PRICE OPTIMIZATION IN STOCHASTIC LOSS SYSTEMS

PRICE OPTIMIZATION IN STOCHASTIC LOSS SYSTEMS

BY

SEYYED MOHAMMAD HASHEMI KAROUEE, B.Sc.

A THESIS

SUBMITTED TO THE SCHOOL OF COMPUTATIONAL SCIENCE AND ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2021)	McMaster University
(Computational Science and Engineering)	Hamilton, Ontario, Canada

TITLE:	Price Optimization in Stochastic Loss Systems
AUTHOR:	Seyyed Mohammad Hashemi Karouee B.Sc. (Industrial Engineering), Sharif University of Technology, Tehran, Iran
SUPERVISOR:	Dr. Yun Zhou
NUMBER OF PAGES:	xiv, 111

Abstract

Loss systems are of great importance for price optimization and revenue management even after more than a century since their first appearance. In this thesis, we analyze the optimal pricing problem for an M/M/1/1 Erlang-loss systems, and apply the model to inspect the impacts of vacancy tax regulations on short-term rental hosts. We then work on M/M/N/N loss systems while considering both advance reservation and multinomial logit (MNL) choice-model for the customers. We develop a simulation for this system and then train a machine learning (ML) model based on the outputs of this simulation to predict the utilization of each server based on different queueing parameters. Finally, we train another ML model for price optimization when the decision-maker sets the price for all servers to maximize the revenue of the whole system. We show that the presence of advance reservation decreases the utilization, consequently reducing the profit in the corresponding system.

To my parents, my sister, and my late aunt

Acknowledgements

To begin with, I would thank my family, who, despite being thousands of miles away from me, has helped me in many hard situations and always believed in me.

Furthermore, I would like to express my genuine appreciation to my supervisor, Dr. Yun Zhou, who has helped me through this challenging journey in the past two years.

Moreover, I would also like to thank the examination committee professors, Dr. Manish Verma and Dr. Ned Nedialkov, who did me the honor of reviewing my thesis and spending some time to read and give valuable suggestions for improving my current work.

Lastly, I would like to thank all the other department members and professors of Mathematics and DeGroote school of business who have arranged many valuable seminars, courses, and events during these two years of my Master's study.

Contents

List of Figures

List of Tables

Notation

Table of Notations

Notation	Definition
λ	the overall arrival rate
λ_{now}	the overall arrival rate of the type of customers who want to seek service for now
$\lambda_{reserve}$	the overall arrival rate of the type of customers who want to reserve for some time in future
N	the number of servers in Erlang-loss system
τ	the average number of days that the customers book in advance for the reservation

κ	the split rate which indicates what percentage of the customers want the service upon their arrival
μ	the service rate
ρ	the utilization rate
ρ_e	the effective utilization rate
θ	the amount of the threshold utilization
Δ	the amount of fixed tax to be paid in a year, Model II
δ	the amount of vacancy tax to be paid in a year, Model III
Π_i	the probability of the state that there are i people in the queueing system in the steady-state
P_S	the price offered in the system for using the service for one time unit
P_S^*	the price which leads to the optimal profit in Model I
π_{S_α}	the profit in Model I, benchmark model

π_{S_β}	the profit in Model II, regulation by a fixed amount of tax
π_{S_γ}	the profit in Model III, tax based on utilization
$\pi^*_{S_\alpha}$	the optimal profit of the server in Model I, benchmark model
$\pi^*_{S_\beta}$	the optimal profit of the server in Model II, regulation by a fixed amount of tax
$\pi^*_{S_\gamma}$	the optimal profit of the server in Model III, tax based on utilization
$P^*_{S_\alpha}$	the price which leads to the optimal profit in Model I
$P^*_{S_\beta}$	the price which leads to the optimal profit in Model II
$P^*_{S_\gamma}$	the price which leads to the optimal profit in Model III
$P^\theta_{S_\gamma}$	the threshold price which leads to the utilization equal to the threshold value in Model III
π_L	average yearly profit in the outside, long-term system
Y	number of days in a fixed period of time, in our case, 365

u	the fixed costs of every host in the short-term rentals
k	a binary variable for indicating whether a server needs to pay the vacancy tax
U_j^i	the utility gained by customer i from server j in MNL model
ϵ_j^i	unknown utility for alternative j in the view of customer i in MNL choice model

Table 1: List of all the notations used in this study

Chapter 1

Introduction

Price optimization problems are of great importance for managing many service systems. For example, restaurants, short-term rental platforms, and ride-hailing systems all need to set a reasonable price to achieve profits and customer satisfaction. In many cases, customers are impatient and will opt for alternative options if not served promptly. In this thesis, we study price optimization problems in the context of stochastic loss systems, in which customer demand is lost unless it is fulfilled (or guaranteed to be fulfilled) upon arrival.

There are two main motivations in our study that has shaped our thesis characteristics. In the following we will briefly describe each of them.

First, we are intrigued to study the impacts of tax policies on short-term rentals' pricing strategies. Recently, specific tax regulations have been levied in some countries, which have affected the decision-making of the hosts in short-term housing systems. On the other hand, it is important to see what the government's goals (or, in more general terms, the regulator) are by imposing these different types of tax. Therefore, we would like to determine how much control the government can have on the decision-making process by setting these tax policies over short-term rentals and analyzing how the hosts in short-term rentals will react to these regulations.

The second motivation is to make classic loss systems more realistic by adding several modifications. Therefore, we assumed that advance reservation is included in the traditional loss system; not all the customers will receive service upon their arrival. Some of them may reserve for some time in the future, which is common in real-world cases. On the other hand, in loss systems, the customer choice model is also an essential factor for customers' decision-making process. Therefore, we will consider a choice model that is popular in literature for our loss systems as well. As a result, we would like to see the impacts of adding these modifications to the optimal prices and, consequently, the optimal profits.

