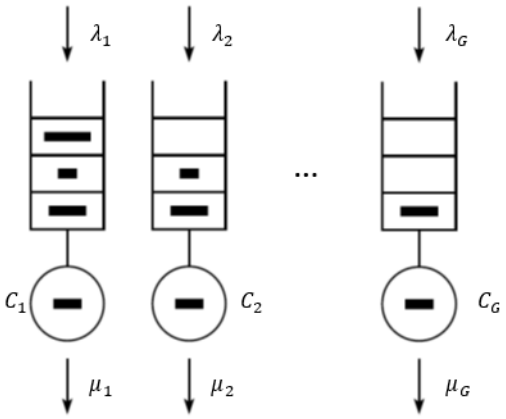

Figure 3.7: Model A - a queueing system with dedicated servers

5. A server serves only one customer at a time.

6. Balking, reneging, collusion or jockeying do not exist.

3.11.2 Notations

Sets:

J	set of groups; $\|J\| = G$
I	set of servers

Parameters:

S	total number of available servers; $\|I\| = S$
T	a sufficiently large number
w_j	the weight of wait for group j ; $j = 1, ..., G$
λ_j	arrival rate of group $j; j = 1, ..., G$
μ_j	service rate of group $j; j = 1, ..., G$

Variables:

W_{q_j}	average wait time of group $j; j = 1, ..., G$
$x_{ij} = 1$	if the ith server is assigned to group $j; j = 1, ..., G; i = 1, ..., S$
a_j	total number of servers assigned to group $j; j = 1, ..., G$
r_{ji}, k_{ji}	dummy variables; $j = 1, ..., G; i = 1, ..., S$

3.11.3 The Model

Using the above notations, the developed model is as follows:

Objective:

$$\min \quad Z = \sum_{j=1}^{G} w_j . W_{q_j} - \sum_{j=1}^{G} \sum_{i=0}^{S} r_{ji} + \sum_{j=1}^{G} \sum_{t=0}^{T} k_{jt} \tag{3.20}$$

Subject to:

$$W_{q_j} = \frac{B_j}{a_j\mu_j - \lambda_j} \qquad j = 1, ..., G \qquad (3.21)$$

$$\pi_{0j} = \frac{1}{\sum_{i=0}^{S} r_{ji} \frac{\lambda_j^i}{i!\mu_j^i} + \frac{\lambda_j^{a_j}}{a_j!\mu_j^{a_j}}} \qquad j = 1, ..., G \qquad (3.22)$$

$$B_j = \sum_{t=0}^{T} k_{jt} \frac{\lambda_j^{a_j}}{a_j!\mu_j^a} \pi_{0j} \frac{(\frac{\lambda_j}{\mu_j})^{(t-a_j)}}{a_j!} \qquad j = 1, ..., G \qquad (3.23)$$

$$\sum_{i=0}^{S} x_{ij} = a_j \qquad j = 1, ..., G \qquad (3.24)$$

$$\sum_{i=0}^{S} x_{ij} \leq 1 \qquad j = 1, ..., G \qquad (3.25)$$

$$i \times r_{ji} < a_j \qquad i = 1, ..., S \qquad j = 1, ..., G \qquad (3.26)$$

$$t \times (1 - k_{jt}) \leq a_j \qquad t = 1, ..., T \qquad j = 1, ..., G \qquad (3.27)$$

$$W_{q_j} \geq 0 \qquad j = 1, ..., G \qquad (3.28)$$

$$x_{ij}, r_{ji}, k_{jt} \in \{0, 1\} \qquad i = 1, ..., S \qquad t = 1, ..., T \qquad j = 1, ..., G \qquad (3.29)$$

Equation 3.20 defines the Weighted Sum of Average Waiting Times (WSAWT). Constraints 3.21 - 3.23 compute the waiting times for each group. Constraint 3.24 calculates the number of servers assigned to each group. Constraint 3.25 ensures that all servers are assigned to a specific group. Constraints 3.26 and 3.27 define the relationship between the number of assigned servers to each group and the probabilities that the system is empty or busy.

3.12 Solving the Independent Queues

Solving the proposed model to optimality, if at all possible, is time-consuming and complicated even for small problems due to its nonlinear constraints. Accordingly, the problem is coded by and solved by Python. An enumeration method is used to compute the weighted sum of average waiting times for the problem. Since there are G groups in the system with S servers, the number of possible solutions equals the number of positive and integer solutions of equation (3.30), i.e., $\binom{S-1}{G-1} = \frac{(S-1)!}{(G-1)!}$.

$$x_1 + x_2 + ... + x_G = S \qquad x_j \geq 1 \tag{3.30}$$

3.12.1 Numerical Example

Assume a system with five priority groups ($G = 5$), and 13 servers ($S = 13$). Each group has an independent arrival rate and service rate. Table 3.5 shows the characteristics of each group.

Table 3.3: Group specification

Group	λ_j	μ_j	Weight assigned to wait (w_j)
1	0.5	0.125	0.25
2	0.5	0.2	0.3
3	0.2	0.25	0.1
4	0.3	0.35	0.2
5	0.5	0.3	0.15

The first group has the longest service time among others. The mean service time for customers of the first group is $\frac{1}{\mu_1} = \frac{1}{0.125} = 8$. On the other hand, one arrival from this group is expected every two minutes ($\frac{1}{\lambda_1} = \frac{1}{0.5} = 2$). The second group incurs the highest cost of waiting between all groups.

The number of servers assigned to each group and the average waiting times of the groups, after minimizing WSAWT, are depicted in table 3.4. Also the table helps to find the average number of customers in the system. It is expected that the system has the same number of customers for groups

1 and 2. In addition, the optimum value of WSAWT is with the objective function $Z = 6.077$.

Table 3.4: Optimal solution for the problem

Group (j)	Optimal number of servers (a_j) for group j	Average number of customers (L) for group j	Average waiting time (W_j) for group j	Weighted average waiting time for group j
1	5	6.21	4.43	1.107
2	3	6.01	7.02	2.106
3	1	4	16	1.600
4	2	1.05	0.64	0.128
5	2	5.45	7.57	0.135

3.12.2 Calculating the Minimum Number of Required Servers

Desirable queue systems remain stable in the long run. Therefore, one of the most important problems is finding the minimum number of servers that keep the system stable. Equation (3.31) finds the minimum number of servers for the dedicated server system.

$$S_{\min} = \sum_{j=1}^{J} \left(\lfloor \frac{\lambda_j}{\mu_j} \rfloor + 1 \right) \tag{3.31}$$

According to this equation, the minimum number of servers to keep the analyzed system stable is $S = 12$.

3.12.3 Effect of Number of Servers on the Weighted Sum of Average Waiting Times

The goal is to measure the effect of increasing the number of servers assigned to each group. For this purpose, the considered number of servers varies between $S_{min} = 12$ and $S = 20$.

Figure 3.8 shows the optimum value for the weighted sum of average waiting times (WSAWT) based on the total number of available servers. As expected, when the total number of available servers increases, the system experiences a reduction of the measured variable. It can be verified

that the slope of decline of the objective function value decreases as more and more servers are hired. Also, table 3.5 indicates the optimal number of servers in each states. Figure 3.9 is a box-plot of the average waiting times for each group. Increasing the number of servers will decrease the average waiting times for the customers. This figure implies that if the number of servers increases to 16 (S=16), the average waiting time for each group will be less than five minutes. The main focus of this figure is showing the average waiting time not WSAWT, and helps with selecting the desired number of servers based on waiting times.

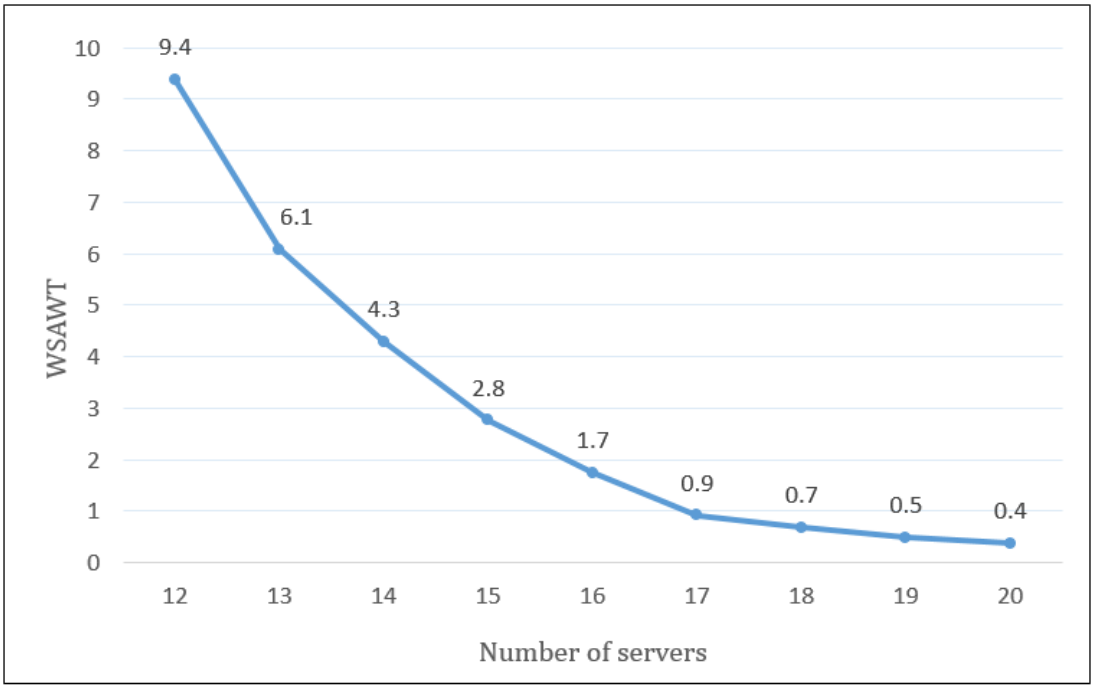

Figure 3.8: Optimum value of weighted sum of average wait times based on the total number of servers

Table 3.5: Effect of number of servers on WSAWT

Number of servers	Optimal number of servers (a_1) for group 1	Optimal number of servers (a_2) for group 2	Optimal number of servers (a_3) for group 3	Optimal number of servers (a_4) for group 4	**WSAWT**
12	5	3	1	1	9.377
13	5	3	1	2	6.077
14	5	4	1	2	4.289
15	5	4	2	2	2.756
16	5	4	2	2	1.742
17	6	4	2	2	0.917
18	6	5	2	2	0.677
19	7	5	2	2	0.4845
20	7	5	2	3	0.3725

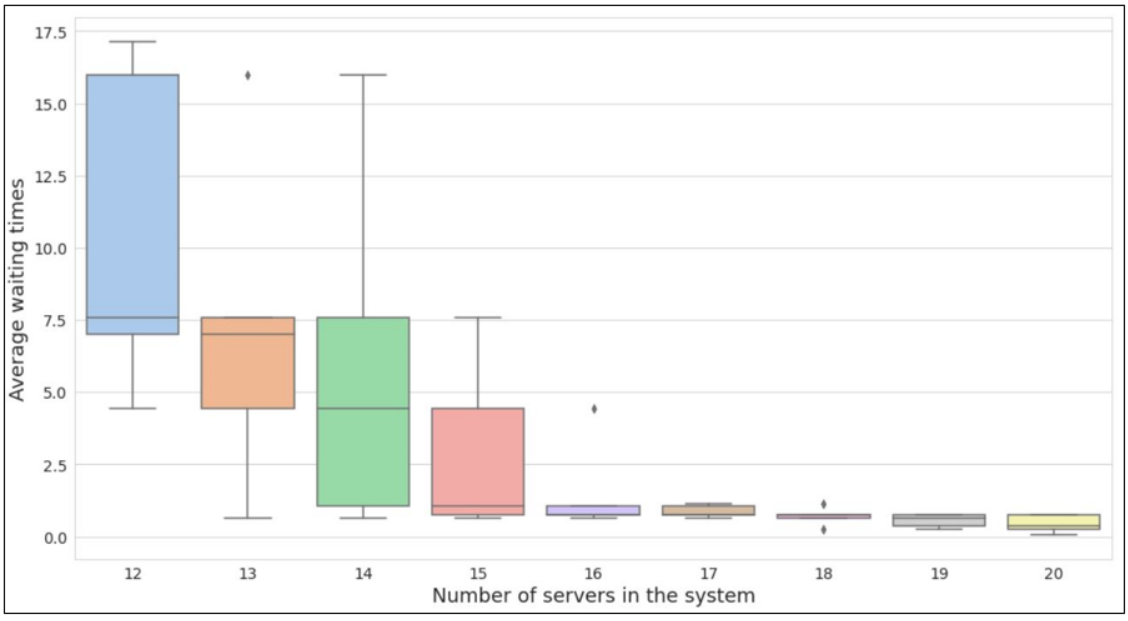

Figure 3.9: Average waiting times box-plot for customers

4. Systems with Merged Queues

4.1 Introduction

Sometimes the system has flexible servers or machines that are capable of serving several types of customers. These kinds of queue systems are common in industrial and service organizations such as hospitals, call centers, and factories. In this case, all customers share a single queue with shared servers who process the requests based on a specific prioritization regime. Before diving into the problem, some related concepts should be elaborated in the following sections.

4.2 Matrix-Analytic Methods

Matrix-Geometric Methods (MGM) and Matrix-Analytic Methods (MAM) are well-known as some of the most efficient achievements in computational probability [69]. MAM can solve a wide variety of stochastic problems in which the transition matrix has a repeating structure and a state space can have up to one infinite dimension. Such methods help discover and model a stochastic process by an iterative method [67].

Quasi-Birth-Deaths are known as Markov processes in two dimensions, the *level* and *phase*, to prevent the process from jumping across several levels in one transition. The process can transit from one level to another level. Generally, in QBDs the problem is solved by the help of finding blocks of states and transition sub-matrices.

The Continuous Time Markov Chain (CTMC), which corresponds to the $M/M/C$ queue of chapter 3, is a special case of a one-dimensional birth and death process (or chain). A one-dimensional birth and death chain is a CMTC, $\{X(t), t \geq 0\}$ on $S = 0, 1, 2, ...$ with generator matrix Q, which is of the below form [70]:

$$\mu = \begin{cases} \lambda_i & \text{if} \quad j = i + 1 \\ \mu_i & \text{if} \quad j = i - 1 \quad \text{and} \quad i > 0 \\ 0 & \text{otherwise} \end{cases} \qquad \forall i, j \in S \tag{4.1}$$

Essentially with every transition, the stochastic process jumps to a state with one more entity (birth) or with one less entity (death). The rates $\lambda_0, \lambda_1, ...$ are known as birth rates and the rates $\mu_1, \mu_2, ...$ are known as death rates. For an $M/M/1$ queue, all birth rates are equal to λ and all death rates are equal to μ. The steady-state distribution of the one-dimensional birth and death process is easy to compute. There are several queuing systems with Poisson arrivals and exponentially distributed service times that can be modeled as multi-dimensional CTMCs. With this introduction, next a Quasi-birth-death process is defined and compared it with a birth-death process.

4.3 Quasi-Birth-Death Processes

As mentioned before, many real-world problems can be modeled by multi-dimensional Markov chains. Demonstrating the state-space plays a crucial role in Markov chain modeling; sometimes a clever way of showing the states and their number leads to a chain with a block-tridiagonal transition probability matrix, known as QBD [71]. QBDs apply to a diverse set of biological [72, 73], social [74], economic [75], and technical processes such as cell growth, biochemical reaction kinetics, epidemics, demographic trends, and queueing systems among others [76]. A QBD process is a CTMC whose infinitesimal generator matrix Q or transition matrix, showing transition rates between the states, is of the following block-diagonal form [70]:

$$Q = \begin{bmatrix} B_1 & B_0 & 0 & 0 & 0 & 0 & \cdots \\ B_2 & A_1 & A_0 & 0 & 0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & 0 & 0 & \cdots \\ 0 & 0 & A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & 0 & 0 & A_2 & A_1 & A_0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where $A_0, A_1, A_2, B_0, B_1, B_2$ are finite-sized matrices (instead of scalars, which is the case in one-dimensional birth and death processes). Any queuing system that has a generator of this form is called a QBD process.

Usually, QBD processes can be applied to queueing systems for, when considering the system as a whole, customers arrive one by one and according to a Poisson process (unless they are in batches), and get served one after the other (hence depart from the system one by one). The total number of entities in the system most likely goes up or down by one unit, and thereby, in that dimension one does have a birth-death-like structure [70]. To clarify QBD's applications in queueing systems, two examples are presented.

Example 1.

Consider a coffeehouse with two clerks; arrival pattern for customers is a Poisson process with rate λ. Once the customers enter the store, they join the shorter queue to receive their service. If both queues have the same number of customers, then a new customer selects one of the lines by flipping a coin. The service time has an exponential distribution with mean $\mu_i; i = 1, 2$. As soon as a customer is served and exits the system, if the other queue has two or more customers waiting in line, the customers jockey to the smaller line.

Let $X_1(t)$ and $X_2(t)$ be the number of customers in queues 1 and 2 at time t. Assuming that $X_1(0) = 0$ and $X_2(0) = 0$, and $|X_1(t) - X_2(t)| \leq 1$ for all t. One can model the bi-variate stochastic process $\{(X_1(t) , X_2(t)), t \geq 0\}$ as a *QBD* by obtaining A_0, A_1, A_2, B_0, B_1, and B_2; the first step is making the state-space. The bi-variate CTMC $\{(X_1(t) , X_2(t)), t \geq 0\}$ has the following state-space:

$$S = \{(0, 0), (1, 0), (0, 1), (1, 1), (2, 1), (1, 2), (2, 2), (3, 2), (2, 3), ...\}$$

By writing down the Q matrix of the CTMC, and considering sets of three states as a "level" with three "phases," it is easy to verify that Q has a QBD structure with the following sub-matrices:

$$A_0 = B_0 = \begin{bmatrix} 0 & 0 & 0 \\ \lambda & 0 & 0 \\ \lambda & 0 & 0 \end{bmatrix}$$

$$A_2 = B_2 = \begin{bmatrix} 0 & \mu_2 & \mu_1 \\ 0 & 0 & \mu_1 \\ 0 & 0 & \mu_2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} -\lambda - \mu_1 - \mu_2 & \lambda/2 & \lambda/2 \\ \mu_1 + \mu_2 & -\lambda - \mu_1 - \mu_2 & 0 \\ \mu_1 + \mu_2 & 0 & -\lambda - \mu_1 - \mu_2 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -\lambda & \lambda/2 & \lambda/2 \\ \mu_1 & -\lambda - \mu_1 & 0 \\ \mu_2 & 0 & -\lambda - \mu_2 \end{bmatrix}$$

Sometimes some of these blocks can be equal to other sub-matrices. For examples, in this problem $A_0 = B_0$ and $A_2 = B_2$.

Example 2.

Consider a bank with two groups of customers and one teller. An arriving customer belongs to the high-priority group with probability α, or to the low-priority group with probability $\beta = 1 - \alpha$. The service rate is dependent on type of customer, i.e., $\mu_i; i = 1, 2$. The policy for serving customers is first come first served.

To show the state-space for this Markovian process, consider $X(t)$ to be the total number of customers at any given time t, and $Y(t)$ be the class of customer in service (with $Y(t) = 0$ if there are no customers in the system). The problem can be modeled as a QBD bi-variate stochastic process $\{(X(t), Y(t)), t \geq 0\}$ by obtaining A_0, A_1, A_2, B_0, B_1, and B_2. The bi-variate CTMC $\{(X(t), Y(t)), t \geq 0\}$ has the following state-space depicted in 4.1:

$$S = \{(0,0),(1,1),(1,2),(2,1),(2,2),(3,1),(3,2),...\}$$

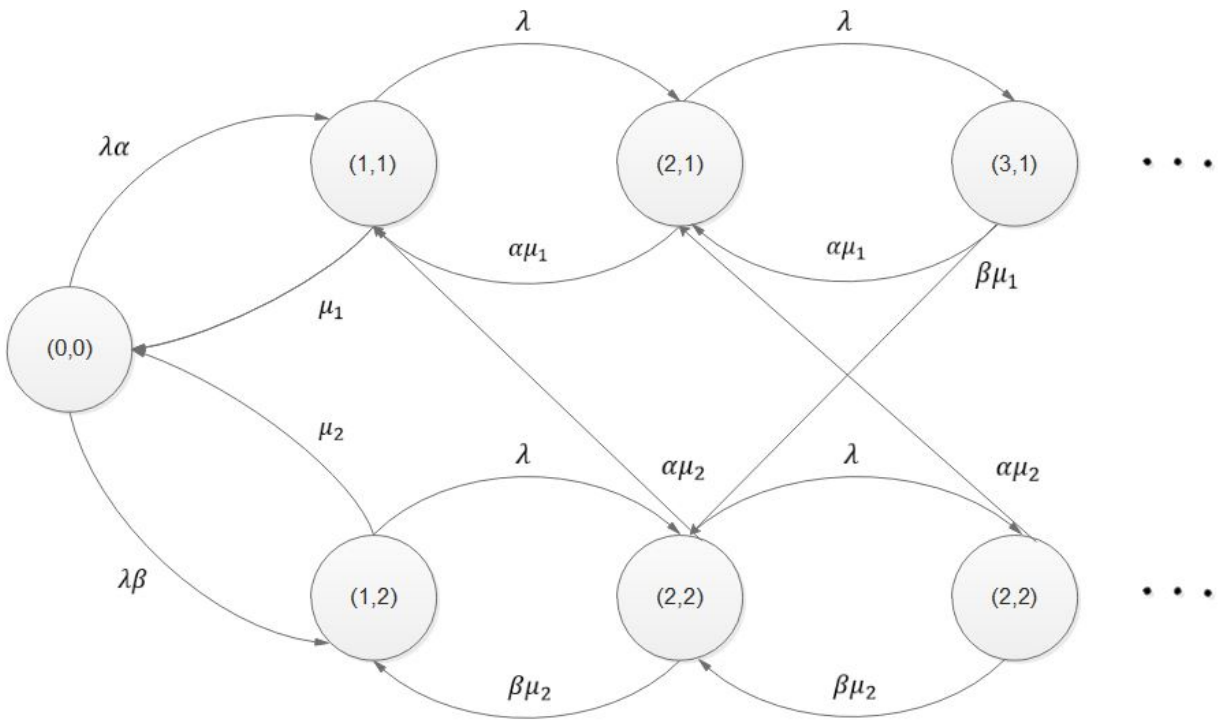

Figure 4.1: State-space for a bank with two groups of customers

By writing down the Q matrix of the CTMC and considering the sets of two states as "levels" with two "phases," Q will have a QBD structure with $l = 3, m = 2$.

$$A_0 = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 0 & 0 \\ \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \alpha\mu_1 & \beta\mu_1 \\ \alpha\mu_2 & \beta\mu_2 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & \alpha\mu_1 & \beta\mu_1 \\ 0 & \alpha\mu_2 & \beta\mu_2 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} -\lambda - \mu_1 & 0 \\ 0 & -\lambda - \mu_2 \end{bmatrix}$$

$$B_1 = \begin{bmatrix} -\lambda & \alpha\lambda & \beta\lambda \\ \mu_1 & -\lambda - \mu_1 & 0 \\ \mu_2 & 0 & -\lambda - \mu_2 \end{bmatrix}$$

4.4 Matrix-Geometric Method for QBD Analysis

Matrix-Geometric Method (MGM) is an approach widely used for optimal analysis of frequently encountered types of queuing systems modeled as QBDs whose transition matrix has a repetitive block structure. MGM is useful not only for finding solutions of continuous and discrete-time QBDs, but also for a wide range of systems that include repetitive structures. MGM can be applied if the system can be decomposed into two parts: the initial and the repetitive portions which are typically found in all QBDs. To perform MGM-based analysis, assume that the CTMC with generator Q is irreducible [70]. MGM methods allow efficient analysis of Markov chains on state-spaces which are infinite in one dimension and finite in another dimension.