For the first part of the thesis, we will try to be more in-depth in the tax policies that we mentioned and after that, we will do some price analysis. As an introduction to those taxing regulations, we can mention British Columbia as one of the pioneers. There has been a Speculation and Vacancy tax policy to assist British Columbians in housing expenses as of 2018 (Government of British Columbia, 2018). Based on this regulation, if a home is not a principal residence, it needs to be rented for at least six months in a year to be exempted from paying the vacancy tax. Moreover, short-term units renting for less than a month will not be counted as those minimum six months of occupancy. As of 2019, the vacancy tax value is considered 2% of the assessed value of one's residential property if they are foreigners and 0.5% if they are Canadian (Government of British Columbia, 2020).

The aforementioned regulation will be applied in Toronto city as of January 2022 (Rocca, 2021). The tax rate will be equal to 1% of that corresponding home's current value. It is not yet known how many houses are vacant based on this rule; however, assuming only 1% of the homes are empty, the Toronto city could gain $55 to $65 million yearly. This indicates that setting this policy needs much prior study as it is dealing with a large value of money.

Similar regulation is happening in other countries as well. For example, in Hong Kong, the vacancy tax has been dictated with the same rule; if a unit is vacant more than 183 days a year, they are to pay the vacancy tax. Nevertheless, the tax value is 200% on the annual rental value of the house, and the target is developers, not home-owners. Australia has formulated a similar policy as of 2017 (Australian Government, 2021). As another example, in Malaysia, there was a proposal for developing vacancy tax on vacant houses for a specific period during a year, which is still on hold for further decision (Chong, 2020). Finally, Paris has started collecting tax on empty homes since 2015 and tripled the tax value to 60% worth of the unit in 2017 (Better Dwelling, 2017).

Besides the vacancy tax, there is another tax regulation, specifically in Ontario, Canada, called Municipal Accommodation Tax (MAT) which has to be paid by any short-term rental operator as of 2017 (Government of Ontario, 2017). The tax value is 4% of the rental revenue for any short-term unit under 28 consecutive days (City of Toronto, 2021). This could be a great concern for anyone who is deciding to share her unit in the short-term rental housing in Ontario.

Since most of the regulations mentioned earlier are recent and new, there has not been much study over these fields and the impacts of these policies on the decision-making process of the stakeholders in short-term rentals or the hosts in long-term housings. The first article that provides an evaluation only on vacancy tax (and not on MAT) is Segú (2020). She showed there was a 13% decrease in vacant units after formulating the vacancy tax in France in 1999, proving regulation's efficiency. She also mentioned that most of the vacant units turned into the principal residence after the regulation.

However, except for the above article, to the best of our knowledge, there is no other article that analyzes these two different types of regulation on the housing units in the form of queueing systems. In Chapter 3 of this thesis, we work on price optimization in short-term rentals as a loss system of M/M/1/1 with the single server in this system being a focal host operating accommodation service and compare those above two taxing policies with the case of no regulation. Then, we try to give some insights into the strategies for both the government and the hosts in short-term rentals.

For the second part of the thesis, based on the second motivation that we earlier mentioned, we start working on larger scale loss systems, e.g., M/M/N/N, however, with more realistic assumptions in our computations. In other words, we decided to explore the gaps that currently exist in revenue management of loss systems literature which prevent the model from being close to real-world problems. As an example, one of the extensions that we can add to our current loss system is the advance reservation.

Advance reservation is typical in many queueing systems. For example, in a short-term rental system such as Airbnb, a customer reserves her favorite unit (among the available ones) in advance for some time in future. Thus, she is assigned to a server without entering the system yet. Then, she needs to wait until the beginning of the reserved time; she will enter the system, stays there until her trip is over, and then she will leave. As another example, ride-sharing systems such as Uber let the riders reserve a ride 5 minutes to one month in advance (Uber Help, 2016). Lyft also has this option but for shorter periods, i.e., reserve for only seven days ahead (Lyft Help, 2017). Another example for reservation systems, as previously mentioned, can be car-sharing organizations such as Zipcar where customers can book a car from one hour up to 14 days in advance (Zipcar Support, 2021). This option lets customers (tourists and riders in our cases) more flexibility for their future planning, which is necessary for almost any queueing system. There have been numerous studies on queueing systems that support the users to reserve a server in advance.

A closely related paper is Chen *et al.* (2017), who worked on revenue management in loss networks while accepting advance reservations. One of the differences in the assumptions is that in this article, the goal is to set policies to either accept new customers to the system or not. In contrast, in our work, we assume customers are accepted unless all the servers in the system are occupied. Instead, we focused more on finding the optimal price for servers based on the demand function. In fact, we assume there is a third-party who is a decision-maker in the system and is willing to maximize the total revenue of the system. A similar to what Zhu *et al.* (2019a) did in their study. Moreover, we decided to use a choice-based model for the customers' selection process in our price optimization problem, i.e., multinomial logit choice model (MNL).

MNL defines a utility function for each customer based on each alternative (in our model, each server). Then, a customer will choose the alternative which leads to the highest utilization for her. Also, another advantage of this model is the random utility variable that is different for each option and makes the model more realistic by showing that the servers does not bring identical utility for each customer. Moreover, the use of MNL in revenue management problems is common in literature as Strauss *et al.* (2018) proposes a review of the choice-based models in the revenue management literature, including the MNL choice model.