This section presents an algorithm for solving simple QBDs. Section 4.12 explains a Quadratic method for solving more complicated QBDs. Note that more general, yet similar, matrix-analytic methods can be used in non-QBD contexts. For example, multi-dimensional Markov chains are defined as chains with infinite dimensions in every or some of the directions.

To illustrate the MGM method, the following example is selected from [77], in which one-step transitions are restricted to phases within the same level or between phases of adjacent levels; the resulting QBD Markov chain is depicted in figure 4.2. This QBD is a birth-death process in a random environment with $m + 1$ different states, ordered lexicographically according to the following levels and phases:

$$S = \{(0,0),(0,1),...,(0,m),(1,0),(1,1),...,(1,m),(2,0),...\}$$

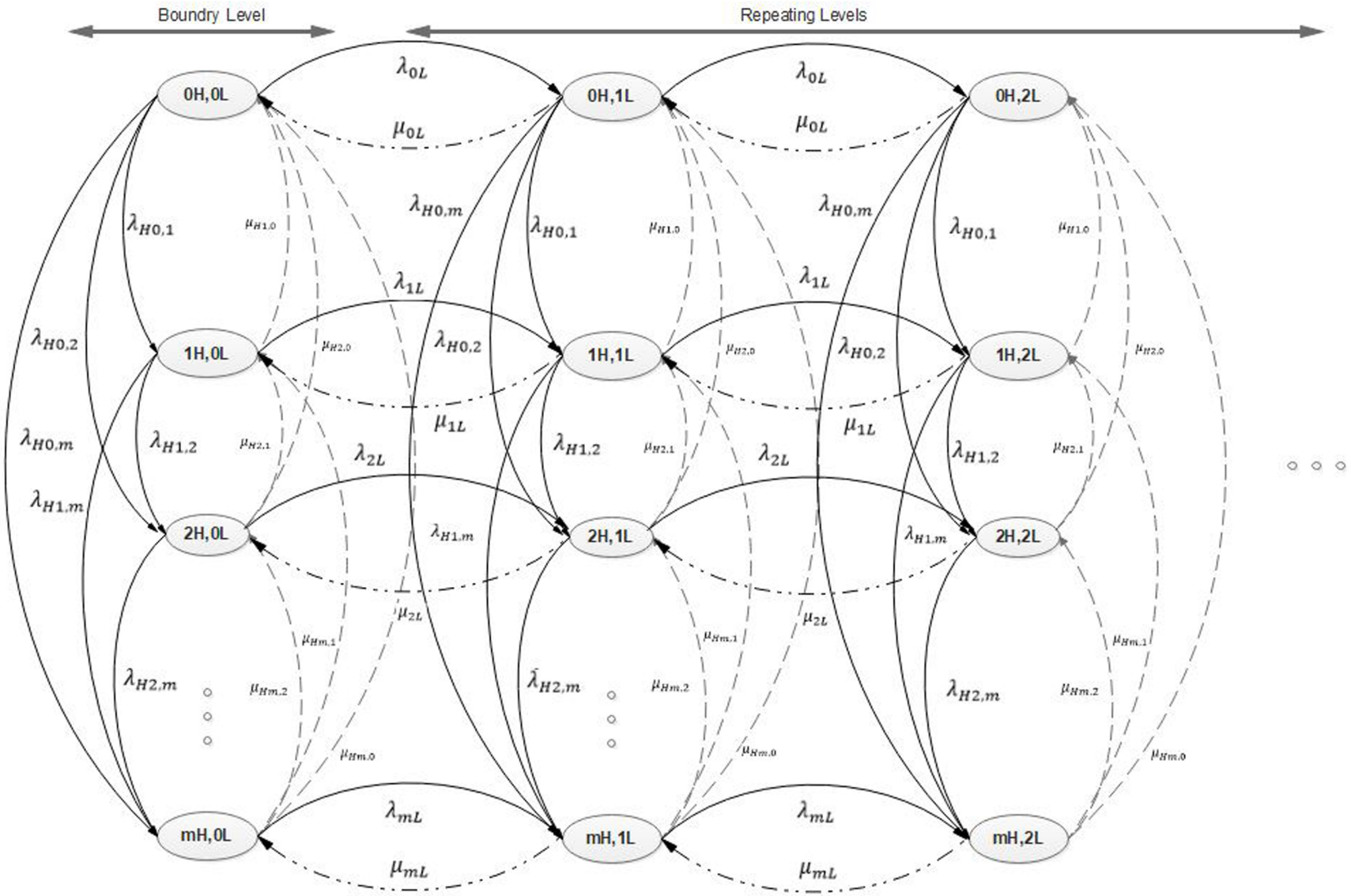

Figure 4.2: Quasi-birth-death process

The resulting structured generator matrix Q is as follows:

$$
Q = \left[\begin{array}{cccc|cccc|ccc}
a_0 & \lambda_{H0,1} & \cdots & \lambda_{H0,m} & \lambda_{0L} & 0 & \cdots & 0 & 0 & 0 & \cdots \\
\mu_{H1,0} & a_0 & \cdots & \lambda_{H1,m} & 0 & \lambda_{1L} & \cdots & 0 & 0 & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\
\mu_{Hm,0} & \mu_{Hm,1} & \cdots & a_m & 0 & 0 & \cdots & \lambda_{mL} & 0 & 0 & \cdots \\
\hline
\mu_{0L} & 0 & \cdots & 0 & b_0 & \lambda_{H0,1} & \cdots & \lambda_{H0,m} & \lambda_{0L} & 0 & \cdots \\
0 & \mu_{1L} & \cdots & 0 & \mu_{H1,0} & b_1 & \cdots & \lambda_{H1,m} & 0 & \lambda_{1L} & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & \cdots & \mu_{mL} & \mu_{Hm,0} & \mu_{Hm,1} & \cdots & b_m & 0 & 0 & \cdots \\
\hline
0 & 0 & \cdots & 0 & \mu_{0L} & 0 & \cdots & 0 & b_0 & \lambda_{H0,1} & \cdots \\
0 & 0 & \cdots & 0 & 0 & \mu_{1L} & \cdots & 0 & \mu_{H1,0} & b_1 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots
\end{array}\right]
$$

The off-diagonal elements of Q are given by the state transition rates of the QBD:

- $\lambda_{Hj,j'}$: Transition rate from phase $0 \leq j \leq m$ to phase $0 \leq j' \leq m$, $j < j'$, within a level.
- $\mu_{Hj,j'}$: Transition rate from phase $0 \leq j \leq m$ to phase $0 \leq j' \leq m$, $j > j'$, within a level.
- λ_{jL}: Transition rate from phase j of an arbitrary level $i \geq 0$ to phase j of level $i + 1$.
- μ_{jL}: Transition rate from phase j of an arbitrary level $i \geq 1$ to phase j of level $i - 1$.

Thus, the diagonal elements of Q are given by:

$$
a_j = \begin{cases}
-\lambda_{0L} - \sum_{n=1}^{m} \lambda_{H0,n} & j = 0 \\
-\lambda_{jL} - \sum_{n=0}^{j-1} \mu_{Hj,n} - \sum_{n=j+1}^{m} \lambda_{Hj,n} & 0 < j < m \\
-\lambda_{mL} - \sum_{n=0}^{m-1} \mu_{Hm,n} & j = m
\end{cases}
$$

and:

$$b_j = a_j - \mu_j \qquad\qquad 0 \leq j \leq m$$

To ensure that the elements of each row of Q sum up to 0. After defining the following sub-matrix for the boundary level $i = 0$ (here: $l = 1$):

$$B_{0,0} = \begin{bmatrix} a_0 & \lambda_{H0,1} & \cdots & \lambda_{H0,m} \\ \mu_{H1,0} & a_0 & \cdots & \lambda_{H1,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{Hm,0} & \mu_{Hm,1} & \cdots & a_m \end{bmatrix}$$

and the following sub-matrices for the repeating levels $i \geq 1$:

$$A_0 = \begin{bmatrix} \lambda_{0L} & 0 & \cdots & 0 \\ 0 & \lambda_{1L} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{mL} \end{bmatrix}$$

$$A_1 = \begin{bmatrix} b_0 & \lambda_{H0,1} & \cdots & \lambda_{H0,m} \\ \mu_{H1,0} & b_1 & \cdots & \lambda_{H1,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{Hm,0} & \mu_{Hm,1} & \cdots & b_m \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \mu_{0L} & 0 & \cdots & 0 \\ 0 & \mu_{1L} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_{mL} \end{bmatrix}$$

Hence, the generator matrix can be written in block form as below:

$$Q = \begin{bmatrix} B_{0,0} & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Note that different sources use different notations when dealing with matrix Q, specifically, the non-repetitive section of the matrix. For example, Guatam [70] uses the following matrix, divided into five different sub-matrices:

$$Q = \begin{bmatrix} B_{0,0} & A_0 & 0 & 0 & \cdots \\ A_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

On the other hand, Bolch [77] uses the following structure with six sub-matrices:

$$Q = \begin{bmatrix} B_1 & B_0 & 0 & 0 & \cdots \\ B_2 & A_1 & A_0 & 0 & \cdots \\ 0 & A_2 & A_1 & A_0 & \cdots \\ 0 & 0 & A_2 & A_1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

In this research for the defined problem, matrix Q is used by a different form of introduced by Bloch. Also, the first representation is used to show linearly convergent method.

Next step is computing the steady-state probabilities for the QBD process. The graph of the QBD is presented in figure 4.2. The probabilities are organized lexicographically into the steady-

state probability vector π:

$$\pi = [\pi_{0,0}, \pi_{1,0}, ..., \pi_{m,0}, \pi_{0,1}, \pi_{1,1}, ..., \pi_{m,1}, \pi_{0,2}, ...] \tag{4.2}$$

the vector π is partitioned according to levels:

$$\pi = [\pi_0, \pi_1, \pi_2, ...] \tag{4.3}$$

in each level the vector contains m states:

$$\pi_i = [\pi_{0,i}, \pi_{1,i}, ..., \pi_{m,i}] \tag{4.4}$$

π_i is an m dimensional vector, which describes the probability of being in state i. In other words, whereas in the birth and death process a scalar value exists for each state, in QBD the scalars are generalized to vectors of probabilities for each state. To calculate π, the following equation should be solved:

$$\pi Q = 0 \tag{4.5}$$

while obeying the normalization condition:

$$\pi \vec{1} = 1 \tag{4.6}$$

The repetitive structure of the QBD facilitates calculating the steady-state probabilities via the global balance equations for the repeating levels of $i \geq 1$:

$$\pi_{i-1} A_0 + \pi_i A_1 + \pi_{i+1} A_2 = 0 \qquad i > 0 \tag{4.7}$$

It is shown that the vector π_i can be defined in terms of the vector π_{i-1} [77]. In addition, the transitions among the levels of QBD are independent of level i. Therefore, a constant rate matrix **R**

can be introduced that leads to the matrix-geometric equation:

$$\pi_i = \pi_{i-1}\mathbf{R} = \pi_1\mathbf{R}^{i-1} \qquad i > 0 \tag{4.8}$$

To compute matrix $\mathbf{R}$, equation (4.8) is plugged in equation (4.5):

$$\pi_1\mathbf{R}^{i-2}A_0 + \pi_1\mathbf{R}^{i-1}A_1 + \pi_1\mathbf{R}^{i}A_2 = 0 \qquad i > 0 \tag{4.9}$$

$$\pi_1(\mathbf{R}^{i-2}A_0 + \mathbf{R}^{i-1}A_1 + \mathbf{R}^{i}A_2) = 0 \qquad i > 0 \tag{4.10}$$

multiplying the two sides by the inverse of $\mathbf{R}^{i-2}$ results in:

$$A_0 + \mathbf{R}A_1 + \mathbf{R}^2A_2 = 0$$

4.5 Steady-State Probabilities for a QBD Process

Latouche et al. [67] analyze three general-purpose iterative schemes for solving QBD problems. The first approach is linearly convergent with convergence rate equal to the maximum eigenvalues of matrix $\mathbf{R}$. The second algorithm is another linearly convergent method with much smaller rate of convergence; the last scheme is quadratically convergent. In linearly convergent methods, the error at one iteration is proportional to error in the next iteration. On the other hand, quadratically convergent algorithms converge faster relative to their linearly convergent counter-parts. In these algorithms, the *square* of error in one iteration is proportional to the error in the next iteration.

In section 4.5.1, the first linear method is presented. The quadratically convergent method is presented in section 4.12; this method is applied to the problem introduced in this research. To read more about the aforementioned algorithms, interested reader is referred to [67], chapter 8.

In finding the steady-state probabilities for a QBD process, the following characteristic function arises:

$$A_0 + \mathbf{R}A_1 + \mathbf{R}^2A_2 = 0 \tag{4.11}$$

where $\mathbf{R}$ is an unknown matrix and must be calculated. In some systems, the matrices A_0, A_1 and A_2 have simple structures and $\mathbf{R}$ can be obtained analytically. In addition, numerical methods to compute $\mathbf{R}$ are readily available. To present one approach, first rearrange the terms in equation (4.11) to obtain:

$$\mathbf{R} = -(A_0 + \mathbf{R}^2 A_2)A_1^{-1} \tag{4.12}$$

Equation (4.12), when calculated recursively, leads to a successive substitution scheme in which a guess is made for $\mathbf{R}$. This guess is used in the right-hand-side of the equation to produce a new estimate for $\mathbf{R}$. The new estimate is then used again in the right-hand-side to obtain another estimate of $\mathbf{R}$, and the process continues. This algorithm is explained in [78].

4.5.1 Successive Substitution Algorithm

Let A_0, A_1 and A_2 be sub-matrices of the generator Q, and let $A_s = A_0 + A_1 + A_2$. Define the vector v such that $vA_s = 0$ and $v1 = 1$, and define the traffic intensity as:

$$\rho = \frac{A_0 1}{vA_2}.$$

If $\rho < 1$, the following iteration procedure yields an approximation of matrix $\mathbf{R}$ that satisfies equation (4.12).

Step 1. Assure that the steady-state conditions hold, i.e., $\rho < 1$.

Step 2. Let $\mathbf{R}_0 = \mathbf{0}$ (i.e., a matrix of all zeros), let $n = 0$, and fix ε to be a small positive real value.

Step 3. Define the matrix $\mathbf{R}_{n+1}$ as:

$$R_{n+1} = -(A_0 + \mathbf{R}_n^2 A_2)A_1^{-1}.$$

Step 4. Define δ as:

$$\delta = \max_{i,j} \{|\mathbf{R}_{n+1}(i, j) - \mathbf{R}_n(i, j)|\}$$

Step 5. If $\delta < \varepsilon$, let $\mathbf{R} = \mathbf{R}_{n+1}$ and stop. Otherwise, increment n by one and return to Step 3.

The above successive algorithm is often slow to converge, especially if the traffic density, ρ, is close to one. However, programming this algorithm into a computer code is straightforward. Hence, the algorithm has the potential to become a convenient method for obtaining the matrix $\mathbf{R}$.

So far, matrix $\mathbf{R}$ is calculated, which enables computing the steady-state probabilities for states with repetitive structure. To derive π_0 and π_1, equation (4.5) can be employed. Hence the following equations can be shown:

$$\pi_0 B_{0,0} + \pi_1 A_2 = 0 \tag{4.13}$$

$$\pi_0 A_0 + \pi_1 A_1 + \pi_2 A_2 = \pi_0 A_0 + \pi_1 (A_1 + \mathbf{R} A_2) = 0 \tag{4.14}$$

also, based on equation (4.6):

$$\sum_{n=0}^{\infty} \pi_n \vec{1} = 1 \tag{4.15}$$

$$\pi_0 1 + \sum_{n=0}^{\infty} \pi_1 \mathbf{R}^n \vec{1} = 1 \tag{4.16}$$

$$\pi_0 1 + \pi_0 (\sum_{n=0}^{\infty} \mathbf{R}^n) \vec{1} = 1 \tag{4.17}$$

Consider:

$$\sum_{n=0}^{\infty} \mathbf{R}^n = \mathbf{I} + \mathbf{R} + \mathbf{R}^2 + ... \tag{4.18}$$

Multiplying both sides by $\mathbf{R}$ yields:

$$(\sum_{n=0}^{\infty} \mathbf{R}^n)\mathbf{R} = \mathbf{R} + \mathbf{R}^2 + \mathbf{R}^3 + ... \tag{4.19}$$

Subtracting these two equations result in:

$$\sum_{n=0}^{\infty} \mathbf{R}^n - (\sum_{n=0}^{\infty} \mathbf{R}^n)\mathbf{R} = \mathbf{I} \tag{4.20}$$

factoring out $\sum_{n=0}^{\infty} \mathbf{R}^n$ results in:

$$\sum_{n=0}^{\infty} \mathbf{R}^n(\mathbf{I} - \mathbf{R}) = \mathbf{I} \tag{4.21}$$

Thus:

$$\sum_{n=0}^{\infty} \mathbf{R}^n = \mathbf{I}(\mathbf{I} - \mathbf{R})^{-1} \tag{4.22}$$

Plugging equation (4.22) in equation (4.17) yields:

$$\pi_0 1 + \pi_1(\mathbf{I} - \mathbf{R})^{-1}\mathbf{1} = 1 \tag{4.23}$$

To summarize, steady-state probability vector of QBD can be obtained using equation (4.11) to derive $\mathbf{R}$, and equation (4.8) together with (4.14) and (4.23) to derive π. The calculations can be carried out in several ways. The following steps summarize one method of calculating the steady-state probabilities.

After computing $\mathbf{R}$, find π_0 and π_1 by solving:

$$\begin{cases} \pi_0 B_{0,0} + \pi_1 A_2 = 0 \\ \pi_0 A_0 + \pi_1(A_1 + \mathbf{R}A_2) = 0 \\ \pi_0 1 + \pi_1(\mathbf{I} - \mathbf{R})^{-1}\mathbf{1} = 1 \end{cases} \tag{4.24}$$

The above equation can be written in matrix multiplication format as below:

$$\begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} \begin{bmatrix} B_{00} & (A_0)^* & 1 \\ A_2 & (A_1 + RA_2)^* & (I - \mathbf{R})^{-1}1 \end{bmatrix} = \begin{bmatrix} \vec{0} & 1 \end{bmatrix} \tag{4.25}$$

$$\begin{bmatrix} \pi_0 & \pi_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} B_{00} & (A_0)^* & 1 \\ A_2 & (A_1 + RA_2)^* & (I - \mathbf{R})^{-1}1 \end{bmatrix}^{-1} \tag{4.26}$$

in which $(*)$ indicates that the last column of the matrix is removed and replaced by normalization condition (removing the last column and replacing it with a vector of ones) to avoid linear

dependence.

After computing π_0 and π_1 calculate π_i for $i \geq 2$ based the formula :

$$\pi_i = \pi_1 \mathbf{R}^{i-1}$$

If this algorithm is implemented to example 1 with parameters $\lambda = 0.3, \mu_1 = 0.1$ and $\mu_2 = 0.3$, the steady-state probabilities for states 0 and 1 can be calculated by equation (4.26):

$$\pi_0 = \begin{bmatrix} 0.118 & 0.168 & 0.056 \end{bmatrix}$$

$$\pi_1 = \begin{bmatrix} 0.168 & 0.076 & 0.049 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0.747 & 0.338 & 0.219 \\ 0.747 & 0.338 & 0.219 \end{bmatrix}$$

The steady-state probabilities for other states can be calculated using $\mathbf{R}$ and π_1. For example, to calculate π_2 and π_3:

$$\pi_2 = \pi_1 \mathbf{R} = \begin{bmatrix} 0.168 & 0.076 & 0.049 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0.747 & 0.338 & 0.219 \\ 0.747 & 0.338 & 0.219 \end{bmatrix}$$

$$\pi_2 = \begin{bmatrix} 0.094 & 0.042 & 0.028 \end{bmatrix}$$

and:

$$\pi_3 = \begin{bmatrix} 0.052 & 0.023 & 0.015 \end{bmatrix}$$

4.6 Using Stationary Probabilities to Analyze Performance

As mentioned in section 3.6, some criteria is required to compare systems in terms of their performance. The stationary or steady-state probabilities obtained above can be used to compute the defined performance criteria. To do this, first the average number of customers in the system should be calculated based on equation (4.27):

$$E[N] = \sum_{n=0}^{\infty} n\pi_n \vec{1} \tag{4.27}$$

Similarly, the kth moment of the number of customers in the system can be calculated:

$$E[N^k] = \sum_{n=0}^{\infty} n^k \vec{R_n} \vec{1} \tag{4.28}$$

Finally, the sojourn time can be found using Little's Law and $E[N]$:

$$E[T] = \frac{E[N]}{\lambda} \tag{4.29}$$

4.7 Phase-Type Distribution

Phase-type or PH distributions are a natural generalization of exponential and Erlang distributions and provide a versatile set of tractable models for applied probability and stochastic models. PH distributions provide a simple framework for modeling more complex models without losing computational tractability; they generate models that can be solved by matrix-analytic methods [64].