Finally, motivated by the advance of machine learning (ML) techniques in recent years, we decided to use a simulation to create the queueing system environment to overcome the complexities in this modified loss system due to its intractability. Then, we trained a machine learning model based on the simulation to derive our numerical results and pricing policies in a shorter time. The usage of ML in price optimization has been increasing recently. An example could be the article by Helseth and Sveen (2020). They combined machine learning techniques to forecast prices in a price optimization problem, which inspired us to utilize a similar approach, i.e. using ML techniques to predict the effective utilization for our model.

To the best of our knowledge, there has not been any work combining ML techniques with price optimization in Erlang-loss systems while considering advance reservation and MNL choice models in the queueing system. We, therefore, decided to fill this gap by proposing our model with the following structure throughout our thesis.

Chapter 2 will provide a review of what similar works have been done so far in the literature. Chapter 3 will analyze the M/M/1/1 loss system while considering the two taxing policies for short-term rentals as an application. We will then give some valuable insights for the decision-makers in that specific context. Chapter 4 will work on a general case of M/M/N/N but with considering advance reservation and MNL choice model. We will also introduce our simulated queueing environment and compare the impacts of different parameters on the effective utilization of each server in a loss system. Chapter 5 will compare three proposed ML models to predict the modified Erlang-loss system's utilization and choose the best one among those. Chapter 6 will work on the price optimization in the aforementioned queueing system. Finally, Chapter 7 will be the conclusion and the suggestions for future work. The work breakdown structure has been illustrated in the next page:

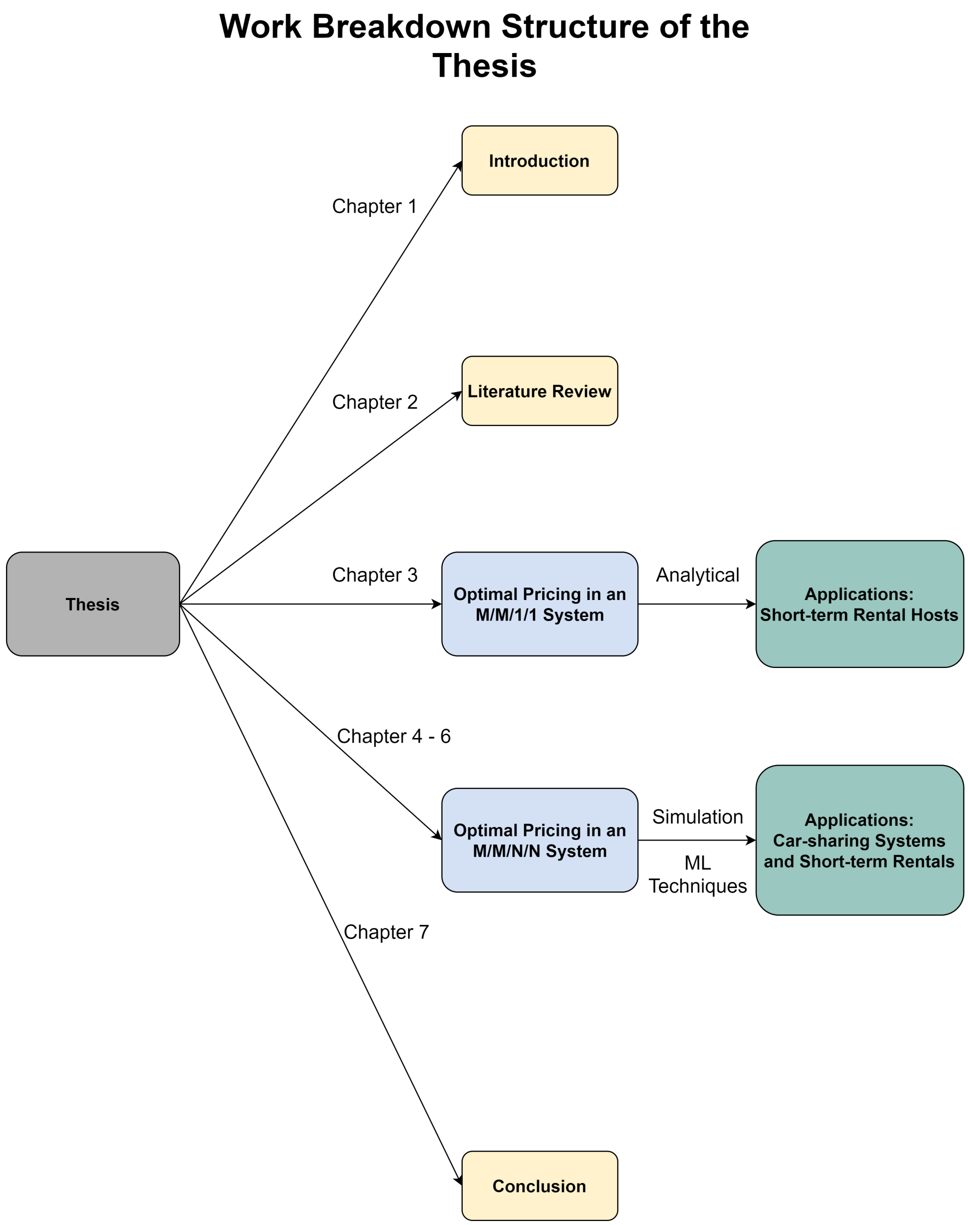


Figure 1.1: The structure of the thesis

Chapter 2

Literature Review

In this chapter, we classify the relevant literature into several categories; these domains are as the following. We first start with reviewing the literature for the first part, i.e., the first motivation of our thesis by defining the queueing theory principles, and then we will focus on the literature of loss systems among all the queueing systems. After that, we mention the applications of queueing systems to real-world problems in the literature. The next part will focus on the literature over the price optimization and revenue management in these kinds of systems. We then proceed to follow the literature based on our second motivation, starting with reviewing the related literature in advance reservation, and customer choice models. Finally, we will dig the literature on simulations and machine learning techniques.

Queueing Theory

Queueing systems are the types of systems in which servers provide service to customers seeking different service categories; customers enter the system, receive service and then leave. Then, there might be a time when servers are all occupied, and customers need to wait for some time to receive service; that is when queues are formed (Bose, 2013). Thus, the queueing theory will be defined as studying the characteristics of the waiting lines mentioned above to find the corresponding possibilities of different states of queues and then find the optimal value for queueing parameters (Thomopoulos, 2012). The parameters which are of interest to the analysts are customers arrival processes to the systems, the service process in the queueing system, the capacity of the system in terms of the number of servers, and the number of available waiting slots in queues (Bose, 2013). Hence, there are many types of queueing models based on different values for each of these parameters, which have led to the definition of Kendall's notations for queueing systems (Kendall, 1953). Based on Kendall's notation, the most common form is A/B/C/K/N/D which shows a system in which the arrival process is A, the service process follows B. Also, the number of servers is C, the number of queue capacity is K, the potential population is N, and the service discipline, e.g., the way different classes of customers are served in the system, is D (Kendall, 1953).

Loss Systems

Of the various types of queueing systems based on different queueing parameters, the M/M/N/N Erlang-loss queueing system is one of the most popular (Medhi, 2006) where customers' arrival pattern is the Poisson process. Moreover, the service time distribution is exponential. Devised by Erlang, a Danish mathematician motivated to work on the telephone network queues (Erlang, 1917), there have been many other works on this type of loss system afterwards (Medhi, 2002, Isguder and Uzunoglu-Kocer, 2014, and Shortle, J. F., Thompson, J. M., Gross, D., & Harris, 2018). For example, Palm (1943) proved that Erlang's model could be used for the cases when the arrival process follows any arbitrary distribution. Later, Brumelle (1978) showed that Erlang's model could be even extended to the systems with their arrival and service rates being state-dependent. Almost a century since the first time Erlang proposed his model, Kingman (2009) provided a history of achievements in Erlang's model and how it evolved in the area of queueing theory.

One of the reasons these kinds of queues are of significant importance is that they illustrate states in which there is customer loss. As Garnett, O., Mandelbaum, A., & Reiman (1998) referred to the customers in M/M/N/N queues as very impatient customers, if a customer arrives and all the servers are busy, there will be no queue, and she will leave. In other words, she has been *blocked* upon her arrival, which leads to the name of *blocking systems* as an alternative name for loss systems. Subsequently, the *effective* utilization of the servers in this system which is smaller than the nominal utilization due to the customer loss is essential for further analysis. Due to the importance of impatient customers in queueing theory, Wang *et al.* (2010) provided a review for these types of customers in the literature by first introducing different behaviours in impatient customers, then providing analysis of various queues with such customers and finally optimizing for both sides of the queue.

Applications of Queueing Systems to Real-World Problems

There are many contexts in which different models of queueing theory can be utilized. For example, there have been many works in healthcare systems that there is a literature review only for categorizing healthcare management problems related to queueing principles by their topics (Lakshmi, C., & Iyer, 2013). The goal in these papers is mainly to study the patients' waiting times in queues and improve the utilization, e.g., the percentage of the time when servers are not idle in a queueing system (Palvannan and Teow, 2012). As another example, Xiao and Zhang (2010) applied queueing theory techniques to improve the long queues in banking systems and successfully decreased the waiting time and increased customer satisfaction. Later, Cowdrey *et al.* (2018) also worked on the same topic but analyzed various queueing disciplines and determined that the shortest job first (SJF) discipline results in the best customer satisfaction. Moreover, transportation-related problems are also solvable with the techniques of queueing theory. Van Woensel and Vandaele (2007) provided a review for traffic flows with queueing theory approach including both finite and infinite buffer size (e.g., queue length) and concluded that with a sufficiently large enough queue capacity, both finite and infinite buffer sizes lead to almost the same results.

Price Optimization and Revenue Management

On the other hand, one of the other areas of great interest to the authors of operations management fields is price optimization and revenue management. Cross *et al.* (2011) mentions that price optimization led to increased profits for various types of companies without changing the customer types or system's products.

Furthermore, price optimization and revenue management techniques have been extensively studied in various contexts and systems. Vives *et al.* (2018) provided a comprehensive review on explaining these techniques and the current trends of research of revenue management in hotel systems. Guillet and Mohammed (2015) brought a critical review of the studies that have been done on revenue management in the hospitality and tourism industry. For revenue management in the car rental industry, Oliveira *et al.* (2017) presented a literature review. Tekin and Erol (2017) delivered a broad review study on price optimization and revenue management in retail stores. Finally, Ammirato *et al.* (2020) brought an in-depth literature review for revenue management in passenger transportation systems.

Based on the wide range of studies in both queueing systems and price optimization techniques, some have combined these two methods, and have done researches for price optimization in queueing systems. One of the pioneers in this topic is Naor (1969), who introduced a cost structure for M/M/1 queueing systems and optimized the corresponding revenue. Other noteworthy related works are Miller, B. L., & Buckman (1987), who considered capacity cost in M/M/N/N systems while proposing optimal pricing policies, and Mendelson, H., & Whang (1990), who categorized the classes of customers in an M/M/1 queueing setting while maximizing the expected profit for the system. Moreover, Ziya *et al.* (2006) worked on finding optimal prices under a specific price elasticity assumption for M/M/1/m and M/GI/N/N queues. The latter is a broader version of the Erlang-loss system accepting any independent general distribution for service time (including Markovian). Later he proved that the results could be extended for different classes of customers while assuming the price is static (Ziya *et al.*, 2008).