Distribution of time X before absorption to absorbing state 0 is called PH distribution and is represented as $PH(\tau, T)$.

Definition 3.1 A random variable has a Phase-type distribution with parameters m, G_* and α_* if it can be expressed as a first-passage time random variable for Markov process with state-space

$E = \{1, ..., m, \Delta\}$, generator:

$$G = \left[\begin{array}{c|c} G_* & G_\Delta \\ \hline \mathbf{0} & 0 \end{array}\right],$$

and initial probability vector $\alpha = (\alpha_*|0)$.

First-passage time random variables have well-known properties that can be described mathematically; however, these variables are not necessarily easy to calculate. Some of these properties are presented below.

Property 3.1 Let T be a random variable with a Phase-type distribution. Cumulative distribution function and first two moments of T are:

$$\begin{aligned} F(t) &= 1 - \alpha_* e^{tG_*} 1 \qquad\qquad t \geq 0 \\ E[T] &= -\alpha_* G_*^{-1} 1 \\ E[T^2] &= 2\alpha_* G_*^{-2} 1 \end{aligned}$$

where G_*^{-2} indicates the square of the inverse of G and $\mathbf{1}$ is a vector of ones.

property 3.1 can be used to determine the mean and variance. Also, the third moment is useful as a measure of skewness. For Phase-type distributions, the nth moment is given as:

$$E[T^n] = (-1)^n n! \alpha_* G_*^{-n} \mathbf{1}$$

4.8 The $M/M/3$ Preemptive Queue System with Two Priorities

This section is devoted to explaining **Model B** and using QBD to analyze it. Assume that the customers entering a system can be clustered in two groups. Harchol and Osogami [79] used Dimensionality Reduction (DR) to solve an $M/M/2$ queueing system; in this research, DR is employed to solve an $M/M/3$ queue. Customers who should receive their services as soon as possible are known as high-priority customers. The other group consists of customers who receive their services if there is not any high-priority customer in the system. It means if the system is empty of

high-priority customers and there is an idle server, then low-priority customers can be served until a high-priority customer enters the system.

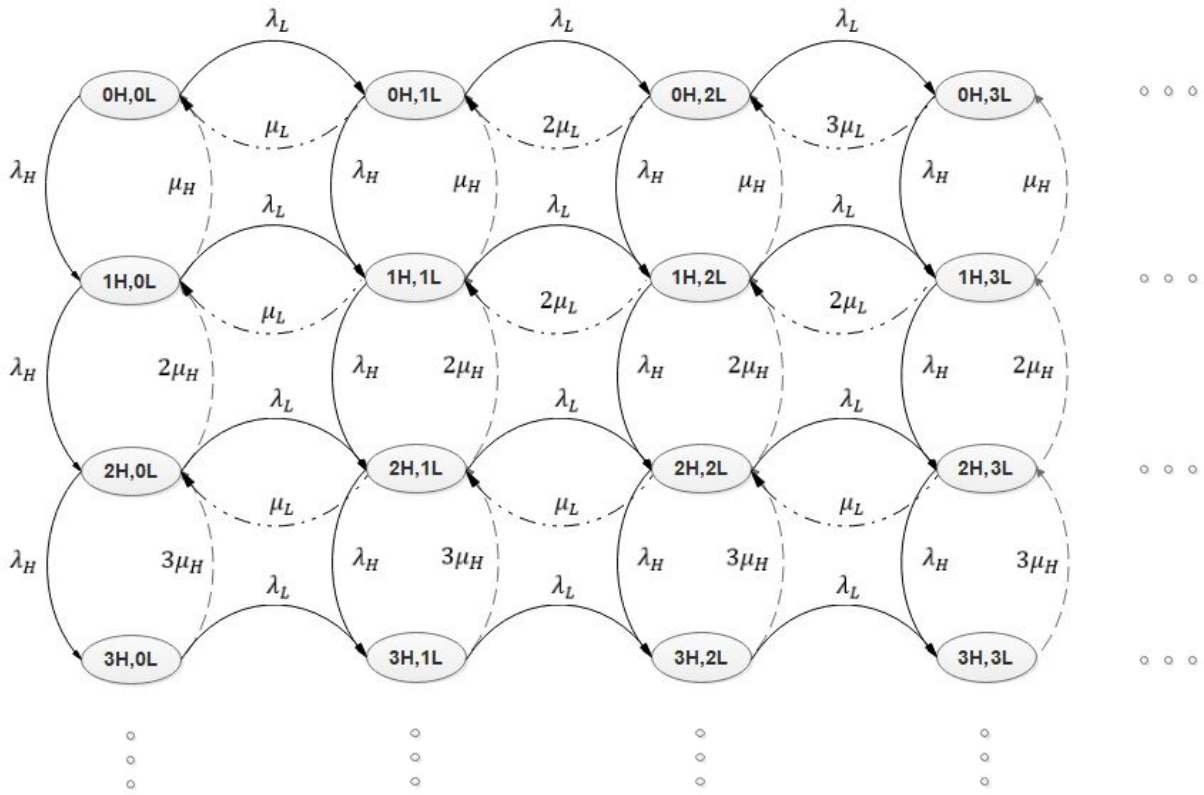

Figure 4.3: Model B - state-space for a two-priority queue system

Figure 4.3 is a combination of two birth and death processes, one for high-priority customers and the other for low-priority customers. Whenever there are high-priority customers in the system, they are served before the low-priority customers. Thus, the state-space for high-priority customers can be shown as figure 4.4. In other words, serving high-priority customers is independent from the number of low-priority customers in the system. On the contrary, the state-space of low-priority customers is dependent on the number of high-priority customers.

The state-space for the high-priority group is shown in figure 4.4. This figure depicts that high-priority customers experience an $M/M/3$ queue.

For low-priority group service rate is dependent on the number of high-priority of customers in

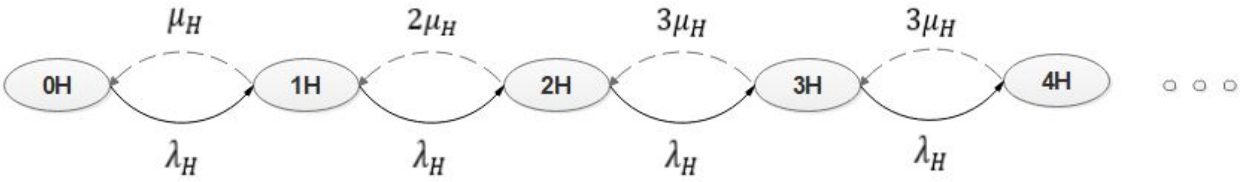

Figure 4.4: Markov chain for high-priority customers

the system. Hence, the state-space can be divided into four types:

1. The system is empty of high-priority customers. In this case, according to figure 4.5, an $M/M/3$ queue system exists for the low-priority customers.

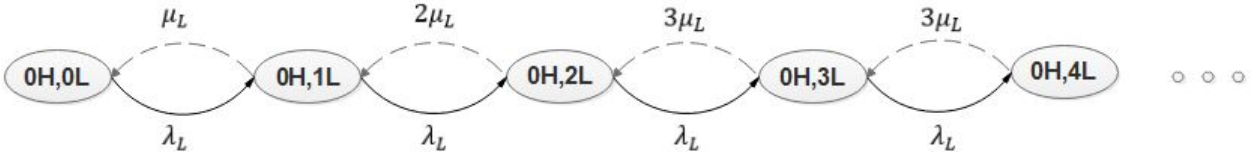

Figure 4.5: Markov chain for low-priority customers in the absence of high-priority customers

2. The system contains *one* high-priority customer. In this case, because one of the servers is busy serving the high-priority customer, low-priority customers can see the idle servers. Thus, according to figure 4.6, the rate of service with more than two low-priority customers present in the system will be $2\mu_L$.

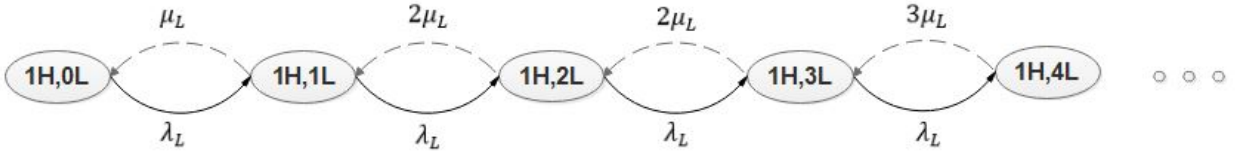

Figure 4.6: Markov chain for low-priority customers in the presence of *one* high-priority customer in the system

3. When there are *two* high-priority customers in the system, low-priority customers have an

$M/M/1$ queue system. In the other words, as figure 4.7 highlights, low-priority customers are served by the one idle server.

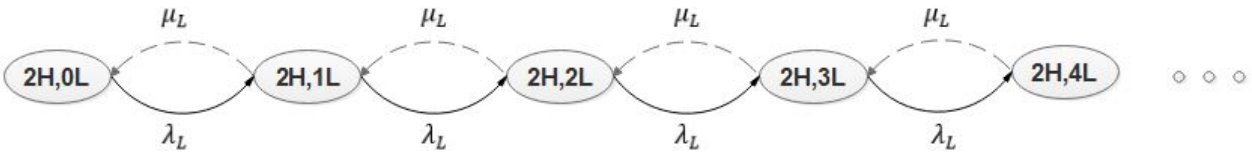

Figure 4.7: Markov chain for low-priority customers in the presence of two high-priority customers

4. When there are at least three high-priority customers in the system, and since high-priority customers are served before low-priority customers, all servers will be busy serving the high-priority group. Therefore, as figure 4.8 illustrates, the number of waiting low-priority customers will increase based on their arrival rate.

Figure 4.8: Markov chain of low-priority customers in the presence of at least three high-priority customers

To summarize, QBDs are capable of modeling many applications in the real-world; although, sometimes they can not be implemented. QBD processes can be solved via matrix-geometric method, as mentioned in section 4.4. The transition matrix usually has a structure presented in figure 4.2. However, it is possible that the transition matrix for problems presented in figure 4.3 cannot be written as a QBD, mainly because they are 2D infinite. Thus, a technique that has been introduced by Osogami [79] is used to analyze such problems.

4.9 Moment Matching Algorithm

In many fields such as engineering [80, 81], physics [82, 83] and Earth sciences [84, 85], one requires to estimate the Probability Distribution Function (PDF) of a phenomenon. With a define PDF, one can obtain several statistical properties. PDF fitting is the process of finding the parameters of stochastic variables. Various parametric methods exist for PDF fitting; examples are Maximum Likelihood Estimation (MLE) and the method of moments [86]. In case there is no prior knowledge about the distribution, methods such as kernel density estimation [87] or B-splines [88] can be used.

To describe the Moment Matching Algorithm (MMA), consider two distribution functions P and G. If at least the first three moments of these distributions are the same, then these two variables are approximately interchangeable. This algorithm has been developed by Pafnuty Chebyshev in 1887. Through moment matching algorithm a general probability distribution can be mapped to a combination of exponential distributions known as Phase-type (PH) distribution introduced by Neuts [69].

If the first moment of G, and P match, then the corresponding random variables have the same mean. Similarly, if the second moment for P and G match, the two random variables have the same variance. As a result, matching more moments leads to a better approximation of G via P. The most important reason for using a Phase-type (PH) distribution is that a large portion of Markov chain can be properly fitted by this distribution [89].

4.10 Dimensionality Reduction of Markov Chains

As mentioned before, Markov modeling has been employed to analyze the performance of real-world problems such as computer systems [90, 91], service facility [92, 93], manufacturing systems [94] and transportation networks [95]. The performance of a system, for example the mean response time of a queue, can be analyzed by modeling the system as a Markov chain and calculat-

ing the stationary probabilities. Although these Markov chains can be helpful in many situations, analyzing their behavior when the state-space grows in multiple dimensions is challenging.

In this section, the dimensionality reduction with the help of MMA introduced by Osogami [15] will be elaborated.

4.10.1 Complete Closed-Form Algorithm for Dimensionality Reduction

Osogami and Balter [15] defined the subset, S, of PH distribution called EC (Erlang-Coxian) distribution. The EC distribution, regardless of the number of phases, has only six free parameters shown in figure 4.9; this allows for deriving a closed-form solution for these parameters in terms of the moments of the input distribution.

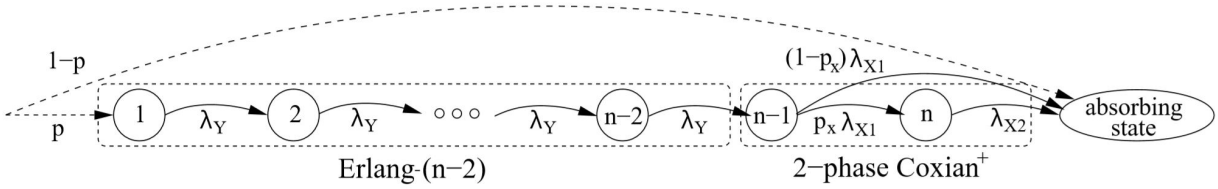

Figure 4.9: A one-dimensional Markov chain with dimension reduction

DR reduces (approximates) a multi-dimensional hard-to-analyze Markov chain to an easy-to-analyze one-dimensional Markov chain. The 1D Markov chain will be an acceptable approximation for the multi-dimensional Markov chain.

$(n, p, \lambda_{X1}, \lambda_{X2}, p_X) = \text{Complete}(\mu_1^G, \mu_2^G, \mu_3^G)$

Inputs: the first three moments of a distribution G

Output: parameters of the EC distribution

1. $m_2^G = \dfrac{\mu_2^G}{(\mu_1^G)^2}$ $\qquad$ $m_3^G = \dfrac{\mu_3^G}{\mu_1^G \mu_2^G}$

2. $p = \begin{cases} \frac{(m_2^G)^2+2m_2^G-1}{2(m_2^G)^2} & \text{if} \quad m_3^G < 2m_2^G - 1, \\ \frac{1}{2m_2^G - m_3^G} & \text{if} \quad m_3^G < 2m_2^G - 1 \\ 1 & \text{otherwise} \end{cases}$

3. $\mu_1^W = \dfrac{mu^G}{p}, \qquad m_2^W = pm_2^G, \qquad m_3^W = pm_3^G$

4. $n = \begin{cases} \lceil \frac{m_2^W}{m_2^W - 1} \rceil - 1, & \text{if} \quad m_3^W = 2m_2^W - 1 \quad \text{and} \quad m_2^W \le 2 \\ \lfloor \frac{m_2^W}{m_2^W - 1} + 1 \rfloor & \text{otherwise} \end{cases}$

5. $m_2^X = \dfrac{(n-3)m_2^W - (n-2)}{(n-2)m_2^W - (n-1)}, \qquad \mu_1^X = \dfrac{\mu_1^W}{(n-2)m_2^X - (n-3)}$

6. $\alpha = (n-2)(m_2^X - 1)(n(n-1)(m_2^X)^2 - n(2n-5)m_2^X + (n-1)(n-3))$

7. $\beta = \Big((n-1)m_2^X - (n-2)\Big)\Big((n-2)m_2^X - (n-3)\Big)^2$

8. $m_3^X = \dfrac{\beta m_3^W - \alpha}{m_2^X}$

9. $u = \begin{cases} 1 & \text{if} \quad 3m_2^X = 2m_3^X \\ \frac{6-2m_3^X}{3m_2^X - 2m_3^X} & \text{otherwise} \end{cases}$

10. $v = \begin{cases} 0 & \text{if} \quad 3m_2^X = 2m_3^X \\ \frac{12-6m_2^X}{m_2^X(3m_2^X - 2m_3^X)} & \text{otherwise} \end{cases}$

11. $\lambda_{X1} = \dfrac{u + \sqrt{u^2 - 4v}}{2\mu_1^X}, \quad \lambda_{X2} = \dfrac{u - \sqrt{u^2 - 4v}}{2\mu_1^X}, \quad P_X = \dfrac{\lambda_{X2}(\lambda_{X1}\mu_1^X - 1}{\lambda_{X1}}, \quad \lambda_Y = \dfrac{1}{(m_2^X - 1)\mu_1^X}$

4.11 Using DR to Analyze the $M/M/3$ Queue with Two Priorities

Since the waiting time of the high-priority customers is not impacted by the low-priority group, the high-priority group can be studied independently. To review the queue for high-priority customers, a birth and death process as an $M/M/3$ queue model, as presented in figure 4.4, should be analyzed.

On the other hand, low-priority customers are impacted by the presence and number of high-priority customers in the system. Therefore, analyzing the low-priority group is more complicated. Whenever there is no high-priority customer in the system, low-priority customers have an $M/M/3$ model because all of the servers are available for them 4.5. If there is one customer from the high-priority group in the system, the low-priority customers will face an $M/M/2$ queue system (one of servers is busy with the high-priority customer); check figure 4.6. The third case is an $M/M/1$ queue in which there are two high-priority customers in the system. Finally, in case there are three or more than three high-priority customers in the system, no low-priority customer will be served. Thus, the number of low-priority customers in the system will increase until the number of high-priority customers decreases to two or less; check figure 4.8.

To analyze the performance of the system in dealing with low-priority customers, it is necessary to convert the resulting multi-dimensional Markov chains to one-dimensional Markov chains. In this section, the DR technique introduced by Osogami [15] is employed to convert 2D infinite Markov chains to 1D infinite chains, as shown in figure 4.10.

As is described in section 4.7, PH distribution can be used to model Markov chains. In this step, the parameter $\mu_{\geq 3H}$ is substituted by a PH distribution. Thus, the chain with infinite dimensions for high-priority customers is converted to the 1D Markov chain of figure 4.11. The values for β_{12}, β_1 and β_2 can be computed using the following equations from the DR algorithm based on complete-

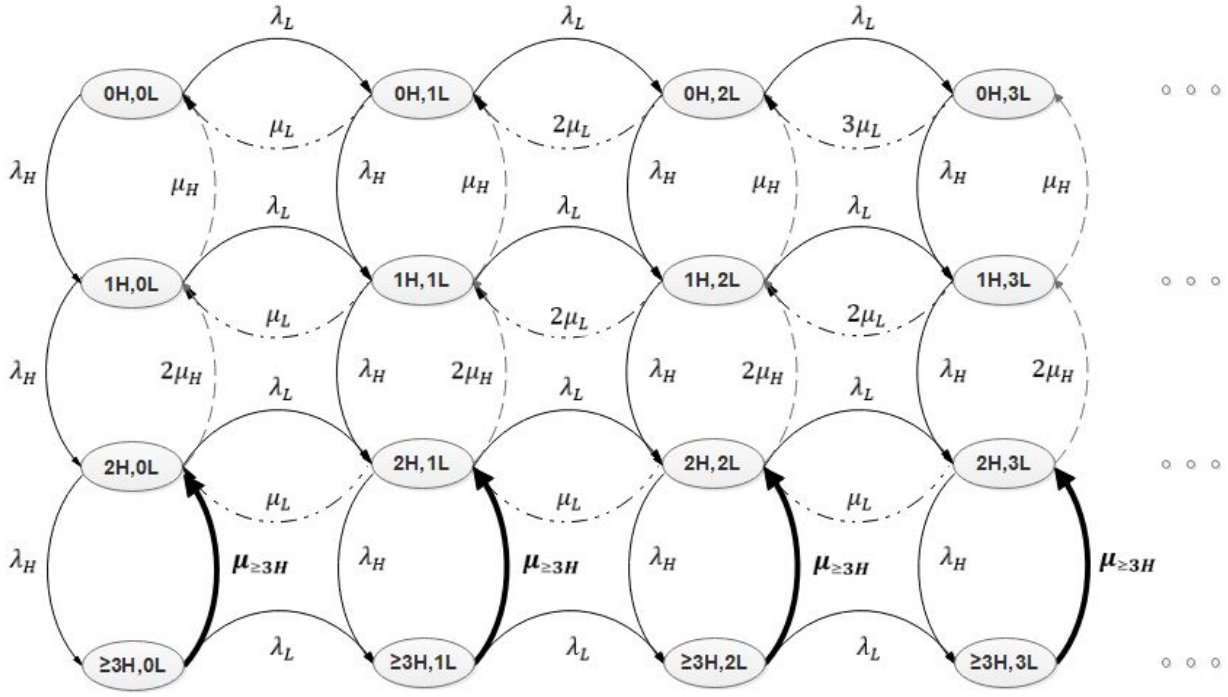

Figure 4.10: Converting a 2D Markov chain to a 1D chain using DR approach

closed-form algorithm presented in section 4.10.1:

$$\beta_{12} = p_X \lambda_{X_1}; \qquad \beta_1 = (1 - p_X)\lambda_{X_1}; \qquad \beta_2 = \lambda_{X_2} \tag{4.30}$$

In the dimension reduction process shown in figure 4.10, all the states in levels $\geq 3H$ can be aggregated to two states labeled A and B; this is illustrated by figure 4.11. If T denotes the sojourn time for busy states of $\geq 3H$, T can be approximated by a two-Phase-type (PH) distribution. To do this, the first three moments of T are required. The sojourn time T is exactly the same as the busy period in an $M/M/1$ queue, where the arrival rate is λ, the service rate is μ, and $\rho = \frac{\lambda}{n\mu}$. The moments for busy period of $M/M/1$ is presented by [96]:

$$E(T) = \frac{1}{(1-\rho)\mu}; \qquad E(T^2) = \frac{2}{(1-\rho)^3\mu^2}; \qquad E(T^3) = \frac{6(1+\rho)}{(1-\rho)^5\mu^3} \tag{4.31}$$

After implementing PH distribution approximation, 2D infinite Markov chain is converted to

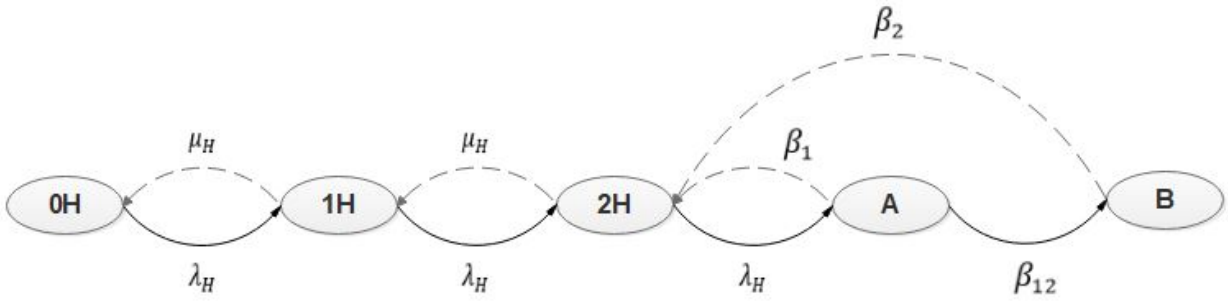

Figure 4.11: 1D Markov chain with dimension reduction

a 1D infinite Markov chain that can be analyzed via QBDs process. To analyze the performance criteria for low-priority customers, a PH distribution approximation has been used to compute three parameters of β_1, β_2 and β_{12}.

Once the dimension reduction procedure is finished, figure 4.12 is created. This 1D infinite Markov chain can be modeled as a QBD process, where level l of the process denotes the l-th column, resulting in states of the form (iH, lL) for each l. Thus, according to [89], Q, the generator matrix for this process, can be presented as a block diagonal matrix:

$$Q = \begin{bmatrix} L^{(0)} & F^{(0)} & & & \\ B^{(1)} & L^{(1)} & F^{(1)} & & \\ & B^{(2)} & L^{(2)} & F^{(2)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

Here, sub-matrix $F^{(l)}$ shows the forward transition from level or column l to the next level $(l+1)$ for $l \geq 0$. Also, sub-matrix $B^{(l)}$ indicates backward transition rates from level l to level $l-1$ for $l \geq 1$. Sub-matrix $L^{(l)}$ will be used to show the rates within level l for $l \geq 0$.

Hence, resulting in sub-matrix $L^{(l)}$:

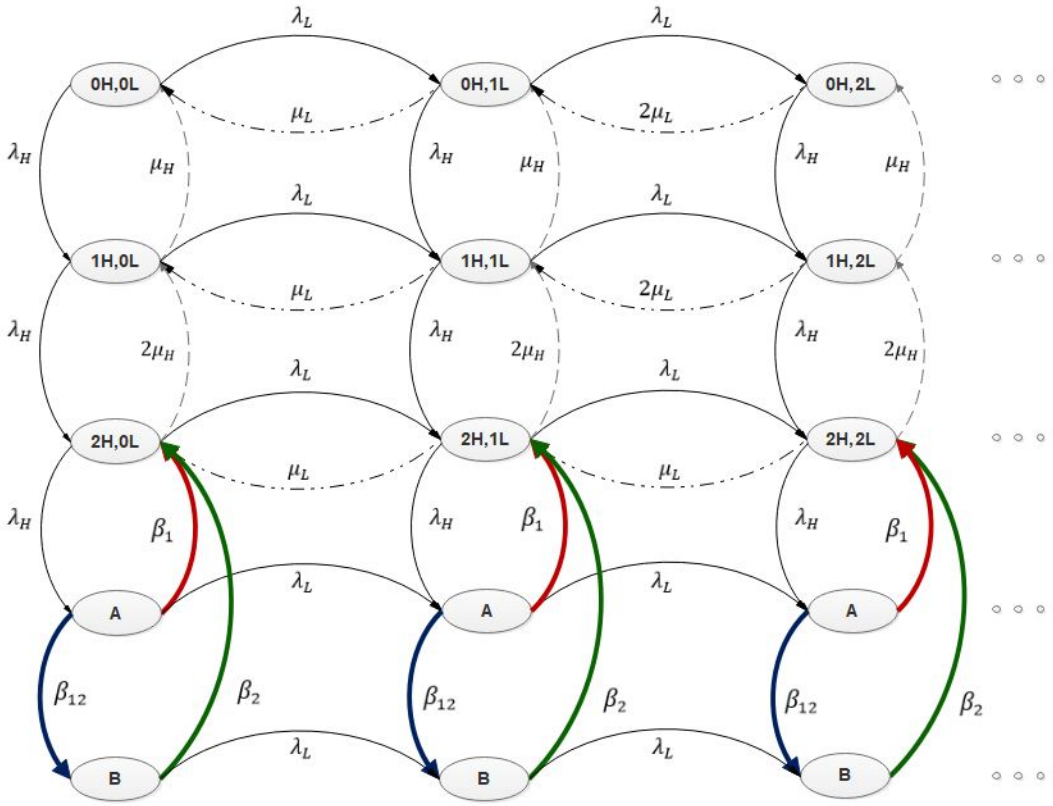

Figure 4.12: 1D infinite QBD for two priority customers with DR approximation

$$L^{(0)} = \begin{bmatrix} b_1^* & \lambda_H & & & \\ \mu_H & b_2^* & \lambda_H & & \\ & 2\mu_H & b_3^* & \lambda_H & \\ & & \beta_1 & b_4^* & \beta_{12} \\ & & \beta_2 & & b_5^* \end{bmatrix}$$

in which $b_i^* = \sum_{j \neq i}^{\infty} Q_{ij}$. Also, it should be mentioned that since $\widehat{l} = 3$ ($\widehat{l}$ is the level after which the QBD sub-matrices show a repeating structure), b_i^* will have different values for every sub-matrices $l < \widehat{l}$. The other sub-matrices will be computed by the following formula:

$$F^{(l)} = \lambda_L \mathbf{I}$$

$$F^{(0)} = .. = F^{(2)} = F^{(3)} = F = \begin{bmatrix} \lambda_L & & & & & \\ & \lambda_L & & & & \\ & & \lambda_L & & & \\ & & & \lambda_L & & \\ & & & & \lambda_L & \end{bmatrix}$$

$$B^{(1)} = \begin{bmatrix} \mu_L & & & & & \\ & \mu_L & & & & \\ & & \mu_L & & & \\ & & & 0 & & \\ & & & & 0 & \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 2\mu_L & & & & & \\ & 2\mu_L & & & & \\ & & \mu_L & & & \\ & & & 0 & & \\ & & & & 0 & \end{bmatrix}$$

$$B^{(3)} = \begin{bmatrix} 3\mu_L & & & & & \\ & 2\mu_L & & & & \\ & & \mu_L & & & \\ & & & 0 & & \\ & & & & 0 & \end{bmatrix}$$

4.12 Quadratically Convergent Algorithms

If the QBD process has an infinite number of levels, the state-space of QBD needs to be truncated.

$$Q = \begin{bmatrix} L^{(0)} & F^{(0)} & & & \\ B^{(1)} & L^{(1)} & F^{(1)} & & \\ & B^{(2)} & L^{(2)} & F^{(2)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix}$$

When the QBD process repeats after a certain level, $\widehat{l}$, (i.e. $F^{(l)} = F$, $L^{(l)} = L$, $B^{(l)} = B$ for all $l \geq \widehat{l}$), $\mathbf{R}^{(l)}$ is the same for all $l \geq \widehat{l}$, and $\mathbf{R} = \mathbf{R}^{(l)}$ for $l \geq \widehat{l}$. Once $\mathbf{R}^{(\widehat{l}+1)} = \mathbf{R}$ is obtained, $\mathbf{R}^{(l)}$ for $1 \leq l \leq \widehat{l}$ is given recursively via equation 4.32. Various approaches for calculating $\mathbf{R}$ have been proposed, and in figure 4.13 an algorithm from the literature is presented [67, 89]:

$$F^{(l-1)} + \mathbf{R}^{(l)} L^{(l)} + \mathbf{R}^{(l)} \mathbf{R}^{(l+1)} B^{(l+1)} = 0 \tag{4.32}$$

Once $\mathbf{R}$ and $\mathbf{R}^{(l)}$s are obtained, the stationary probability vector $\vec{\pi_l}$ can be calculated recursively; $\vec{\pi_l} = \vec{\pi_{l-1}} \mathbf{R}^{(l)}$ for $l \geq 1$. Thus, all that remains is to calculate $\vec{\pi_0}$. Vector $\vec{\pi_0}$ is given by the positive solution of:

$$\vec{\pi_0}(L_0 + \mathbf{R}^{(1)} B^{(1)}) = \vec{0}. \tag{4.33}$$

For this problem, an $M/M/3$ preemptive system with two priorities, the transition matrix will repeat after level 3. Therefore, $\widehat{l} = 3$.

4.13 Numerical Results

In this section, a Monte Carlo simulation has been developed to compare the introduced algorithm with the simulation results. Assume an $M/M/3$ queueing system with two priorities and investigate the effect of ρ (utilization factor or the portion of time the service station is busy and cannot serve other customers) on the waiting time of each priority.

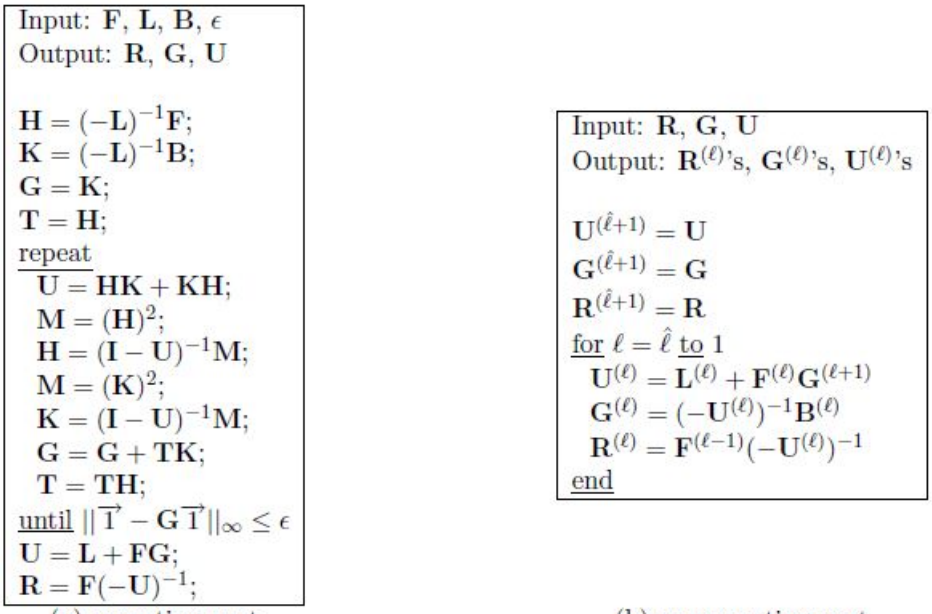

Figure 4.13: Algorithms for calculating $\mathbf{R}$ and $\mathbf{R}^{(l)}$

Figure 4.14 shows the performance of simulation compared to the queueing theory results for high-priority customers based on the utilization factor. As mentioned, high-priority customers have an $M/M/3$ queue model; therefore, their average wait time can be calculated by equation (3.16). The results show that the performance of the developed simulation method is almost equal to the queueing models.

Figure 4.15 compares the results of simulation method and dimensionality reduction for the second priority group. Since both DR and simulation are approximate methods, there is a small gap of about 7% between their results. In both figures, waiting times grow exponentially when ρ increases. In other words, busier systems result in a considerable growth of waiting times for customers, especially for the second priority group.

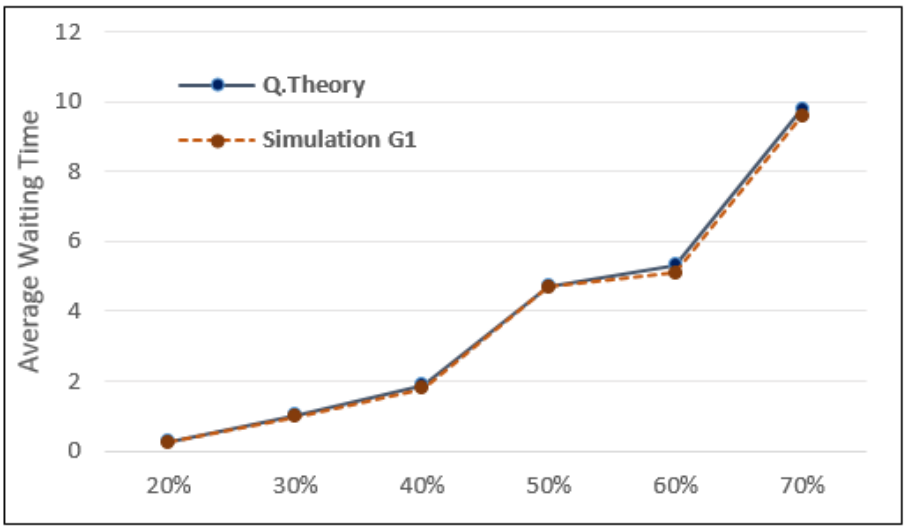

Figure 4.14: The effect of increasing ρ on the average waiting time of the first group

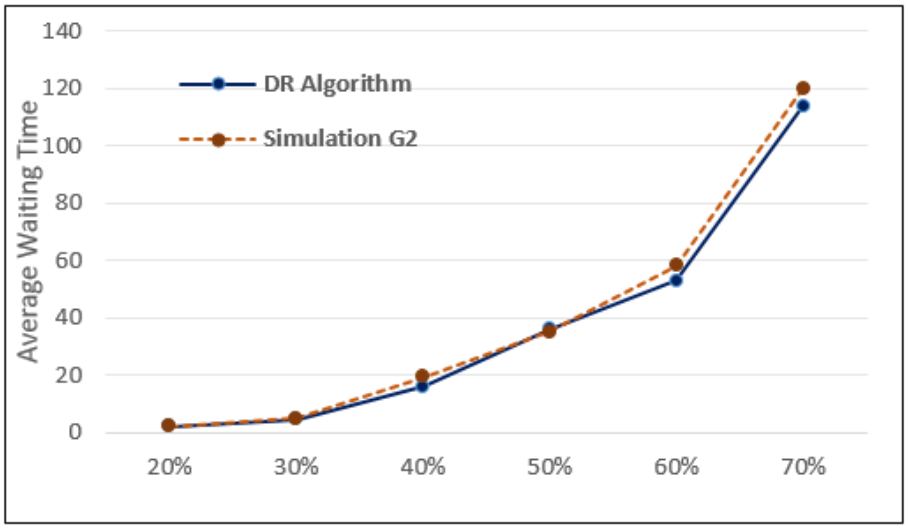

Figure 4.15: The effect of increasing ρ on the average waiting time of the second group

5. Customer Segmentation with Machine Learning Techniques

5.1 The Importance of Data

Today, there is considerable amounts of data generated and stored in any organization. Companies try to exploit the data for a competitive advantage. Whereas in the past companies tended to use teams of statisticians and analysts to work with data, nowadays with powerful computers, a high volume of data can be analyzed in a fraction of time. Therefore, the application of data has increased drastically compared to the past [97].

Technology allows researchers to analyze vast volumes of data and develop strategies. Data analysis is now critical for business strategy in every organization. Businesses try to use data in their strategic decisions, which impact their profits. These strategic decisions have an advantage for they consider all the available data. Therefore, the capability of using data to extract knowledge is considered as a crucial strategic asset; and it is evident that firms improve their ability to approach problems using data analytic methods. The science of extracting insight from raw data is called data science.

As Dhar [98] described in his research "data science is an interdisciplinary field using diverse statistics methods and algorithms to extract knowledge from data in various forms, both structured and unstructured, similar to data mining." Data science includes three phases, design for data, collection of data, and analysis of data [99].

5.2 Machine Learning

Samuel [100] defined Machine Learning (ML) as "the field of study that gives computers the ability to learn without being explicitly programmed." Every learning model needs a dataset to extract insight; the data is the fuel for the learning problems. A dataset contains two essential di-

mensions: features and number of data points. Usually, the dataset has a tabular structure such as customer specifications in banks or patients in hospitals; even voices and pictures that are unstructured data.

A dataset with tabular format was selected for this research. The selected dataset contains many data points, each of which has a specific attribute called a feature. The number of features shows the dimension of the dataset. The quality of a machine learning model depends on such features. If the number of features is not enough, the model will not be able to perform appropriately; on the other hand, if the number of features are too many, the training of the model will be time-consuming and costly. Feature engineering is a branch of machine learning that optimizes the features.

5.3 Principal Component Analysis

Consider a dataset with two or three features; evidently, analyzing and visualizing this dataset is more straightforward that a dataset with several features. As the number of features increases, data analysts may experience the curse of dimensionality, a phrase coined by Richard E. Bellmen [101], referring to the different problems one may experience when analyzing data with high-dimensionality. As the number of dimensions of a dataset increases, the number of samples required for an estimator to generalize increases exponentially. Furthermore, high-dimensionality leads to sparseness of data. Thus, it can become more difficult to detect similar instances in high-dimensional spaces for all of the instances are similarly sparse [102].

In this section, Principal Component Analysis (PCA) is presented as a statistical and linear transformation technique widely used for dimensionality reduction. PCA reduces the dimension of the data without considerable loss of information. Besides, PCA can be beneficial in exploratory data analyses and removing noise. PCA facilitates patterns recognition in data based on the correlation between the features [103]. PCA examines the correlations among the original inputs and uses this information to construct the appropriate composite measures, named principal components [104].

PCA looks for maximum variance in high-dimensional data and projects all the data to a new sub-space with fewer or at least equal number of dimensions (k) than the original dataset's (d). These principal components are orthogonal axes with a specific pattern for variance. The first axis, representing the first principal component, has the highest variance; the second axis has the second highest variance, and so on. Therefore, assuming that the original data has d dimensions (a dataset with d features), then a matrix $W_{d\times k}$ can be established to map a sample vector x to the new sub-space with k dimensions ($k \ll d$).

$$\begin{aligned} x &= [x_1, x_2, ..., x_d] & x \in R^d \\ z &= xW & W \in R^{d\times k} \\ z &= [z_1, z_2, ..., z_k] & z \in R^k \end{aligned}$$

Using PCA for dimentionality reduction consists of the steps described in section 5.3.1.

5.3.1 Standardizing the Dimensions of Data

Features are categorized in two groups:

1. *Categorical feature* is a data attribute which belongs to a finite set of possible categories or classes; for example, sex, countries, and weather.

2. *Numerical feature* is a data feature which can be explained by either integer or float numbers; examples include age, weight, and credit score.

A dataset may contain one or multiple numerical features. Assume a dataset with two differently scaled numerical features: weight and income. These different scales in PCA usually cause a problem because the method searches for the most amount of dispersion between the features. In fact, models that are smooth functions of the inputs are impacted by the scale of the inputs [105]. In other words, PCA requires feature scaling, i.e., conversion of the scale of all the features to a scale such that the features are comparable with each other. One common feature scaling approach

is standardization. Feature standardization is defined as:

$$\hat{x} = \frac{x - \mu_x}{\sqrt{var(x)}} \tag{5.1}$$

Standardization converts a feature to a standard value with $\mu = 0$ and $\sigma = 1$.

5.3.2 Constructing the Covariance Matrix

Covariance is defined between two variables and explains the degree of two variables changing together. In other words, covariance is a measure of the joint variability of two random variables and describes the tendency of linear relationship between these two variables. A positive covariance between two features indicates that the features increase or decrease together, whereas a negative covariance indicates that the features move in opposite directions. The covariance between two variables X and Y can be calculated in the following way:

$$\sigma_{X,Y} = \ cov(X, Y) = \ \sum_{i=1}^{n} \frac{(X_i - \mu_X)(Y_i - \mu_Y)}{n} \tag{5.2}$$

Therefore, the covariance matrix for a dataset with d dimensions is:

$$\sum = \begin{bmatrix} \sigma_{x_1,x_1} & \sigma_{x_1,x_2} & \dots & \sigma_{x_1,x_d} \\ \sigma_{x_2,x_1} & \sigma_{x_2,x_2} & \dots & \sigma_{x_2,x_d} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{x_d,x_1} & \sigma_{x_d,x_2} & \dots & \sigma_{x_d,x_d} \end{bmatrix}$$

5.3.3 Computing Eigenvectors and Eigenvalues

A vector is described by its direction and magnitude. An eigenvector of a matrix is a non-zero vector that satisfies equation (5.3). The eigenvectors of the covariance matrix show the direction of

maximum variance. Eigenvalues show the magnitude of these directions.

$$\sum \vec{v} = \lambda \vec{v}$$
$$(\sum -\lambda \mathbf{I})\vec{v} = 0 \tag{5.3}$$

In this equation, $\vec{v}$ is an eigenvector, $\sum$ is the covariance matrix of the dataset, and λ is a scalar called eigenvalue.

Dimensionality reduction in a dataset means compressing the data to a new feature sub-space while preserving the most volume of information possible. The k largest eigenvalues represent the k principal components. Equation (5.4) is used to show how much variance is covered by each principal component. The ratio of total variance covered by an eigenvalue λ_j is computed by dividing λ_j by the sum of all eigenvalues:

$$\frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j} \tag{5.4}$$

After selecting k eigenvalues, the corresponding eigenvectors are selected to make a new matrix. The new matrix, which is called the *projection matrix*, is used to transform data to the new sub-space.

5.3.4 Transforming the Data

To summarize, PCA is a technique for dimensionality reduction of a dataset feature space. PCA maps a high-dimensional space to a smaller sub-space while keeping a large portion of information. In the new sub-space, the number of features is usually less than the original space.

The first principal component reflects the largest variance in the dataset. The second principal component is orthogonal to the first principal component and reflects the second largest variance, and so on. There are as many principal components as the number of original variables. This reduction facilitates considering some of the available data for visualization or prediction. Figure

5.1 illustrates implementation of PCA to a three-dimensional dataset.