Further achievements have been reached in static pricing strategies in queueing systems, such as the works of Cachon and Feldman (2011), who tried to compare the profit gained from subscription payment of customers or per-use fees. Also, Haviv and Randhawa (2014) worked on the queueing systems in which setting price is done without knowing the demand information and concluded that demand-independent pricing works reasonably well. In more recent years, however, dynamic pricing in queueing systems attracted more attention. For instance, Kim and Randhawa (2018) focused on dynamic pricing with observable queues and price-sensitive customers, e.g., arrivals can see the waiting line and the price before deciding to join the queue and concluded that dynamic pricing outperforms static pricing. In contrast, some other works stated that dynamic pricing, in the long run, has negative impacts on customer demand (Ziya *et al.*, 2008). Nonetheless, despite these adverse effects, static pricing does pretty well, and the loss of profit compared to dynamic pricing is relatively small (Paschalidis and Tsitsiklis, 2000, Paschalidis, I. C., & Liu, 2002, and Hassin and Koshman, 2017).

More specifically, optimization for ride-sharing platforms in queueing systems is also of great significance. As an example, Jacob and Roet-Green (2021) developed a queueing model in ride-sharing platforms to compare the system's revenue when the customers are sharing their ride with other customers and when they are not doing so. The queueing system that Jacob and Roet-Green (2021) proposed for optimizing the profit was the Erlang-loss model of M/M/N/N. As another example, Afeche *et al.* (2018) studied the problem of matching customers and riders in ride-sharing platforms in loss network of the queueing system. In this study, they assumed there are two options of 1) being able to reject the demand of the customers despite the supply of the drivers, and 2) being able to reposition the drivers. Afeche *et al.* (2018) modeled the problem as the loss system and concluded that it might be optimal for the profit gained by the system to ignore the demands (customers) in the low-demand locations despite the high supply of the drivers and make them reposition to high-demand locations. These studies show that it is important to know the context of the queueing system prior to the problem modeling.

Advance Reservation

There have been some works about studying advance reservation in different systems. For example, Shajin *et al.* (2020) worked on advance reservation for queueing-inventory problems and later, they expanded their model by adding cancellation policies in their system (Shajin and Krishnamoorthy, 2021). Then they defined cost functions for the items they were to sold in their system. Moreover, the advance reservation was possible only among K specific time frames.

The impacts of the advance reservation have also been studied in ride-sharing systems. As an example, Bilali *et al.* (2019) compared the system in which riders can schedule a ride only for 2 minutes ahead with the system without the advance reservation option and showed a 30% increase in the chance of finding a shareable trip in the former. The increase in shareable trips is advantageous for the system as it influences the price of the ride, and hence, it will win in the market competition, which will lead to a higher profit subsequently.

Customer Choice Models

In queueing systems, one of the other essential factors to consider, as previously mentioned, is the customer choice, e.g., how the customers are going to choose among the servers upon their arrival. As one of the most recent works, we can mention Pender *et al.* (2020), who studied the impacts of the delayed knowledge of the queue length for the customers while considering the MNL model as the customer choice model. They concluded that the delay value has some threshold that needs to be calculated based on different queueing contexts and that policies should be based on that specific threshold. As another study considering the MNL choice model, we can refer to Farhoodi (2019). He studied the welfare distribution in different systems such as ride-sharing and short-term rental housings with the MNL customer choice model. He then showed that in the former, the location and time of the service, and in the latter, the location, time, and the characteristics of a unit are the key factors in shaping the surplus values. These studies show that customer choice is valuable for accounting for the servers' heterogeneities.

Simulations and Machine Learning Techniques

Since we deal with random systems and queues, we may utilize methods to analyze and solve them more easily. Based on Jain (1990), discrete-event simulation (DES) has been a powerful tool for studying dynamic systems which work in random. Specifically, for simulating queues for further analysis, DES has been extensively utilized (Insua *et al.*, 2012). Furthermore, the use of discrete-event simulation has increased in different areas of study, such as in health systems, as Jun *et al.* (1999) suggests in their review of simulation applications in health care clinics. As another example, Wang, Y. B., Qian, C., & Cao (2010) utilized simulation to optimize an M/M/c queue in the banking systems. Later, he mentioned that the results could be extended to other queueing contexts such as selling tickets or hospital wait-rooms. Moreover, Kambli *et al.* (2020) used discrete-event simulation to reallocate the existing servers (and not just expand the current workforce) to decrease the queue lengths and reduce the waiting time, hence increasing customer satisfaction.

Moreover, simulation has also helped to solve and analyze price optimization problems. As in (Mariello *et al.*, 2020), they have studied the different pricing policies in hotel systems. They have also considered cancellation and reservation in their simulation to make the model more realistic. With simulation, they reached a %19 increase in revenue in comparison to the original pricing policies. Petricek *et al.* (2021) studied the same general idea.

One of the common methods of discrete-event simulation is using the SimPy library in Python (SimPy 4.0.1 Documentation, 2020). There is also a more detailed introduction to discrete-event simulation and SimPy in Matloff (2008). Recent works have also been optimizing queues in SimPy, such as Holden (2017), who studied inventory optimization in SimPy or Corgozinho *et al.* (2018), who studied the scaling of hospital queues, specifically in Brazil. They concluded that the simulation helps the user better understand the queueing system they are working on and aid them in making improvements such as reallocating the servers.