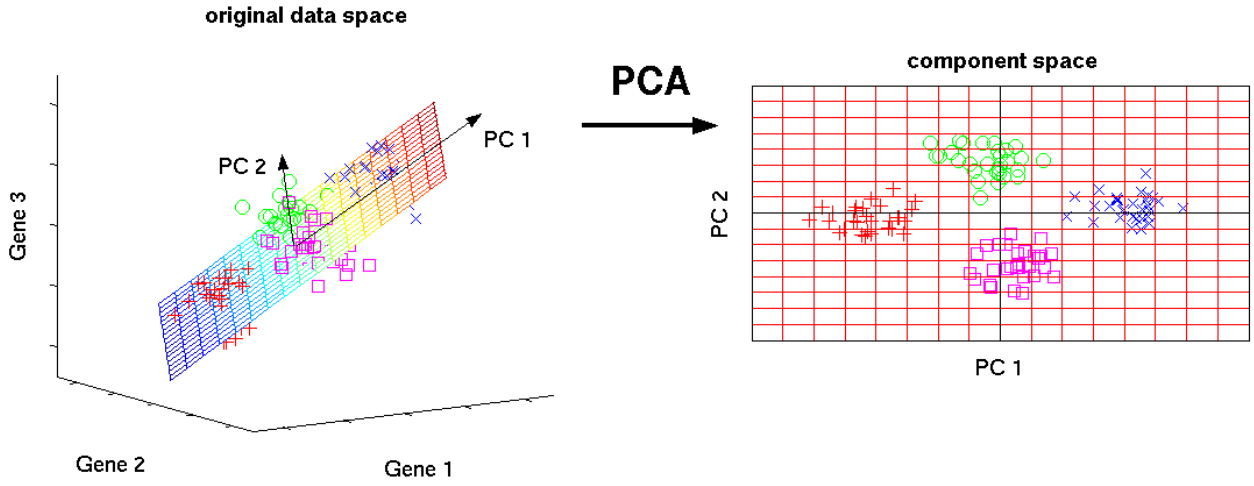

Figure 5.1: Dimentionality reduction with PCA [4]

5.4 Clustering Techniques

Cluster analysis divides a given collection of unlabelled data objects into groups that are meaningful based on a specific similarity measure. Clustering ensures that data objects within a group have significant similarity to one another and enormous difference from the data objects in other groups. After producing several well-separated clusters in the feature space, cluster analysis can adequately summarize and visualize the data in a given collection. Clustering techniques have often been applied in real-life problems for customer segmentation, document clustering, and clustering of gene expression data. All of the various clustering techniques developed in recent decades can be classified into two categories: *partitional clustering* and *hierarchical clustering*.

Partitional clustering, which simply divides data objects into clusters, is the most popular approach. In this method, each data object belongs to exactly one cluster. On the other hand, hierarchical clustering allows clusters to have their own sub-clusters that form a hierarchy (i.e. a tree) of clusters. In this tree, each leaf node denotes a data object (i.e. the smallest cluster), and the root node denotes the cluster containing all the data objects (i.e. the largest cluster). Each inter-

nal node in the tree is a cluster consisting of all data objects in its child nodes (i.e. the union of its sub-clusters). Defining a cutting point at a specific level makes it possible to retrieve a set of non-overlapping clusters.

This study adopts two well-known partitional clustering techniques, *K-Means* and Gaussian mixture model, which will be described in the following sections.

5.5 K-Means Method

The goal of unsupervised learning is detecting hidden patterns in data. The K-Means clustering algorithm is the most well-known clustering method [106]. K-Means is an iterative method of moving the centroids of clusters to the mean position of their constituent points; and re-assigning the data points to the new clusters [102].

K-Means needs a parameter k which defines the number of clusters. k must be positive and less than the number of data points. Since in many problems information about the number of clusters does not exist, and therefore, the optimum number of clusters cannot be determined, a technique known as "elbow method" is employed.

The parameters of K-Means are the centroids of clusters and the data points assigned to each cluster. To find the optimum values of K-Means parameters, a cost function is used. The cost function for K-Means is given by equation (5.5).

$$M = \sum_{k=1}^{K} \sum_{j=1}^{N_k} (x_j - \mu_k)^2 \tag{5.5}$$

x_j represents each data point in the dataset. Also, N_k indicates the number of data points in k^{th} cluster. The goal is clustering the data points based on their feature similarities. The main steps for K-Means can be summarized as follows [103]:

1. Randomly pick K centroids from the data or use the centroids calculated by another algorithm.

2. Assign each data point to the nearest centroid. To do this, calculate the distance between each sample and the centroids and select the minimum distance to form a cluster. Equation (5.6) calculates the squared Euclidean distance between data point $x(x_1, x_2, ..., x_m)$ and μ_k, the centriod of the kth cluster.

$$d(x, y) = \sum_{j=1}^{m} (x_j - \mu_k)^2 \tag{5.6}$$

3. Calculate the centroids of the new samples for each of the K clusters.

$$\mu_k = \frac{\sum_{n=1}^{N_k} x_n}{N_k} \qquad k \in \{1, 2, ..., K\} \tag{5.7}$$

4. Calculate the cost function.

$$M = \sum_{k=1}^{K} \sum_{n=1}^{N_k} (x_j - \mu_k)^2 \tag{5.8}$$

5. Repeat steps 2 to 4 for a predetermined number of iterations or until a user-defined threshold is reached.

5.6 Gaussian Mixture Models

A probabilistic model that can detect the presence of sub-populations in a population is known as a **Mixture model**. A Gaussian Mixture Model (GMM) is a function comprised of several Gaussians. Each Gaussian in the mixture have the following parameters:

1. A mean μ that defines its center.

2. A covariance Σ that defines its width. It determines the dimension of an ellipsoid in a multivariate scenario.

3. A mixing probability π that defines how big or small the Gaussian will be.

5.6.1 Gaussian Density Function

The Gaussian density function is described by the following equation:

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|} exp^{(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))} \tag{5.9}$$

in which x represents a D-dimensional data point. For example, for a dataset with 2000 data points, each with 3 features, x is a 2000×3 matrix; μ is a 1×3 vector, and Σ is a 3×3 matrix.

An example of Gaussian mixture is presented in figure 5.2

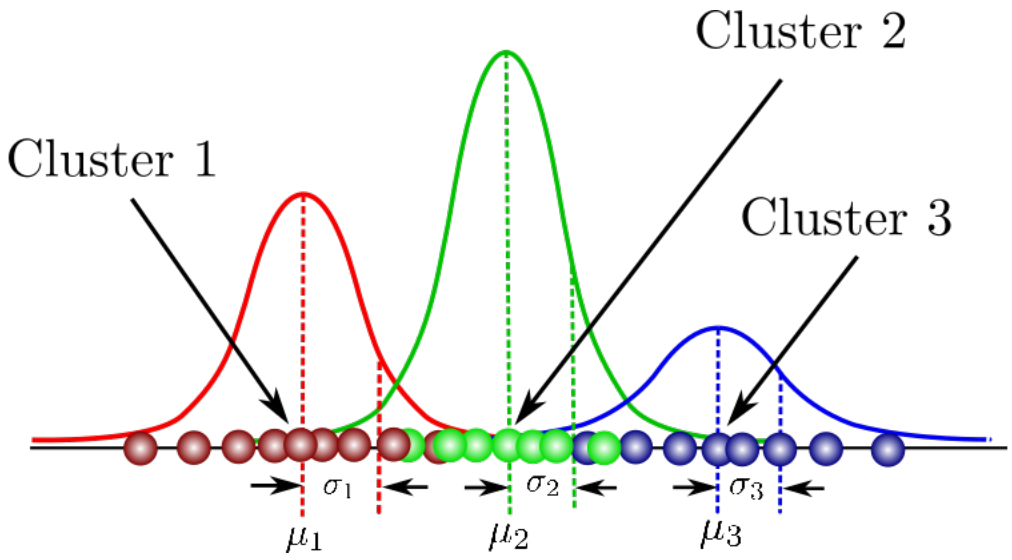

Figure 5.2: Gausian mixture [5]

For example, assume patients in a hospital in which the wait time for patients with more than 60 years of age is a normal distribution with a different mean and variance compared to the wait time for patients who are between 20 and 30 years old. Thus, in this population, two normal sub-populations with different mean and variance are realized. In general, a population may have many multivariate Gaussian distributions instead of having normal distributions.

The notations used in describing the algorithm of finding the parameters of K Gaussian distributions are as follows. First, the probability that a data point x_n comes from a Gaussian k is shown

by (5.10):

$$p(z_{nk} = 1|x_n) \tag{5.10}$$

Here, z is a latent variable that takes only two possible values. $z = 1$ means x is from Gaussian k; $z = 0$ assures that it does not belong to the k Gaussian. Finding the values of z is beneficial since knowing its probability of occurrence facilitates determining the Gaussian parameters.

Equation (5.11) shows the overall probability that a point comes from the Gaussian k.

$$\pi_k = p(z_k = 1) \tag{5.11}$$

$\mathbf{z}$ is defined as the set of all possible latent variables z, hence:

$$\mathbf{z} = \{z_1, ..., z_k\} \tag{5.12}$$

Therefore:

$$p(z) = p(z_1 = 1)^{z_1}\ p(z_2 = 1)^{z_2}\ ...\ p(z_k = 1)^{z_k}\ = \prod_{k=1}^{K} \pi_k^{z_k} \tag{5.13}$$

If z can be determined, the probability of observing data can be computed:

$$p(x_n|\mathbf{z}) = \prod_{k=1}^{K} N(x_n|\mu_k, \Sigma_k)_k^z \tag{5.14}$$

Use the Bayesian rule, the probability of Gaussian mixture is defined by (5.16):

$$p(x_n) = \sum_{k=1}^{K} p(x_n|z)p(z) = \sum_{k=1}^{K} \pi_k\ N(x_n|\mu_k, \Sigma_k) \tag{5.15}$$

Therefore:

$$P(\mathbf{X}) = \prod_{n=1}^{N} p(x_n) = \prod_{n=1}^{N} \pi_k\ N(x_n|\mu_k, \Sigma_k) \tag{5.16}$$

5.6.2 Expectation Maximization

Expectation Maximization (EM) algorithm is a well-known statistical model used for clustering tasks [107]. Based on the assumption that data objects are generated as a result of a statistical process, the EM algorithm aims to discover a set of statistical models (i.e. distributions) that can best fit different data clusters. Therefore, the clustering task for the EM algorithm corresponds to the problem of estimating the parameters of a distribution. EM can be summarized in two main steps:

1. **The expectation step** relocates data objects into different distributions to form data clusters; and,

2. **the maximization step** re-estimates the parameter values of the distributions.

The steps of EM correspond to the K-Means steps, where data objects are assigned to the closest centroid and new cluster centroids are recalculated, respectively. The iterations continue until there is no further change when estimating the parameter values [54].

Here EM algorithm is described in details for predicting the parameters of a model based on the data points. Consider θ as the set of parameters:

$$\theta = \{\pi, \mu, \Sigma\}$$

The following steps describe the EM procedure [5]:

1. **Initialization**: the algorithm starts with initial values for the parameters. The initial values can be randomly chosen or be the result of K-Means method.

2. **Expectation Step**: Evaluate:

$$P(Z|X, \theta)$$

as it was mentioned, this can be calculated as follows:

$$p(z_k = 1|x_n) = \frac{p(x_n|z_k = 1)p(z_k = 1)}{\sum_{j=1}^{K} p(x_n|z_j = 1)p(z_j = 1)} \tag{5.17}$$

Also:

$$p(z_k = 1) = \pi_k \qquad p(x_n|z_k = 1) = N(x_n|\mu_k, \Sigma_k) \tag{5.18}$$

So, plugging (5.18) in (5.17) results in:

$$p(z_k = 1|x_n) = \frac{N(x_n|\mu_k, \Sigma_k).\pi_k}{\sum_{j=1}^{K} N(x_n|\mu_j, \Sigma_j).\pi_j} = \gamma(z_{nk}) \tag{5.19}$$

$$P(z_{nk}|X, \theta) = E[z_{nk}] = \sum_{j=1}^{K} z_{nj}.p(z_k = 1|x_n) = \sum_{j=1}^{K} z_{nj}.\gamma(z_{nj}) = \gamma(z_{nk}) \tag{5.20}$$

3. **Maximization Step**: update the parameters of θ^* using:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} Q(\theta^*, \theta) \tag{5.21}$$

where

$$Q(\theta^*, \theta) = E\left[\ln\left(p(X, Z|\theta^*)\right)\right] = \sum_{z} p(Z|X, \theta)\ \ln\left(p(X, Z|\theta^*)\right) \tag{5.22}$$

To calculate $p(X, Z|\theta^*)$, one can use the likelihood function:

$$p(X, Z|\theta^*) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}}\ N(x_n|\mu_k, \Sigma_k) \tag{5.23}$$

After taking the natural logarithm:

$$\ln\left(p(X, Z|\theta^*)\right) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk}\ \left[\ln\left(\pi_k\right) + \ln\left(N(x_n|\mu_k, \Sigma_k)\right)\right] \tag{5.24}$$

Replacing (5.20) and (5.24) in (5.22), and applying a Lagrange multiplier for coefficients of π result in:

$$Q(\theta^*, \theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) \left[\ln(\pi_k) + \ln(N(x_n|\mu_k, \Sigma_k))\right] - \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right) \tag{5.25}$$

Taking the derivative of equation (5.25) results in:

$$\frac{\partial Q(\theta^*, \theta)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} - \lambda = 0 \tag{5.26}$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) = \pi_k \lambda \Rightarrow \sum_{k=1}^{K} \sum_{n=1}^{N} \gamma(z_{nk}) = \sum_{k=1}^{K} \pi_k \lambda \tag{5.27}$$

Sum of all mixing coefficients π must equal one. In addition, adding the probabilities γ over k must also equal one. Therefore: $\lambda = N$. Hence:

$$\pi_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N} \tag{5.28}$$

Also, by taking derivative of equation (5.25) with regards to μ_k and Σ_k:

$$\mu_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{5.29}$$

$$\Sigma_k^* = \frac{\sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{n=1}^{N} \gamma(z_{nk})} \tag{5.30}$$

In other words, the described algorithm repeats steps 2 and 3 until the parameters are converged.

5.6.3 Differences Between K-Means and GMM

There are some advantages in using GMM over K-Means. For instance, K-Means does not consider the variance when clusters are defined; in GMM, variance plays a crucial role in defining the models. Also, In K-Means, the model implements a circle (a hyper-sphere in higher dimensions) at the center of each cluster; GMM uses oblong clusters that can be defined in different shapes. The third advantage of GMM over K-Means is the classification method; GMM applies a soft classification, whereas K-Means is a hard classification. In other words, GMM calculates the probability of a data point belonging to each of the clusters; K-Means assigns a point to one and only one cluster.

5.7 The Used Dataset

To implement the described methods of this research on a real case, a public dataset of bank customers [1] was selected. First, the specifications of the dataset will be described. Afterward, an Exploratory Data Analysis (EDA) will be presented. EDA is a fundamental part of all machine learning projects. Next, PCA will be applied to reduce the dimension of the dataset. Finally, the customers will be clustered based on two methods: K-Means and GMM.

5.7.1 Features List

The dataset contains features as set out by table 5.1.

5.7.2 Exploratory Data Analysis (EDA)

As mentioned before, EDA is considered as a necessary step in machine learning projects; EDA facilitates learning about the features and their characteristics. In this research, a pair plot is employed to summarize the features of the dataset.

[1] Santush Kumar. "Bank customers churn." Kaggle, July 9, 2018, https://www.kaggle.com/santoshd3/bank-customers/metadata

Table 5.1: The features of the used dataset

Feature	Type	Explanation
CustomerId	Numerical	A unique value assigned to each customer
CreditScore	Numerical	The credit score of the customer
Age	Numerical	The age of the customer
Tenure	Numerical	The number of months a customer has been an account holder
Balance	Numerical	The account balance of the customer
NumOfProducts	Numerical	Number of services that the customer has utilized in the bank
HasCrCard	Categorical	If the customer has a credit card
IsActiveMember	Categorical	If the person an active customer
EstimatedSalary	Numerical	Estimated salary of the customer
Exited	Categorical	If the customer has an account in the bank
Sex	Categorical	Sex of the customer

Figure 5.3 depicts that the credit score has a normal distribution ($\mu \approx 650$). The age feature has a positive skewness, i.e., the right tail is longer than the left tail. Also, it can be seen that a high portion of the customers are between 27-45 years old. The features tenure and estimated salary are almost uniformly distributed. Finally, the Balance feature has two sub-populations, the average for one of the sub-populations is approximately 0; the average for the other sub-population is about 125,000.

5.7.3 Applying PCA

PCA aids with using all of the features and converting them to principal components. To do so, a library called "Scikit learn" in Python was used. To employ PCA effectively, finding the number of components is crucial. In this research, the accumulative function of explained variance ratio is used.

Figure 5.4 depicts the portion of variance of the dataset explained by each component. Accordingly, if the first two principal components are used, about 90% of the variance in the dataset is covered.

5.7.4 K-Means and GMM Clustering

Before applying the K-Means and GMM algorithms, the number of clusters should be determined. To determine the number of clusters, a technique called the elbow method is used in this thesis. Elbow method is a heuristic technique used to find the number of clusters in a dataset. It plots the objective function versus the number of clusters. According to Elbow method, the best number of clusters is attained where a minimization objective function drops the most among all the candidates. Figure 5.5 illustrates that, according to Elbow method, three clusters is adequate for separating customers. After implementing K-Means and GMM on the dataset, as shown in figures 5.6 and 5.7, three groups of customers within the population are created.

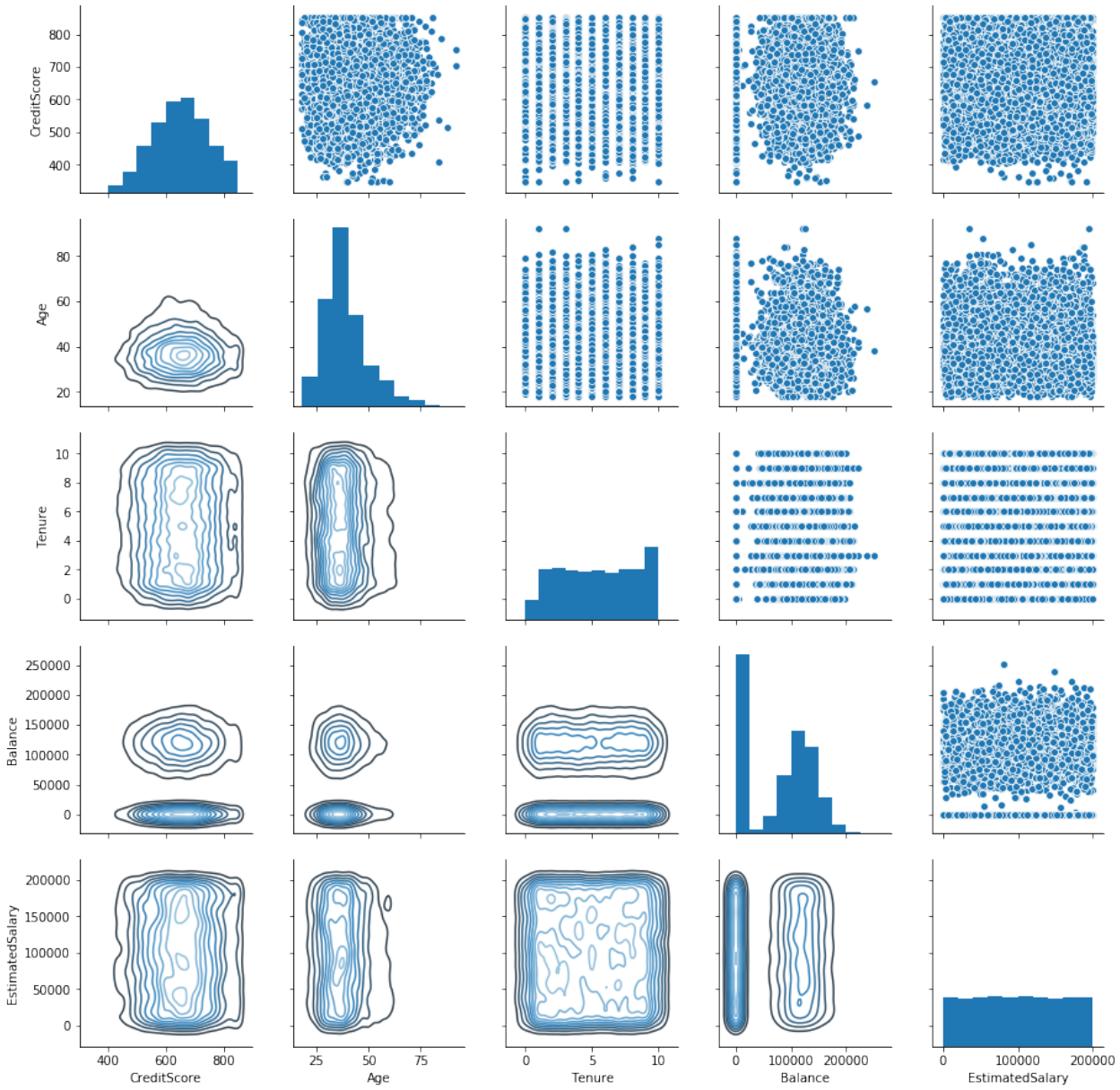

Figure 5.3: A pair plot for continuous features

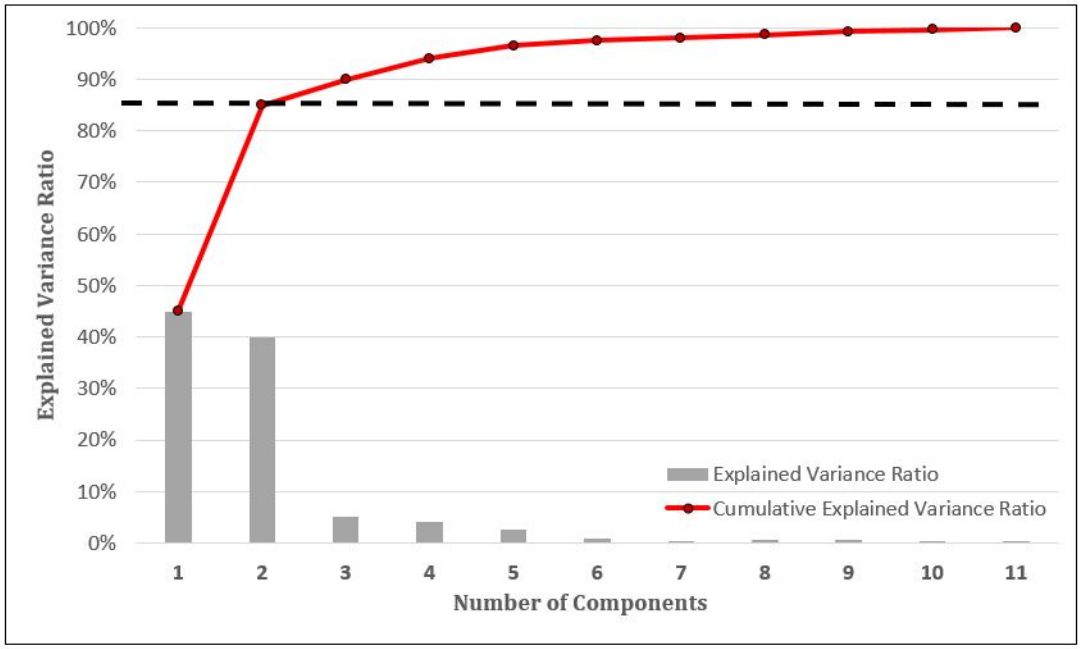

Figure 5.4: Explained variance ratio for each principal component

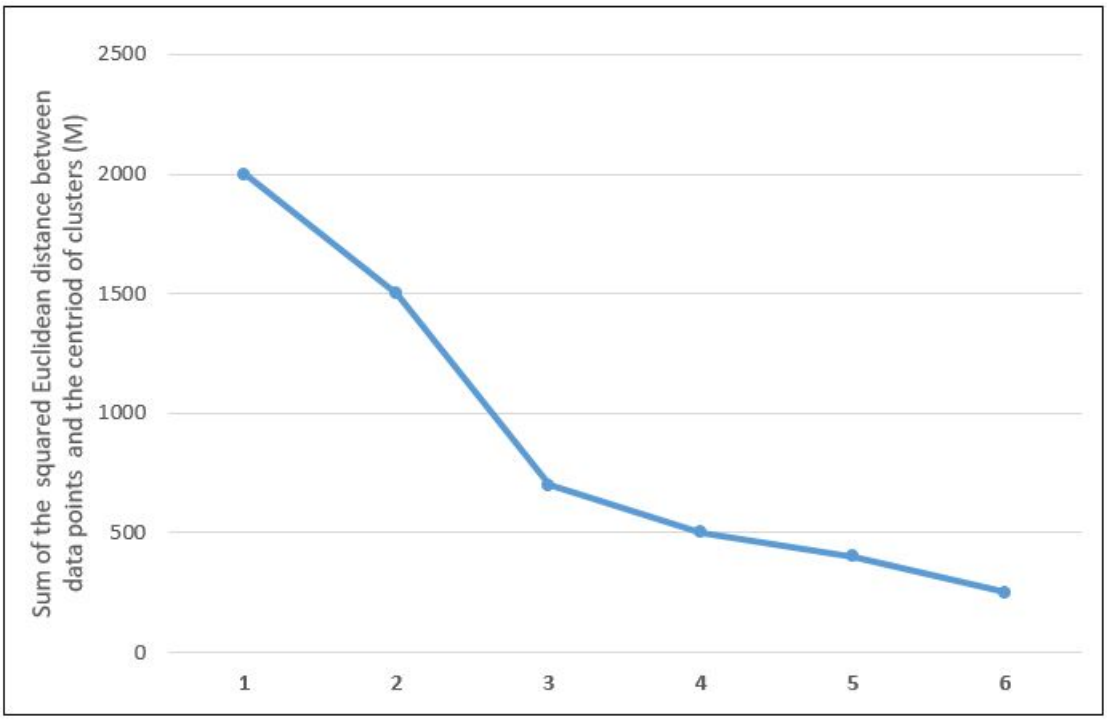

Figure 5.5: The number of clusters based on elbow method

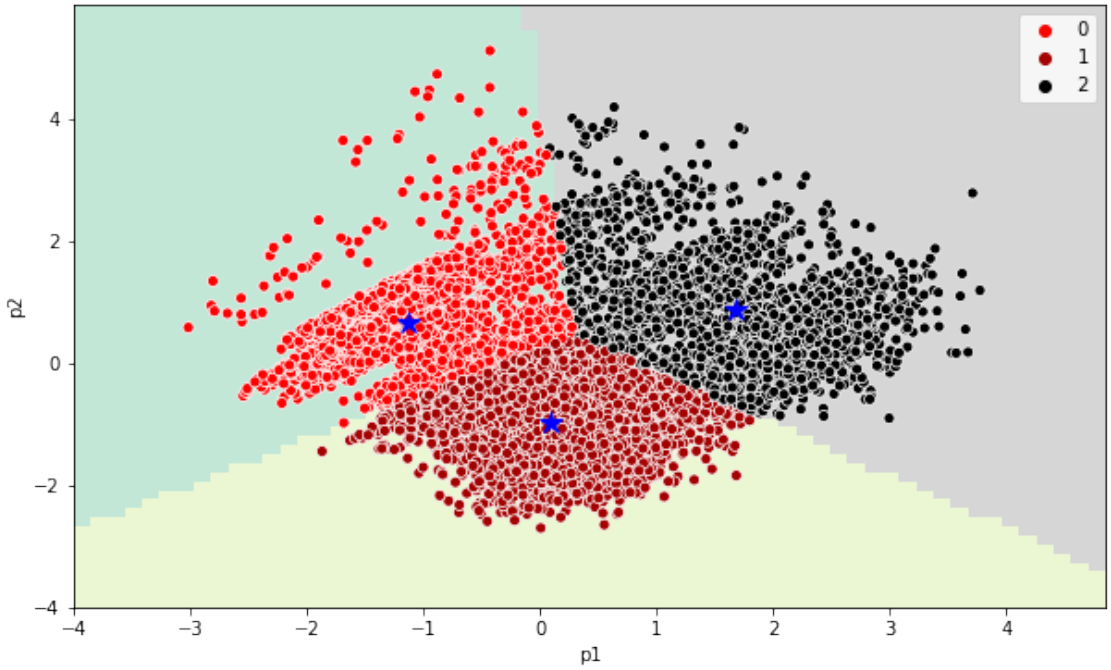

Figure 5.6: The three groups as suggested by the K-Means method

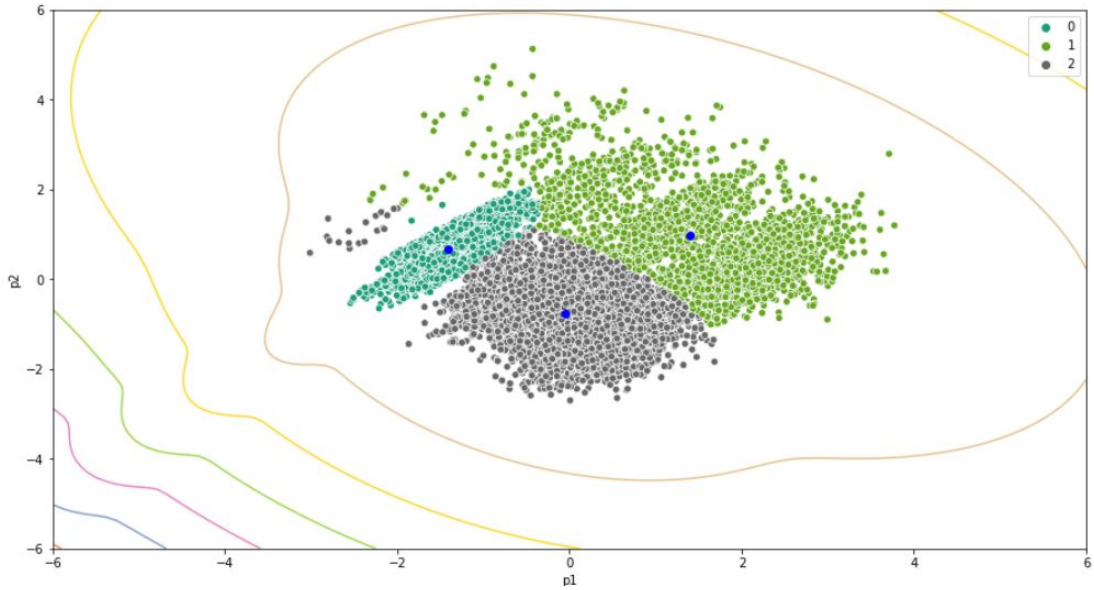

Figure 5.7: The three groups as suggested by the GMM method

6. Predicting the Wait Times for an $M/M/n$ Queue with Multiple Priorities

6.1 Introduction

There are a number of important questions remaining to be answered. For instance, what the wait time is if there are more than two priorities? How can the waiting time for each priority be predicted? After reviewing the literature, the need to addressing such questions is deeply felt.

Queuing systems with more than two priorities often occur in the real-world. For example, in an Emergency Department (ED), patients enter the hospital with various symptoms and acuity. The patients are then triaged, based on the severity of their conditions and according to the Emergency Severity Index (ESI) guidelines, in five groups:

1. Resuscitation: trauma, cardiac arrest, heart beat has stopped.

2. Emergent: life or limb threatening, chest pain, stroke.

3. Urgent: abdominal pain.

4. Non-urgent: vaccinated child pulling at ear, cold.

5. Referred: medication refill.

Realizing the above categories depend on the decision maker, and the impact of the groups on the system. Sometimes, the following groups are used:

1. Immediate and life threatening.

2. Urgent, but not necessarily immediately life threatening.

3. Less urgent.

Computing the waiting time for each group or priority is a challenging task; as the priority decreases, the difficulty of calculating the wait time via classical queueing theory approaches increases dramatically. In this chapter, a simulation-driven machine learning model is proposed for the mentioned problem. This method analyzes queueing systems with several servers and priority groups, or systems with general distribution.

6.2 Supervised Machine Learning Algorithms

Acquiring and analyzing data have a tremendous impact on understanding the situation and solving the problem effectively. Machine learning (ML) algorithms enhance the ability of dealing with data and extracting the desired knowledge. Figure 6.1 depicts the different types of machine learning problems.

Unsupervised learning algorithms were discussed in chapter 5. In the current chapter, supervised learning algorithms will be explained. In the case of supervised learning each data point has a label, which needs to be predicted.

Problems with finite number of target values are called classification problems. When the target variable is continuous, the problem is known as regression model, which replicates a target value based on independent variables. Classification and regression methods realize the relationship between variables and forecasting.

6.3 Simulation-Driven Machine Learning Framework

In every machine learning project two factors play a crucial role in the quality of the developed algorithms. The first factor is the quality of data. For instance, training an algorithm using data that contains many errors, outliers or noise reduces the accuracy of prediction or pattern detection. Therefore, data cleansing and organization play crucial roles [105]. The quality of features and their relevance to the problem affect the output of the algorithms, i.e., the quality of machine learning projects depend on access to proper data and relevant features. To create a beneficial dataset that

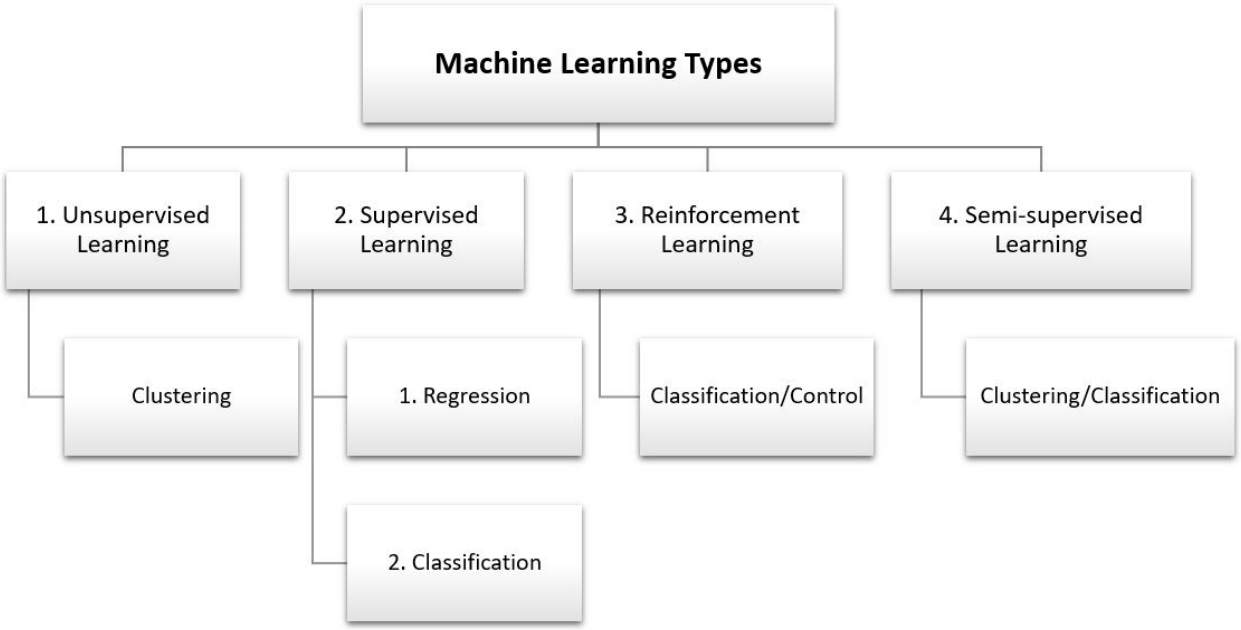

Figure 6.1: Different types of machine learning algorithms

facilitates predicting waiting time of customers, a simulation-driven machine learning model is proposed in this research shown in figure 6.2. One of the advantages of this model is that, instead of computing an average waiting time for a group of customers, it can predict a waiting time for each single type of customer, which is called *dynamic waiting time*.

6.4 Monte Carlo Simulation

Chapter 5 detailed the first section of the developed model, in which algorithms such as K-Means and GMM were used to find similar data points and categorize customers in similar groups. Furthermore, to analyze the system, information about arrival pattern and service time of different priority groups as well as the number of servers is required.

In the previous chapter, a population of customers was divided into three groups using clustering methods. Assume that all customers follow an exponential distribution for the arrival and service rate, as presented in table 6.1, and that the system has three servers. In other words, the problem is

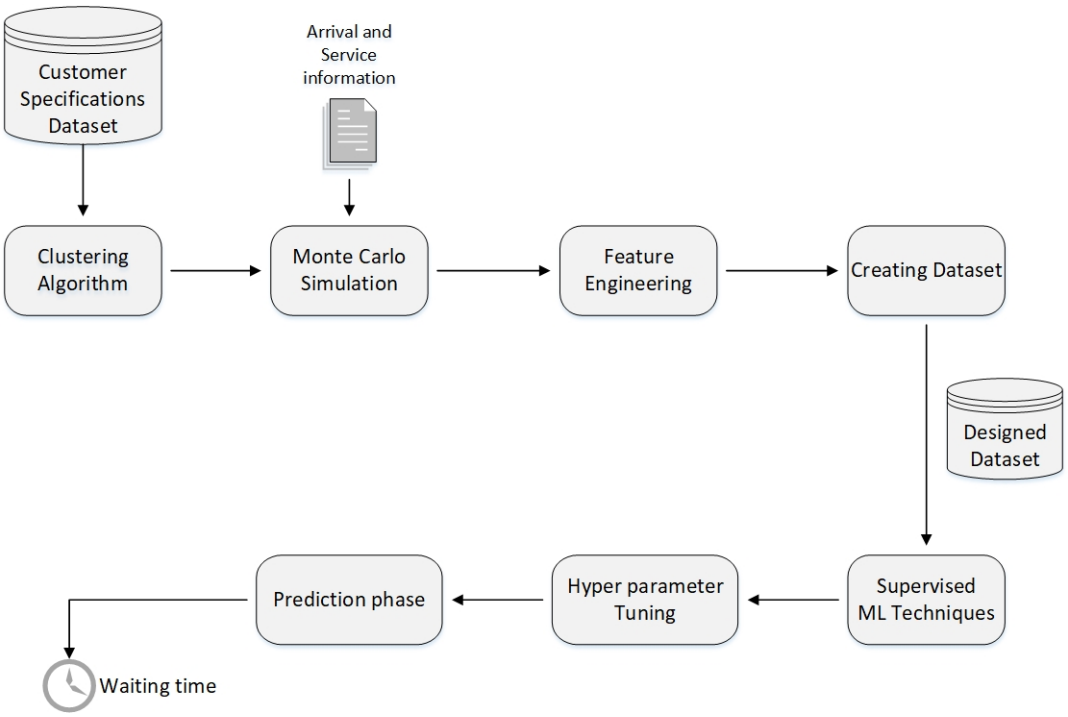

Figure 6.2: The proposed simulation-driven machine learning algorithm for predicting wait time

an $M/M/3$ queue model with three priorities. The number of customers for each group is considered to be 200, and the number of iteration for simulation is 10.

Table 6.1: Customer specification

Group (j)	Mean of arrival time (minutes)	Mean of service time (minutes)
1	15	20
2	8	7
3	6	4

Due to its high complexity, this problem cannot be analyzed in closed-form by the classical mathematical models. Hence, machine learning approaches become viable. To the best of the writer's knowledge, a public dataset with relevant features for predicting waiting times does not exist. Therefore, A Monte Carlo simulation method is used to make a dataset with the desired features. These features will be explained in the next section.

6.5 Feature Engineering

Feature engineering is a branch of machine learning that optimizes the features. Feature engineering consists of operations such as creating new features, adjusting features, normalization, scaling and so on, to improve an algorithm's prediction accuracy. Simulation-output features in this research include time-based elements, as well as the number and types of customers in the system. Table 6.2 explains these features.

Table 6.2: Features generated by simulation

CustomerId	Numerical	A unique value assigned to each customer
Type of Customer	Categorical	The type of customer entering the system
Entrance time	Numerical	The time a customer enters the system
Number 1	Numerical	The number of first-priority entities in the system when a customer enters
Number 2	Numerical	The number of second-priority entities in the system when a customer enters
Number 3	Numerical	The number of third-priority entities in the system when a customer enters
Server 1	Categorical	Type of customer being served by server 1 when a customer enters
Server 2	Categorical	Type of customer being served by server 2 when a customer enters
Server 3	Categorical	Type of customer being served by server 3 when a customer enters
Processing time	Numerical	Processing time of the last customer with the same priority when the customer arrives
Empty server	Numerical	If there is an empty server when a customer arrives
Wait time	Numerical	Wait time of the customer

6.6 Prediction Phase

So far, a dataset containing nine features is generated. Next, one can explore the relationship between these features and the target variable or waiting time, which is often referred to as exploratory data analysis. Afterward, multiple machine learning algorithms can be implemented to predict the waiting time of each customer.

6.6.1 Exploratory Data Analysis for Dataset

Distribution of arrival time

Figure 6.3 illustrates the distribution of arrival intervals between customers of each priority group. Although based on table 6.1 it was anticipated that arrival interval of group 1 will be greater than the other groups, sometimes arrival intervals for other groups are greater. Also, the dispersion of arrival intervals for group 1 is rather wide.

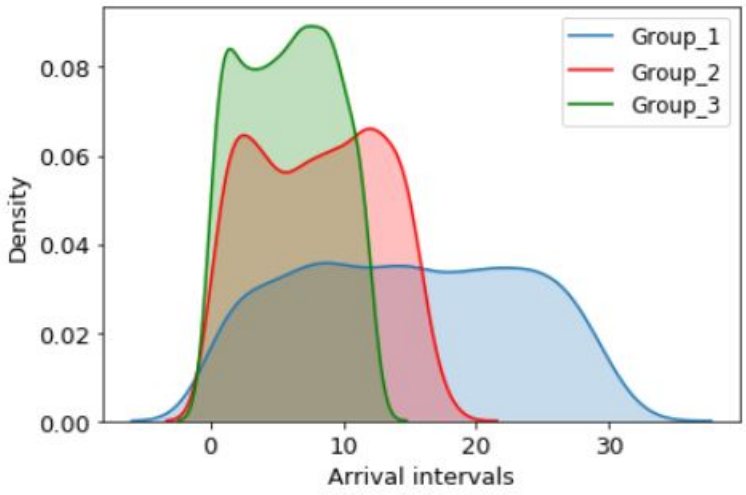

Figure 6.3: Distribution of arrival intervals for customers based on their priority group

Distribution of processing time

Figure 6.4 highlights the distribution of processing time for each priority group. Note that the simulation approach is always affected by outliers; for example, there are some customers with processing times greater than 70 minutes. Therefore, before implementing machine learning approaches, outliers are removed.

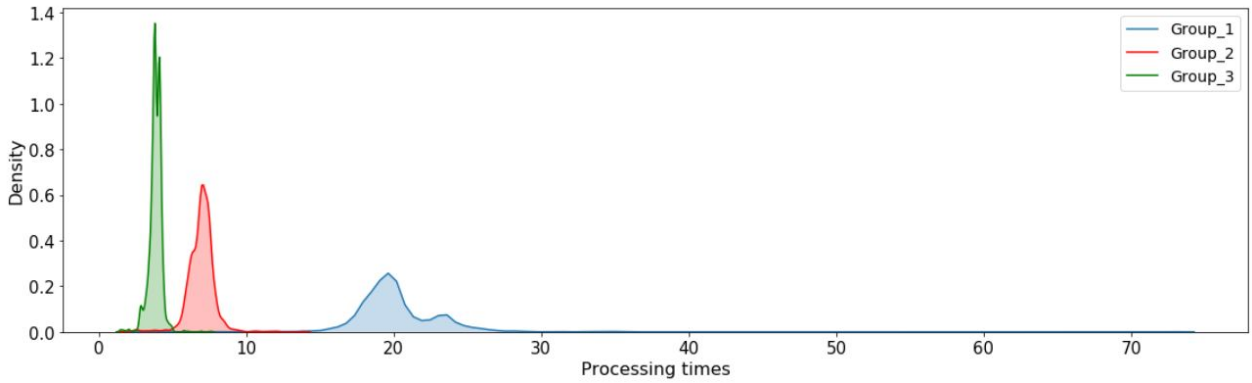

Figure 6.4: Distribution of processing time for each priority group

Server performance

Figure 6.5 shows the performance of the three servers when serving customers. This figure presents which priority group (vertical axis) each server serves through time (horizontal axis).