Another way of overcoming the heavy computations in mathematical models is using machine learning (ML) techniques. Recently, the studies of price optimization with these algorithms have vastly increased. Some of these studies focus on the price prediction. For example, works such as Greenstein-Messica and Rokach (2020) used ML methods to forecast product price elasticity effects for the products that there is no historical information regarding their price elasticity. They used the gradient boosting method based on the data derived from 18 months in a European online store and showed that their proposed method greatly improved accuracy in forecasting. Others focus on predicting the solutions to large-scale problems instead of directly solving them. For example, Abbasi *et al.* (2020) trained four popular ML models, i.e., CART, kNN, RF, and MLP artificial neural network, based on a limited number of optimization problems with their optimal solution. They showed that the trained model works well in the decision-making process of blood supply chain systems.

There have been many reviews for ML models in the literature. One of the recent ones is Ray (2019), which gives insight into the most popular methods and compares their performance based on different parameters. This study will help a reader first be familiar with the general ideas of each of these methods and second, help them identify which algorithm to choose when facing a new problem.

Finally, there have also been studies regarding how to improve ML methods, namely hyperparameter tuning. Bergstra and Bengio (2012) explained the random search for hyperparameters in an ML model, a technique mainly used to optimize ML trained models' performance.

Chapter 3

Price Optimization under Utilization Constraint: M/M/1/1 Erlang-Loss Model

In this chapter, we work on a price optimization problem for an M/M/1/1 loss system. We first do the optimization without any penalties for the server given a demand rate for the system; then, we will study two possible tax policies that can be utilized by a regulator to manipulate the pricing strategies of the system. A related real-world case would be the queueing system in short-term rentals; however, we can generalize the conclusions to other similar contexts.

We start this chapter by first describing the benchmark problem without any constraints on utilization; we then state the two possible scenarios for dictating taxes, referring to them as Model I (benchmark model), Model II, and Model III, respectively. Finally, we will compare the results in these three models and analyze the decision-making process by both the server in the loss system and the regulator.

3.1 Assumptions

The decision-makers in all three models are as the following: **1) Server** - the server needs to decide if she has to enter the M/M/1/1 loss system and offer short-term service (a system with more uncertainty in its demand and with a more potential profit; will be referred to as short-term system) or choose the outside alternative and offer long-term service (a more stable system but with a less potential profit; will be referred to as long-term system) to maximize her profit; from now on, we will use the pronoun of *she/her* for servers, **2) Regulator** - the regulator controls the tax policies and regulation; the municipal accommodation tax, e.g., the fixed value tax that all the servers have to pay in Model II, and the vacancy tax, e.g., the tax that has to be paid if the server is vacant more than a specific threshold in Model III, **3) Outside Alternative (exogenous)** - suppose that each server's average yearly income is π_L if she chooses the outside alternative and provides service there, **4) Customers** - they arrive at the system with the arrival rate of λ and we assume it decreases linearly in the price; inspired by works such as Pedro Aznar *et al.* (2019), Zhu *et al.* (2019a), and Farhoodi (2019) where they assumed the demand or the utility of each customer linearly reduces when price increases. Thus, the normalized arrival rate, e.g., the demand of the customers seeking service will be $\lambda = 1 - P_S$ where P_S is the offered price by the server in the loss system for using the service for one time unit.

3.2 Model I (Benchmark): No Regulation

As mentioned earlier, we are dealing with an M/M/1/1 loss system where M refers to the exponential distribution of service time and the Poisson process of customer arrivals. Moreover, the first 1 refers to the number of servers and the second 1 states the maximum capacity of the system. It conveys the meaning that if the server is occupied, and a new customer has just arrived, she will leave the system. The following figure illustrates the state of the system in which there is already a customer in the system:

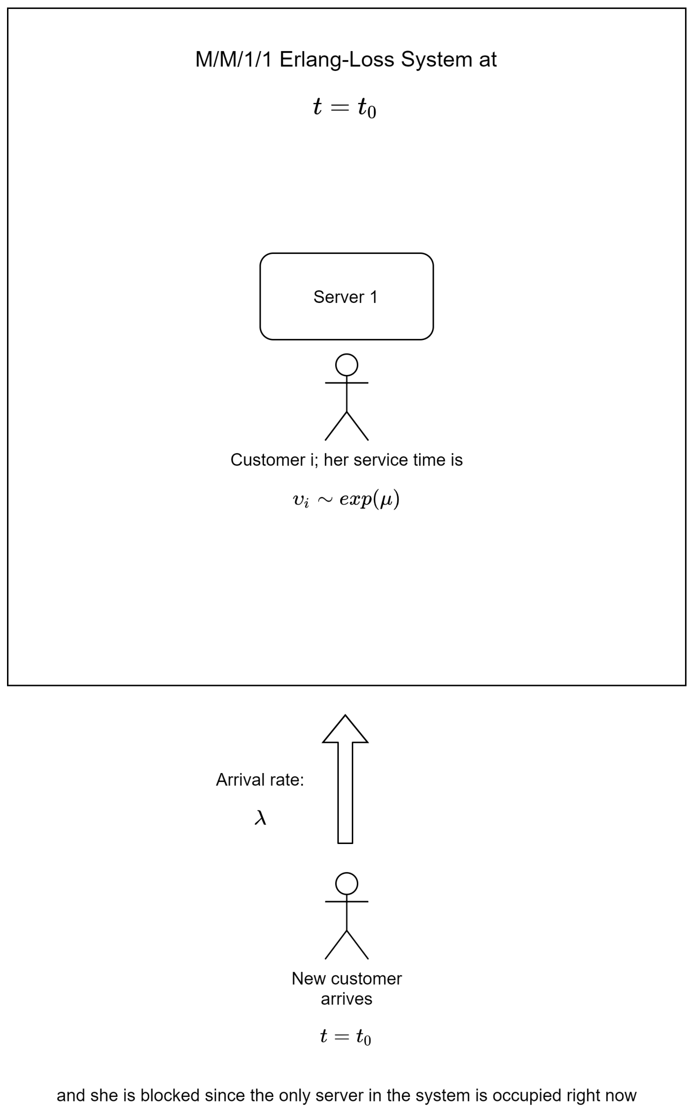


Figure 3.1: Π_1 state in M/M/1/1 loss system

As can be seen, the customer has been blocked since the server is already busy; therefore, she will leave the system and we will have a customer loss at $t = t_0$.