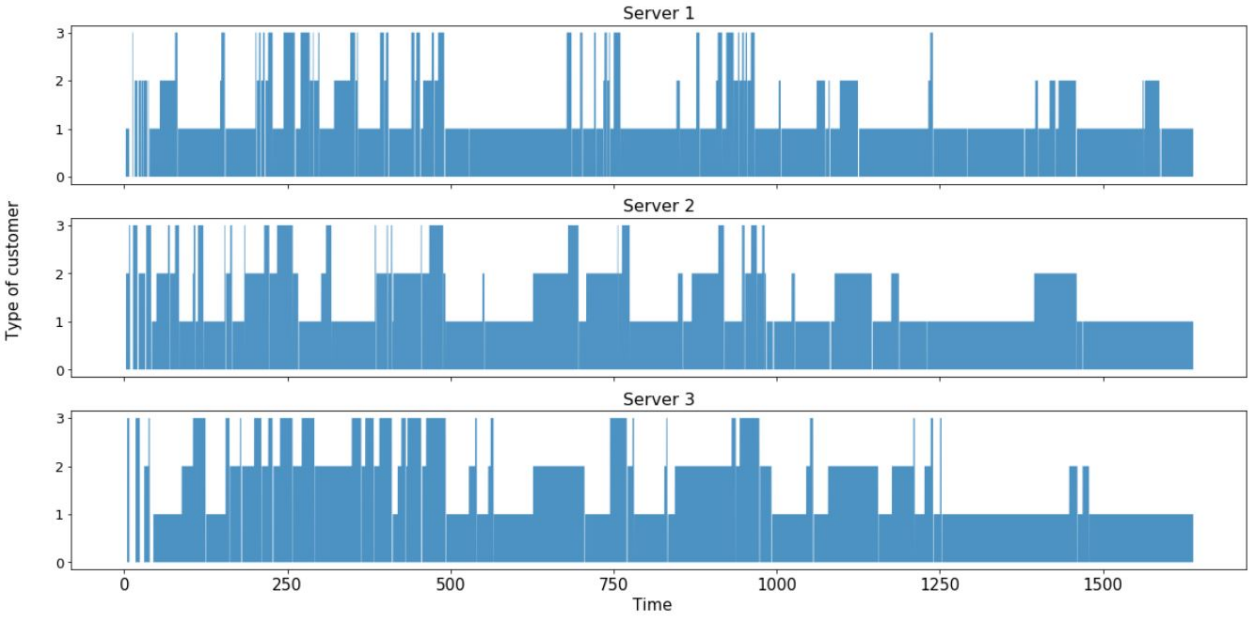

Figure 6.5: Performance of servers based on customer priority group

Waiting time

As depicted in figure 6.6, the target variable has a highly right-skewed distribution. As it was mentioned, simulation methods usually create data points that need to be manipulated before prediction for the results to be reliable. In other words a technique must be devised to handle outliers. An outlier is an observation with an abnormal distance from the rest of the values in a dataset. There are methods to realize and handle outliers. In this research, the policy for manipulating outliers is removing them. Box-plot is a useful tool to identify outliers. Figure 6.7 shows a box-plot for waiting times. Observations beyond the 4^{th} quartile (Q_4) are considered as outliers. Since the dataset contains 6000 records with $Q_4 = 108$, the number of realized outliers is less than 4% of the data; therefore, removing them does not affect the results.

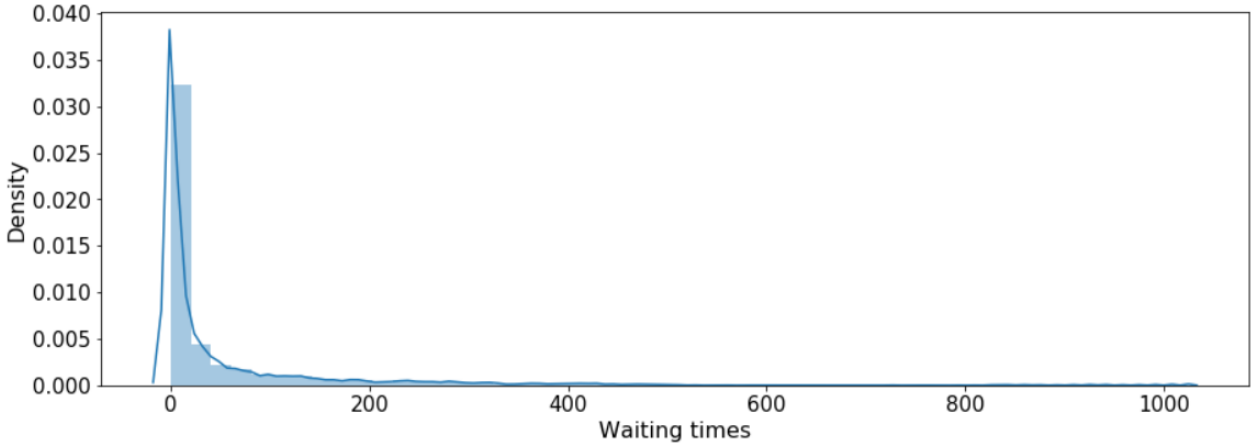

Figure 6.6: Distribution of the wait times

Correlation matrix of the features

Correlation matrix is a table that shows the correlation coefficients between the variables. Analyzing the correlation matrix has benefits such as finding patterns in the dataset, or generating input for more complex analyses such as factor analysis. Figure 6.8 indicates that “type of customer” and “empty server” features have the highest impact on the response variable, “waiting time.”

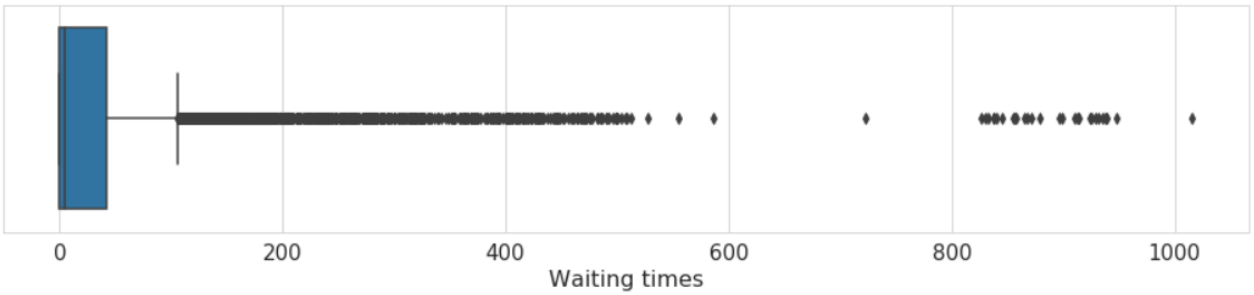

Figure 6.7: Box-plot of waiting times

6.7 Supervised Algorithms

Prediction of waiting time is in fact a regression problem in which the algorithms find a continuous wait value for each data point. Among several methods in the literature, the decision tree approach has proven to be a suitable technique for the problem under study. Once developed, the results of some machine learning approaches are compared with the results of the decision tree. In this section, besides decision tree and random forest, a few traditional machine learning algorithms such as linear regression, Ridge, Lasso and support vector regression are used to predict the waiting time that are elaborated briefly in the following sections.

6.7.1 Support Vector Regression

In contrast to the Ordinary Least Square (OLS) method, the objective function of Support Vector Regression (SVR) is to minimize the coefficients, more specifically, the L2-norm of the coefficient vector, not the squared error. The error term is instead handled in the constraints, where the absolute error should be less than or equal to a maximum error or ϵ. Epsilon can be tuned to gain the desired accuracy for the developed model. Thus, the new objective function and constraints are as follows:

Objective:

$$\min \quad Z = \frac{1}{2}||\mathbf{w}||^2 \tag{6.1}$$

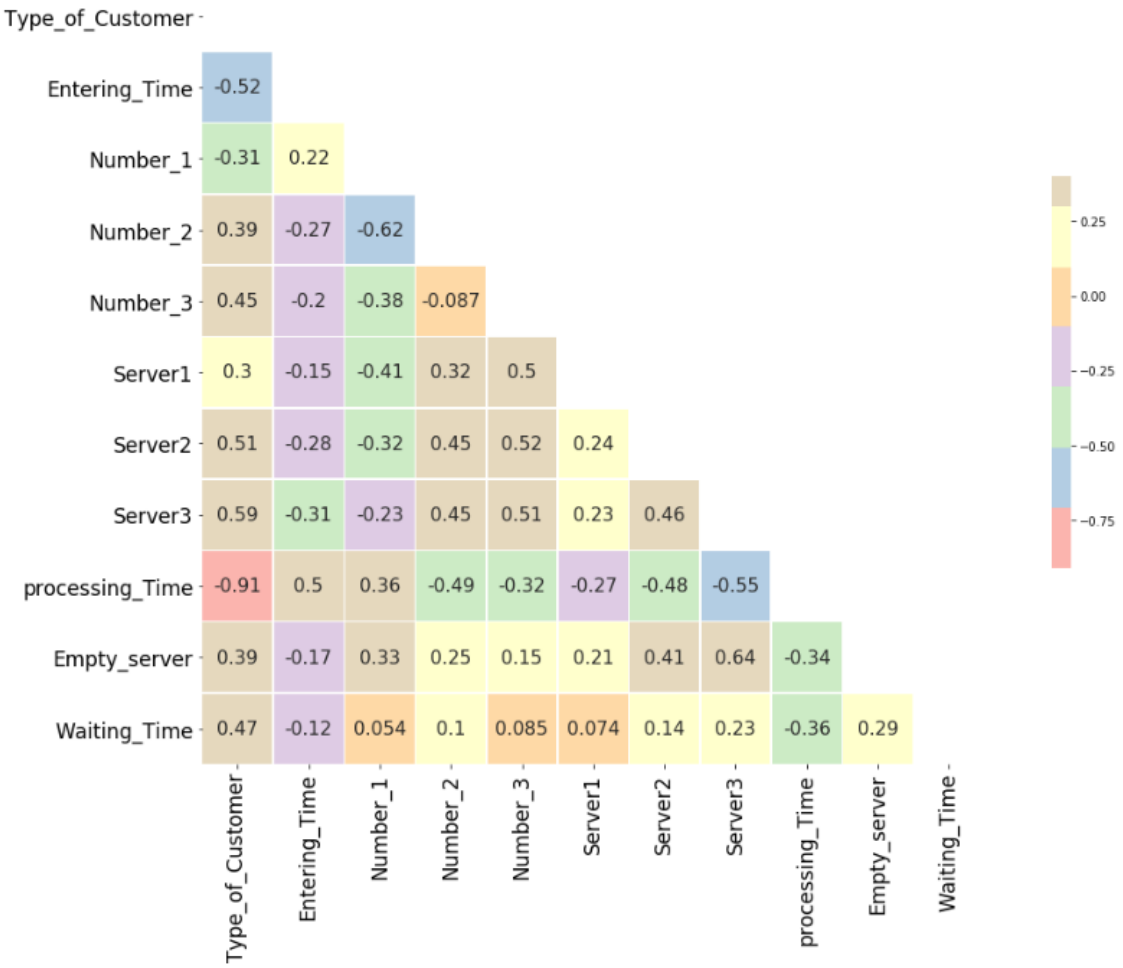

Figure 6.8: Correlation matrix for the considered features

Subject to:

$$\|y_i - w_i x_i\| \leq \epsilon \tag{6.2}$$

In the objective function, $Z = \frac{1}{2} \sum_{i=1}^{k} w_i^2$, k is the dimension of the problem. In SVR, the goal is to fit a model to predict a quantity for future [103]. Therefore, data points must be as close as possible to the hyper-plane. Figure 6.9 depicts the process. SVR determines the hyper-plane such that as many data points as possible separate by the boundary lines.

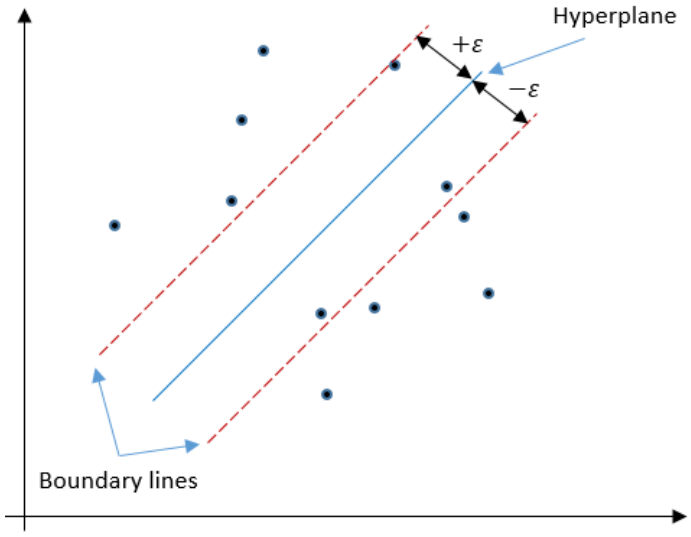

Figure 6.9: An illustrative example for SVR

6.7.2 Lasso and Ridge Regressions

Least absolute shrinkage and selection operator (Lasso) is a type of linear regression that shrinks the data toward a central point. The only difference between Lasso and linear regression is in their objective functions [103]. Equation (6.3) shows the Lasso objective function [103]:

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{k} w_j x_{ij})^2 + \lambda \sum_{j=0}^{k} |w_j| \tag{6.3}$$

λ is called the tuning parameter and performs regularization, which may lead to zero coefficients. Thus, some of the features may be neglected in the model with the help of the tuning parameter. In other words, not only Lasso regression reduces over-fitting, it can be useful in finding the best features for the problem.

Another type of linear regression is Ridge regression, which is similar to Lasso, with the objective function being the big difference [103]. The objective function of the Ridge regression is shown in equation (6.4).

$$\sum_{i=1}^{N}(y_i - \sum_{j=1}^{k} w_j x_{ij})^2 + \lambda \sum_{j=0}^{k} w_j^2 \tag{6.4}$$

The penalty term for Ridge regression keeps all the slopes or w_j terms. In contrast, the penalty term for the Lasso model removes unrelated variables.

6.7.3 Decision Tree

Decision tree is a prediction and classification tool, in which each internal node represents a test of an attribute, and branches denote the outcome. Finally, each leaf represents a label for a class or a prediction for the target variable. Figure 6.10 shows the structure of a decision tree. The decision tree method has its own pros and cons. The key advantages are:

1. Decision trees are easy to interpret and understand.
2. Unlike other techniques, the decision tree method needs little data preparation.
3. Unlike black-box models such as neural networks, the decision tree method is an easy-to-explain white-box technique.

One of the disadvantages of decision trees is the overfitting error. In other words, this technique may suggest a complex tree for a dataset which is not representative of the general case. Mechanisms such as defining the minimum number of samples at a leaf node, or detailing the maximum depth of the tree can mitigate this error. In addition, decision trees are highly sensitive to small variations in the data; using them in the form of ensemble techniques can alleviate this error.

6.7.4 Random Forest Regression

It was mentioned that one of the solutions to improve the performance of the decision tree algorithm is using the ensemble method. Random forest is a specific decision tree in which the tree is trained via a bagging method. The "maximum sample" parameter is defined for the bagging method to determine the size of training set for each tree [6].

Random forest algorithm uses extra randomness in finding the optimal decision tree. This prevents the algorithm from generating over-fitting error. In other words, extra randomness produces a bigger tree with higher bias and lower variance.

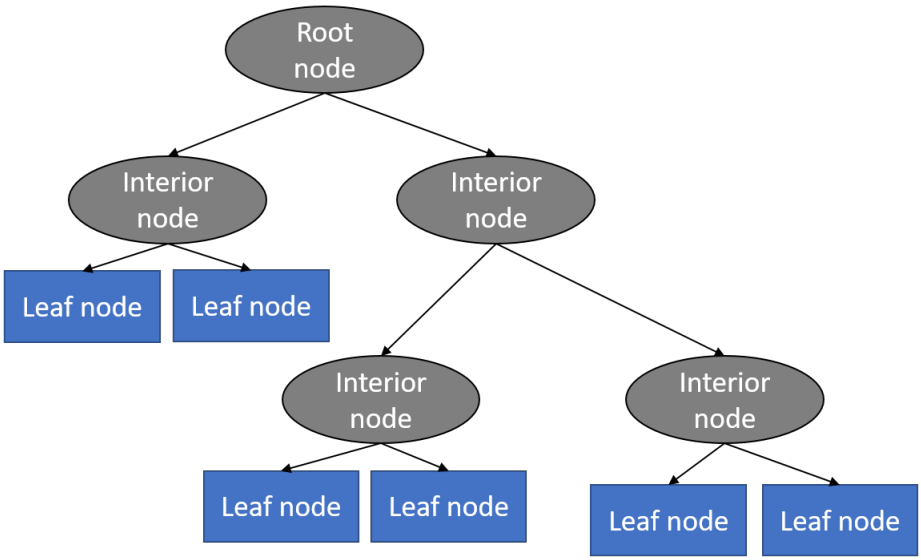

Figure 6.10: The structure of a decision tree

Figure 6.11 shows the concept behind the random forest algorithm. The algorithms combines many decision trees after training each one of them separately with a different set of data points and splitting the internal nodes of each tree considering a limited number of features. In the last step, the algorithm predicts the response variable by combining the predictions which have stemmed from these several decision trees using a method such as averaging.

6.8 Results

So far, five machine learning algorithms are implemented to predict the wait time of each individual customer. To compare the performance of these algorithms, a loss function metric is required.

6.8.1 Loss Function

Defining the loss function is a way for comparing the accuracy of predictions of the ML algorithms. Several loss functions can be defined for regression problems; examples include Mean Square Error (MSE) and Mean Absolute Error (MAE). MAE, which is the selected method of this

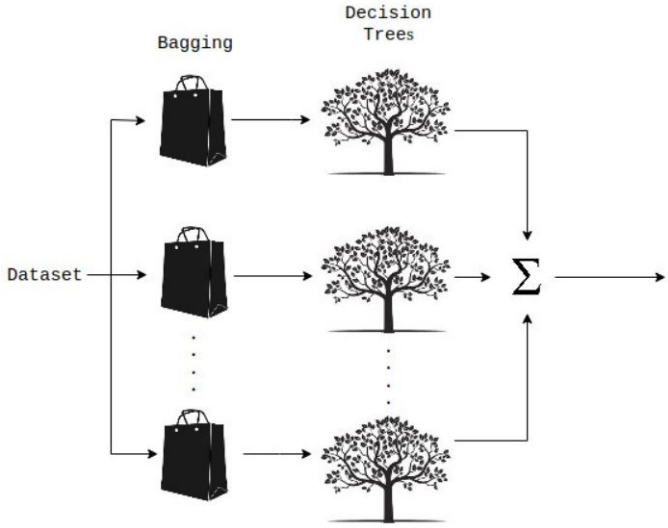

Figure 6.11: Random forest regression [6]

research, is the average of sum of absolute differences between predictions and actual observations:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \widehat{y_i}|}{n} \tag{6.5}$$

Figure 6.12 reports the MAE of each implemented prediction algorithm. Accordingly, the random forest regressor results in the lowest MAE value; SVR gives the highest MAE.

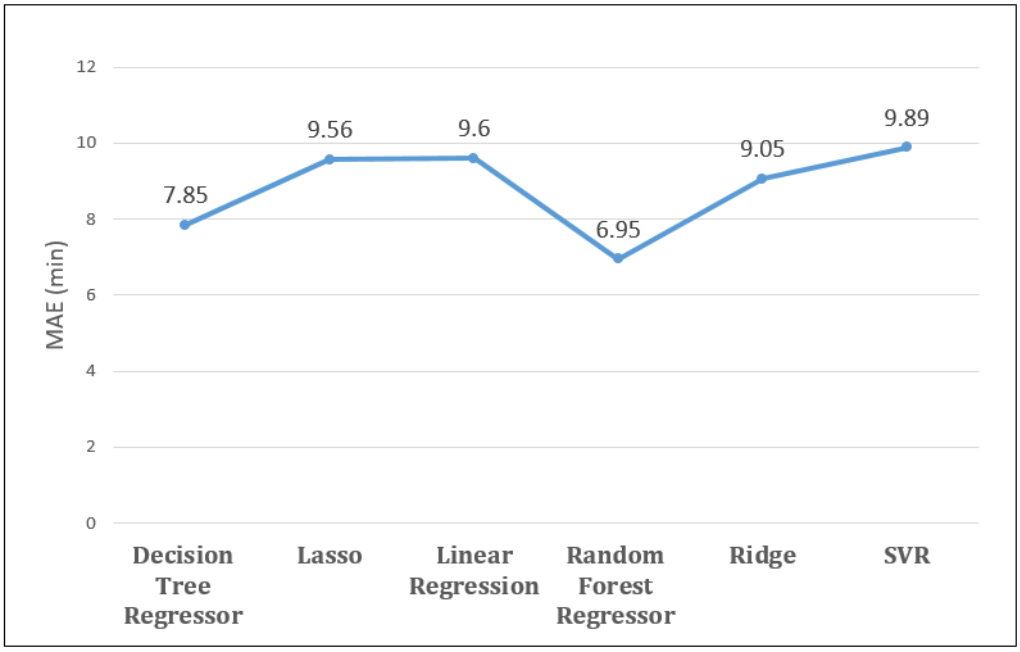

Figure 6.12: Mean absolute error of prediction for each algorithm

7. Conclusion

7.1 Discussion and Summary

In this research, two queueing structures were investigated: 1) independent queues; 2) a single merged queue line. Among the influential factors on the performance of a queueing system, apart from its structure, are number of servers, rate of customer entrance, service rate, and system capacity. Moreover, an organization often assigns different importance levels to the entities entering the queueing system. In other words, the organization prioritizes the entities; such priorities have an immense effect on the behavior and performance of the queueing system.