For the next step, we need to define the profit function of the server. Therefore, the decision variable will be the offered price by the server and the parameters that are going to affect the profit value are the arrival rate of customers, the service rate of the server, and the fixed costs. Hence, the annual profit will be defined as:

$$\pi_{S_\alpha} = Y \rho_e P_S - u \tag{3.1}$$

Where Y is a constant that could be equal to 365, the number of days in a year (note that the arrival and service rates are both defined as people per day). However, without loss of generality, we assume $Y = 1$. Moreover, the index α shows this variable is related to Model I (this is for making the future comparison of different models more convenient). Similarly, the corresponding indices for Model II and Model III will be β and γ. Furthermore, u is the fixed costs incurred by the system during a fixed period of time, i.e., a year. If we assume the system as a short-term rental, these costs include, but are not limited to, cleaning fees, service fees and the fees for providing some essential supplies and amenities for tourists. As a result, the average value for these costs should be deducted from the profit of the host in our system. Finally, throughout the thesis, we assume that $u < \pi_L$, where π_L is the average yearly income of the outside alternative. Finally, ρ_e is the effective utilization, e.g., the percentage of the time that the unit is occupied, which is smaller than the nominal utilization ($\rho = \frac{\lambda}{\mu}$). In fact, effective utilization considers the probability of customer loss which based can be derived as below (see, e.g., equation A.2 in Dunlop *et al.*, 1999):

$$\rho_e = \frac{\lambda}{\lambda + \mu} \tag{3.2}$$

Based on what have been said, the modified profit function will be as follows:

$$\pi_{S_\alpha} = \frac{\lambda}{\lambda + \mu} P_S - u \rightarrow \pi_{S_\alpha} = \frac{P_S - {P_S}^2}{1 - P_S + \mu} - u \tag{3.3}$$

Which is a quadratic function of P_S. The following lemma will find the best price to offer by the host to gain the maximum profit:

Lemma 3.1. *In the M/M/1/1 loss system when there is no regulation, the profit function is concave and it has one maximum over the feasible values of the offered price; that happens when the server chooses* $P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu\,(\mu + 1)}$ *to offer. The optimal profit (the profit based on* $P^*_{S_\alpha}$*) will be* $\pi^*_{S_\alpha} = 2\mu + 1 - 2\sqrt{\mu\,(\mu + 1)} - u$*.*

Proof. Please see the appendix. □

Knowing the result of the above lemma, it will be easy for a server to decide whether to participate in the short-term system or opt for the long-term one. She needs to compare $\pi^*_{S_\alpha}$ with π_L and choose the one which gives her the higher profit. Based on different values of u for different hosts, one may choose the former, and the other might prefer the latter.

In the next section, we will work on Model II where the server who enters the short-term system needs to pay some extra money as the tax.

3.3 Model II: Regulation by a Fixed Amount of Tax

This section will be dedicated to the MAT policy that have been formulated in Ontario as mentioned earlier, the type of tax that every short-term rental host must pay. Although, based on this regulation, the tax value has to be equal to 4% of the revenue that the host gains (City of Toronto, 2021), since we are dealing with an M/M/1/1 system, we are only having one server; therefore, we can assume that the value of tax can be constant. In other words, what is more important in this model is that the server (host) has to pay some value of money as a tax regardless of the utilization of the system. Furthermore, most of the assumptions and calculations will be the same as the previous section, with a slight difference in the definition of the profit function. We assume that the amount of tax that should be deducted from the profit of the server is denoted by Δ in this model. Hence, we can say:

$$\pi_{S_\beta} = Y\rho_e P_S - u - \Delta \rightarrow \pi_{S_\beta} = \pi_{S_\alpha} - \Delta \tag{3.4}$$

Based on the above equation, and the fact that Δ is a constant parameter, we define the following lemma:

Lemma 3.2. *In the M/M/1/1 loss system when there is regulation by a fixed amount of tax, the profit function is concave and it has one maximum over the feasible values of the offered price; that happens when the server chooses* $P^*_{S_\beta} = P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu(\mu+1)}$ *to offer. The optimal profit (the profit based on* $P^*_{S_\beta}$*) will be* $\pi^*_{S_\beta} = 2\mu + 1 - 2\sqrt{\mu(\mu+1)} - u - \Delta$ *or in other words, it will be* $\pi^*_{S_\beta} = \pi^*_{S_\alpha} - \Delta$.

Proof. Please see the appendix. □

The same as the previous section, the server needs to compare $\pi^*_{S_\beta}$ with π_L and choose the better one.

3.4 Model III: Tax Based on Utilization

In this section, we work on the vacancy tax policy, the one which has been formulated in several countries such as Canada (BC), Hong Kong and Australia stating that a host should pay the tax if the unit is vacant more than some specific time in a year. Contrary to the previous two sections, the model is more complicated. As mentioned earlier, the tax has to be paid, only if the utilization is below a certain threshold, θ, stated by the regulator. Thus, we start by modifying the profit function:

$$\pi_{S_\gamma} = Y\rho_e P_S - u - k\delta \rightarrow \pi_{S_\gamma} = \pi_{S_\alpha} - u - k\delta \tag{3.5}$$

Where k is a binary variable that will be equal to 1 if the utilization is below the threshold and will remain zero otherwise, and δ is the amount of tax that has to be paid in this policy.