This research developed a model for an independent queue structure with multiple priorities. The presented model optimizes the weighted sum of average waiting times while finding the optimum number of servers for the defined priority groups. This model was presented as an NLP model, and solved numerically.

A separate model was presented to analyze the merged queue structure. The model was used to evaluate an $M/M/3$ preemptive queue with two priorities, which was evaluated by two infinite-dimensional Markov chains. Later, Osogami's technique was employed to convert the two infinite-dimensional chains to a one infinite-dimensional chain, and to find the stationary probabilities. The results of the developed algorithm was compared with the results of a simulation of the problem; the gap between the two methods was almost 7%.

Another challenging dilemma is defining the various priorities and separating customers based on the designated criteria. This thesis proposed a method to consider all of the measured features of the customers for segmentation. Two renowned methods, K-Means and GMM, were applied to a real dataset to cluster its entities. PCA method classified the customers to three groups.

It was argued that selecting the right machine learning method for the right problem is necessary to achieve the best results. Furthermore, it was argued that feature engineering, i.e., the process of modifying data for machine learning, is also an essential factor in obtaining the best prediction

results. A simulation-driven machine learning framework was developed to predict the waiting time of the customers. This model introduced some beneficial features for measuring the performance of the queueing systems.

All of the designed features were based on time, server, and customer priority; adding these features to the collected data in any organization is straightforward. Among the studied algorithms, random forest regression approach, with MAE of 6.95 minutes, was the best technique to predict the waiting time.

Based on the proposed framework wait time is a dynamic value. In other words, in the proposed approach, unlike the classical queueing theory which calculates an average wait time for each *group*, a specific wait time is calculated for each *customer*. The number of customers in the system, average processing time for each group, and arrival rates are among the influential factors when calculating the wait times.

7.2 Future Research

The above explanations lead to the following opportunities for future research:

- One expansion possibility for the dedicated queues is considering a divided queue structure with *time-dependent* parameters. This generalization results in dynamic systems, frequently seen in the real-world.

  Such expansions are useful for analyzing systems in which the arrival and service rates change at different time intervals. Prime examples include taxi stations in a city, bank servers, or doctors and nurses in healthcare settings. The commonality between all of the mentioned environments is that the system experiences high and low periods, rush hours, or warm-up and cool-down intervals.

- Another motivating extension is a merged queue structure which consists of more than two priorities. Once again, Osogami's approach may be employed to deal with an $M/M/n$ model with three priorities.

- Other techniques such as hierarchical clustering, DBSCAN, or hybrid clustering can be used for customer clustering and segmentation.
- Applying other machine learning techniques such as neural networks or ensemble techniques may result in more accuracy in prediction, and hence, is considered an inspiring future research direction.

Bibliography

[1] John F Shortle, James M Thompson, Donald Gross, and Carl M Harris. *Fundamentals of queueing theory*, volume 399. John Wiley & Sons, 2018. ix, x, 16, 25, 26, 27

[2] Su-Yeon Kim, Tae-Soo Jung, Eui-Ho Suh, and Hyun-Seok Hwang. Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert Systems With Applications*, 31(1):101–107, 2006. x, 12

[3] Goran Petrović, Nikola Petrović, and Zoran Marinković. Application of the markov theory to queuing networks. *Facta Universitatis-Series: Mechanical Engineering*, 6(1):45–56, 2008. x, 19, 20, 29

[4] Matthias Scholz. *Approaches to analyse and interpret biological profile data*. PhD thesis, Mathematics Department, University Postdam, 2006. xi, 78

[5] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. xi, 81, 83

[6] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019. xii, 103, 105

[7] Wenhong Luo, Matthew J Liberatore, Robert L Nydick, Q B Chung, and Elliot Sloane. Impact of process change on customer perception of waiting time: a field study. *Omega*, 32(1):77–83, 2004. 1

[8] Frédéric Bielen and Nathalie Demoulin. Waiting time influence on the satisfaction-loyalty relationship in services. *Managing Service Quality: An International Journal*, 17(2):174–193, 2007. 1

[9] Vikas Kumar, Luciano Batista, and Roger Maull. The impact of operations performance on customer loyalty. *Service Science*, 3(2):158–171, 2011. 1

[10] Olanrewaju A Soremekun, James K Takayesu, and Stephen J Bohan. Framework for analyzing wait times and other factors that impact patient satisfaction in the emergency department. *The Journal of Emergency Medicine*, 41(6):686–692, 2011. 1

[11] Danilo Garcia, Trevor Archer, Saleh Moradi, and Bibinaz Ghiabi. Waiting in vain: managing time and customer satisfaction at call centers. *Psychology*, 3(02):213, 2012. 1

[12] Nigel Hill and Jim Alexander. *The handbook of customer satisfaction and loyalty measurement*. Routledge, 2017. 1

[13] Jianfu Wang, Opher Baron, and Alan Scheller-Wolf. M/m/c queue with two priority classes. *Operations Research*, 63(3):733–749, 2015. 1, 8

[14] Philipp Afeche, Mojtaba Araghi, and Opher Baron. Customer acquisition, retention, and queueing-related service quality: optimal advertising, staffing, and priorities for a call center. *Manufacturing & Service Operations Management*, 19(4), 2015. 1, 9

[15] Takayuki Osogami and Mor Harchol-Balter. *A closed-form solution for mapping general distributions to minimal PH distributions*. Springer, 2003. 4, 63, 65

[16] Håvard Thøgersen, Bjørn Møller, Linn Merete Åsli, Sameer Bhargava, Rune Kvåle, Lars Fjellbirkeland, Trude Eid Robsahm, Stein Aaserud, Ronnie Babigumira, and Inger Kristin Larsen. Waiting times and treatment following cancer diagnosis: comparison between immigrants and the norwegian host population. *Acta Oncologica*, pages 1–8, 2020. 6

[17] Suzanne Robat, Tanja Gagliardi, Daina Greiskalna, and Yazan Masannat. Do waiting times have an impact on breast cancer tumour sizes? *European Journal of Surgical Oncology*, 46(2):e55, 2020. 6

[18] Yuta Kumazu, Koji Oba, Tsutomu Hayashi, Takanobu Yamada, Kentaro Hara, Hiroaki Osakabe, Yota Shimoda, Masato Nakazono, Shinsuke Nagasawa, Yasushi Rino, et al. Relationship between the waiting times for surgery and survival in patients with gastric cancer. *World Journal of Surgery*, pages 1–7, 2020. 6

[19] Dorit Efrat-Treister, Hadar Moriah, and Anat Rafaeli. The effect of waiting on aggressive tendencies toward emergency department staff: Providing information can help but may also backfire. *PLOS One*, 15(1), 2020. 6

[20] Farzad Zeinali, Masoud Mahootchi, and Mohammad Mehdi Sepehri. Resource planning in the emergency departments: A simulation-based metamodeling approach. *Simulation Modelling Practice and Theory*, 53:123–138, 2015. 6

[21] Che Soong Kim, Vilena V Mushko, and Alexander N Dudin. Computation of the steady state distribution for multi-server retrial queues with phase type service process. *Annals of Operations Research*, 201(1):307–323, 2012. 6

[22] KR Rejitha and KP Jose. A stochastic inventory system with two modes of service and retrial of customers. *Opsearch*, 55(1):134–149, 2018. 7

[23] Csaba Farkas and Miklós Telek. Capacity planning of electric car charging station based on discrete time observations and map (2)/g/c queue. *Periodica Polytechnica Electrical Engineering and Computer Science*, 62(3):82–89, 2018. 7

[24] Qingqing Ye and Liwei Liu. Analysis of map/m/1 queue with working breakdowns. *Communications in Statistics-Theory and Methods*, 47(13):3073–3084, 2018. 7

[25] Tuan Phung-Duc and Ken'ichi Kawanishi. Multiserver retrial queue with setup time and its application to data centers. *Journal of Industrial and Management Optimization*, 15(1):15–35, 2019. 7

[26] Gianluca Mazzini, Riccardo Rovatti, and Gianluca Setti. A closed form solution of bernoullian two-classes priority queue. *IEEE Communications Letters*, 9(3):264–266, 2005. 8

[27] Joris Walraevens, Dieter Claeys, and Tuan Phung-Duc. Asymptotics of queue length distributions in priority retrial queues. *Performance Evaluation*, 127-128:235 – 252, 2018. 8

[28] Vahid Sarhangian, Hossein Abouee-Mehrizi, Opher Baron, and Oded Berman. Threshold-based allocation policies for inventory management of red blood cells. *Manufacturing & Service Operations Management*, 20(2):347–362, 2017. 8

[29] Roshli Aniyeri and C Ratnam Nadar. A multiphase queuing system with assorted servers by using matrix geometric method. *International Journal of Applied Engineering Research*, 12(22):12052–12059, 2017. 9

[30] Mark Fackrell. Modelling healthcare systems with phase-type distributions. *Health Care Management Science*, 12(1):11, 2009. 9

[31] Eimutis Valakevicius. The application of phase type distributions for modelling queueing systems. *Journal of Systemics, Cybernetics and Informatics*, 5(6):28–32, 2007. 10, 28, 30

[32] Hideaki Takagi and Yoshitaka Takahashi. Priority queues with batch poisson arrivals. *Operations Research Letters*, 10(4):225–232, 1991. 10

[33] Douglas R Miller. Computation of steady-state probabilities for m/m/1 priority queues. *Operations Research*, 29(5):945–958, 1981. 10

[34] Andreas Brandt and Manfred Brandt. On a two-queue priority system with impatience and its application to a call center. *Methodology and Computing in Applied Probability*, 1(2):191–210, 1999. 10

[35] Don Towsley and Satish K Tripathi. A single server priority queue with server failures and queue flushing. *Operations Research Letters*, 10(6):353–362, 1991. 10

[36] H Richard Gail, Sidney L Hantler, and BA Taylor. On a preemptive markovian queue with multiple servers and two priority classes. *Mathematics of Operations Research*, 17(2):365–391, 1992. 10

[37] Attahiru Sule Alfa. Matrix-geometric solution of discrete time map/ph/1 priority queue. *Naval Research Logistics*, 45(1):23–50, 1998. 10

[38] Gido AJF Brouns and Jan Van Der Wal. Optimal threshold policies in a two-class preemptive priority queue with admission and termination control. *Queueing Systems*, 54(1):21–33, 2006. 11

[39] Andrei Sleptchenko, Jori Selen, Ivo Adan, and Geert-Jan van Houtum. Joint queue length distribution of multi-class, single-server queues with preemptive priorities. *Queueing Systems*, 81(4):379–395, 2015. 11

[40] Andrei Sleptchenko, Aart Van Harten, and Matthieu Van Der Heijden. An exact solution for the state probabilities of the multi-class, multi-server queue with preemptive priorities. *Queueing Systems*, 50(1):81–107, 2005. 11

[41] Azaz Bin Sharif, David A Stanford, Peter Taylor, and Ilze Ziedins. A multi-class multi-server accumulating priority queue with application to health care. *Operations Research for Health Care*, 3(2):73–79, 2014. 11

[42] Rinus Haaijer, Michel Wedel, Marco Vriens, and Tom Wansbeek. Utility covariances and context effects in conjoint mnp models. *Marketing Science*, 17(3):236–252, 1998. 11

[43] Daniel Yankelovich and David Meer. Rediscovering market segmentation. *Harvard Business Review*, 84(2):122, 2006. 11

[44] Derek J Wade and Paul FJ Eagles. The use of importance–performance analysis and market segmentation for tourism management in parks and protected areas: An application to tanzania's national parks. *Journal of Ecotourism*, 2(3):196–212, 2003. 11

[45] Gillian Martin. The importance of marketing segmentation. *American Journal of Business Education (AJBE)*, 4(6):15–18, 2011. 11

[46] Robert S Duboff. Marketing to maximize profitability. *Journal of Business Strategy*, 13(6):10–13, 1992. 11

[47] Brent A Gloy, Jay T Akridge, and Paul V Preckel. Customer lifetime value: An application in the rural petroleum market. *Agribusiness: An International Journal*, 13(3):335–347, 1997. 11

[48] Bruce Cooil, Lerzan Aksoy, and Timothy L Keiningham. Approaches to customer segmentation. *Journal of Relationship Marketing*, 6(3-4):9–39, 2008. 12

[49] Paul E Green. A new approach to market segmentation. *Business Horizons*, 20(1):61–73, 1977. 13

[50] Thorsten Teichert, Edlira Shehu, and Iwan von Wartburg. Customer segmentation revisited: The case of the airline industry. *Transportation Research Part A: Policy and Practice*, 42(1):227–242, 2008. 13

[51] Daqing Chen, Sai Laing Sain, and Kun Guo. Data mining for the online retail industry: A case study of rfm model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19(3):197–208, 2012. 13

[52] Wenjie Bi, Meili Cai, Mengqi Liu, and Guo Li. A big data clustering algorithm for mitigating the risk of customer churn. *IEEE Transactions on Industrial Informatics*, 12(3):1270–1281, 2016. 13

[53] Ning Lu, Hua Lin, Jie Lu, and Guangquan Zhang. A customer churn prediction model in telecom industry using boosting. *IEEE Transactions on Industrial Informatics*, 10(2):1659–1665, 2012. 13

[54] Chih-Fong Tsai, Ya-Han Hu, and Yu-Hsin Lu. Customer segmentation issues and strategies for an automobile dealership with two clustering techniques. *Expert Systems*, 32(1):65–76, 2015. 13, 83

[55] Sriramakrishnan Chandrasekaran and Abhishek Kumar. A clustering approach for customer billing prediction in mall: A machine learning mechanism. *Journal of Computer and Communications*, 7(3):55–66, 2019. 14

[56] Tuncay Bayrak. A review of business analytics: a business enabler or another passing fad. *Procedia-Social and Behavioral Sciences*, 195:230–239, 2015. 14

[57] Chandi Ram Kalita and Gautam Choudhury. An repeated service queue with n-policy and setup time subject to server's breakdown and delayed repair. *Communications in Statistics-Theory and Methods*, pages 1–29, 2019. 20

[58] Chandi Ram Kalita and Gautam Choudhury. The n-policy for an unreliablequeue (mx/g 1 g 2/1 queue) with optional repeated service and delayed repair. *Communications in Statistics-Theory and Methods*, 48(15):3717–3745, 2019. 20

[59] R Sudhesh and R Sebasthi Priya. An analysis of discrete-time geo/geo/1 queue with feedback, repair and disaster. *International Journal of Operational Research*, 35(1):37–53, 2019. 20

[60] Sreekanth Kolledath and Kamlesh Kumar. *Performance Analysis of Series Queue with Customer's Blocking*. Springer, 2019. 20

[61] Sumaiya Salim Al Mahrooqi and Syed Zakir Ali. Petrol pump queue management system for sultanate of oman using artificial intelligence technique. In *2019 China-Qatar International Workshop on Artificial Intelligence and Applications to Intelligent Manufacturing (AIAIM)*, pages 22–29. IEEE, 2019. 20

[62] Ellen C Daniels, Jon Bryan Burley, Trisha Machemer, and Paul Nieratko. Theme park queue line perception. *International Journal and Cultural Heritage*, 2:105–108, 2017. 20

[63] Roshli Aniyeri and Ratnam Nadar. Passengers queue analysis in international airports terminals in kerala using multiphase queuing system. *International Journal of Mathematics in Operational Research*, 12(1):1–30, 2018. 20

[64] Winfried K Grassmann. *Stochastic systems for management*. North-Holland, 1981. 21, 29, 31, 32, 57

[65] Robert B. Cooper. *Introduction to Queuing Theory*. Elsevier North Holland. Inc, 1981. 23

[66] Richard Serfozo. *Basics of applied stochastic processes*. Springer Science & Business Media, 2009. 29

[67] Guy Latouche, V Ramaswami, and VG Kulkarni. Introduction to matrix analytic methods in stochastic modeling. *Journal of Applied Mathematics and Stochastic Analysis*, 12(4):435–436, 1999. 29, 40, 52, 70

[68] Deyvison T Baia Medeiros, Shoshana Hahn-Goldberg, Erin O'Connor, and Dionne M Aleman. Analysis of emergency department length of stay for mental health visits: a case study of a canadian academic hospital. *Canadian Journal of Emergency Medicine*, 21(3):374–383, 2019. 31

[69] Marcel F Neuts. *Matrix-geometric solutions in stochastic models: an algorithmic approach*. Courier Corporation, 1994. 40, 62

[70] Natarajan Gautam. *Analysis of queues: methods and applications*. CRC Press, 2012. 40, 41, 42, 45, 50

[71] Hendrik Baumann and Werner Sandmann. Numerical solution of level dependent quasi-birth-and-death processes. *Procedia Computer Science*, 1(1):1561–1569, 2010. 41

[72] Tuğrul Dayar, Werner Sandmann, David Spieler, and Verena Wolf. Infinite level-dependent qbd processes and matrix-analytic solutions for stochastic chemical kinetics. *Advances in Applied Probability*, 43(4):1005–1026, 2011. 41

[73] Thomas G Kurtz. The relationship between stochastic and deterministic models for chemical reactions. *The Journal of Chemical Physics*, 57(7):2976–2978, 1972. 41

[74] Tien Do and Ram Chakka. *Generalized QBD processes, spectral expansion and performance modeling applications*. 2010. 41

[75] Ji Quan, Wei Liu, Yuqing Chu, and Xianjia Wang. Stochastic evolutionary voluntary public goods game with punishment in a quasi-birth-and-death process. *Scientific Reports*, 7(1):16110, 2017. 41

[76] Jesus R Artalejo and A Gómez-Corral. Markov chains with block-structured state-dependent events: Formulation and correlation analysis. In *Proceedings of the 5th International Conference on Queueing Theory and Network Applications*, pages 23–28, 2010. 41

[77] Gunter Bolch, Stefan Greiner, Hermann De Meer, and Kishor S Trivedi. *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. John Wiley & Sons, 2006. 45, 50, 51

[78] Richard M Feldman and Ciriaco Valdez-Flores. *Applied probability and stochastic processes*. Springer Science & Business Media, 2009. 53

[79] Mor Harchol-Balter, Takayuki Osogami, Alan Scheller-Wolf, and Adam Wierman. Multi-server queueing systems with multiple priority classes. *Queueing Systems*, 51(3-4):331–360, 2005. 58, 61

[80] KG Terry Hollands and Harry Suehrcke. A three-state model for the probability distribution of instantaneous solar radiation, with applications. *Solar Energy*, 96:103–112, 2013. 62

[81] Joakim Munkhammar, Jesper Rydén, and Joakim Widén. Characterizing probability density distributions for household electricity load profiles from high-resolution electricity use data. *Applied Energy*, 135:382–390, 2014. 62

[82] Gilles Chabrier. Galactic stellar and substellar initial mass function. *Publications of the Astronomical Society of the Pacific*, 115(809):763, 2003. 62

[83] Lars Mattsson and Susanne Höfner. Dust-driven mass loss from carbon stars as a function of stellar parameters-ii. effects of grain size on wind properties. *Astronomy & Astrophysics*, 533:A42, 2011. 62

[84] Adam Hugh Monahan. Empirical models of the probability distribution of sea surface wind speeds. *Journal of Climate*, 20(23):5798–5814, 2007. 62

[85] Michel K Ochi. *Ocean waves: the stochastic approach*, volume 6. Cambridge University Press, 2005. 62

[86] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002. 62

[87] Matt P Wand and M Chris Jones. *Kernel smoothing*. Chapman and Hall, 1994. 62

[88] Jean Gallier and Jean H Gallier. *Curves and surfaces in geometric modeling: theory and algorithms*. Morgan Kaufmann, 2000. 62

[89] Takayuki Osogami. *Analysis of multi-server systems via dimensionality reduction of Markov chains*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005. 62, 67, 70

[90] Mingsheng Ying and Yuan Feng. Model checking quantum systems-a survey. *ArXiv Preprint*, 2018. 62

[91] P Pecka, MP Nowak, A Rataj, and S Nowak. *Solving Large Markov Models Described with Standard Programming Language*. Springer, 2018. 62

[92] P Maheswari, C Selvakumar, and C Elango. Optimal control of customers to the service facility with two types of customers. *International Journal of Fuzzy Mathematical Archive*, 15(2):217–226, 2018. 62

[93] S Krishnakumar and C Elango. Perishable inventory control in retrial service facility-semi markov decision process. *International Journal of Fuzzy Mathematical Archive*, 15(2):235–244, 2018. 62

[94] Stratos Ioannidis, Alexandros S Xanthopoulos, Ioannis Sarantis, and Dimitrios E Koulouriotis. Joint production, inventory rationing, and order admission control of a stochastic manufacturing system with setups. *Operational Research*, pages 1–29, 2019. 62

[95] Robert Shone, Kevin David Glazebrook, and Konstantinos Zografos. Stochastic modelling of aircraft queues: A review. *OR60 Annual Conference*, 2018. 62

[96] Jesús R Artalejo and MJ Lopez-Herrero. Analysis of the busy period for the m/m/c queue: An algorithmic approach. *Journal of Applied Probability*, 38(1):209–222, 2001. 66

[97] Foster Provost and Tom Fawcett. *Data Science for Business: What you need to know about data mining and data-analytic thinking*. O'Reilly Media, 2013. 73

[98] Vasant Dhar. Data science and prediction. *Communications of the ACM*, 56(12):64–73, 2013. 73

[99] Chikio Hayashi. *Data science, classification, and related methods*. Springer, 1998. 73

[100] Arthur J Samuel. *Aerosol dispensers and like pressurized packages*. Google Patents, September 15 1959. US Patent 2,904,229. 73

[101] Richard Ernest Bellman and Stuart E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, 1957. 74

[102] Gavin Hackeling. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017. 74, 79

[103] Wei-Meng Lee. *Python Machine Learning*. John Wiley & Sons, 2019. 74, 79, 101, 102

[104] Konstantinos K Tsiptsis and Antonios Chorianopoulos. *Data mining techniques in CRM: inside customer segmentation*. John Wiley & Sons, 2011. 74

[105] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, 2018. 75, 93

[106] Hartigan JA Wong and JA Hartigan. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979. 79

[107] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–22, 1977. 83