Contrary to the previous model, $P^*_{S_\alpha}$ does not necessarily lead to the best possible profit since we have a discontinuity caused by δ in the π_{S_γ}. In other words, since the utilization decreases by the increase in price (when price increases, the arrival or demand rate will decrease and consequently, utilization will also decrease), there will be a specific price which will lead to a utilization exactly equal to the threshold. We call this point the *threshold price*, and we denote it by $P^\theta_{S_\gamma}$; the price that if the host offers, her unit's utilization will be equal to θ. Note that in this state, the host still does not need to pay the tax. However, if she offers $P_S = P^\theta_{S_\gamma} + \epsilon$ for any positive value of ϵ, she will need to pay the tax.

Based on the above discussion, we will reach the following lemma:

Lemma 3.3. *In the M/M/1/1 loss system when there is a tax based on utilization, if a server is willing not to pay the tax, she should set her price in a way that* $P_S \leq P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}$

Proof. Please see the appendix. □

Moreover, we call the region in which the utilization is equal to or above the threshold value the *untaxed region*.

Now, we define a maximization problem for finding the optimal value of profit gained by a server in the short-term rental system:

$$
\begin{aligned}
max \quad & \pi_{S_\gamma} = \left(\frac{1-P_S}{\mu+1-P_S}\right) P_S - u - k\delta \\
\\
s.t.: \quad & P_S - P^{\theta}_{S_\gamma} \leq kM \\
& P_S \geq 0 \\
& k \in \{0, 1\}
\end{aligned}
\tag{3.6}
$$

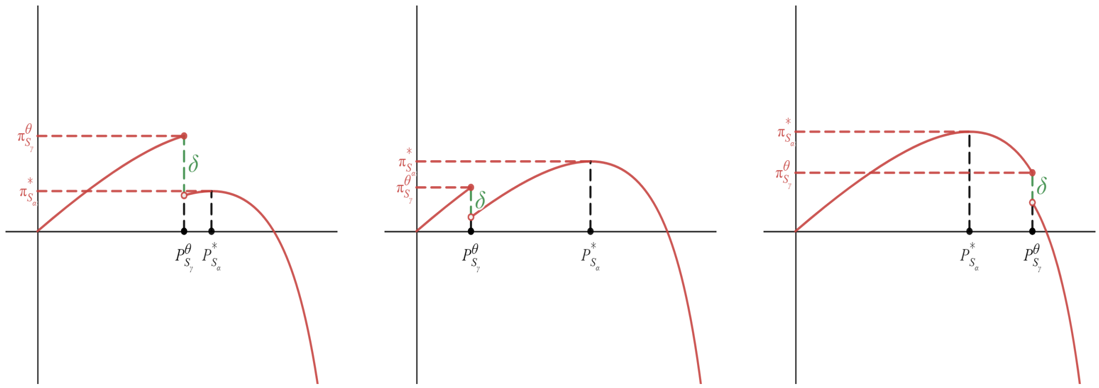
In the objective function, $\frac{1-P_S}{\mu+1-P_S}$ comes from equation (3.2) that we have replaced the value of λ with $1-P_S$. The first constraint comes from *Lemma 3.3*. As a reminder, k is a binary variable which indicates whether the utilization is above or below the threshold. The first constraint, assuming M is a big positive constant, means that whenever P_S is smaller than or equal to $P^{\theta}_{S_\gamma}$ (the conditions in *Lemma 3.3*), k could be both 0 or 1 because the left-hand side will be a negative number and the right-hand side will be 0 or M, respectively. In both situations, the constraint will be met; nevertheless, since we are dealing with a maximization problem, $k = 0$ will always be the better choice since by choosing this value, we are removing the term $-k\delta$ from the objective function. In other words, this means that the host does not need to pay the tax in this case. On the other hand, if P_S is bigger than $P^{\theta}_{S_\gamma}$, k can only be equal to 1 which will result in paying the tax by the host in the objective function.

The following lemma will be a solution to the above optimization problem under certain circumstances:

Lemma 3.4. *In the M/M/1/1 loss system when there is a tax based on utilization, if the utilization threshold is smaller than or equal to* $1-\sqrt{\frac{\mu}{\mu+1}}$*, the best strategy will be offering* $P^*_{S_\gamma} = P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu\,(\mu+1)}$ *to gain the maximum profit. Under these conditions, the server does not need to pay the vacancy tax either. However, if* $\theta > 1-\sqrt{\frac{\mu}{\mu+1}}$ *then* $P^*_{S_\alpha} > P^{\theta}_{S_\gamma}$.

Proof. Please see the appendix. □

According to the above lemma, when we are no longer under the conditions of *Lemma 3.4*, e.g., the utilization threshold is larger than $1-\sqrt{\frac{\mu}{\mu+1}}$, based on the results of this lemma, $P^*_{S_\alpha} > P^{\theta}_{S_\gamma}$. Nonetheless, if the server persists in offering $P^*_{S_\alpha}$, based on *Lemma 3.3*, she needs to pay the tax. Therefore, if the host is still willing to escape from paying the tax, she should offer a smaller price (the lower the price is, the higher the utilization will be). Previously, we mentioned that the profit function in Model I is concave (*Lemma 3.1*). In this model, however, there is a discontinuity in the profit function, and it happens when the utilization gets equal to the threshold value. Based on this fact, there may be three different scenarios which have been depicted below:



(a) $P^{\theta}_{S_\gamma} \leq P^{*}_{S_\alpha}$, $\pi^{\theta}_{S_\gamma} \geq \pi^{*}_{S_\alpha}$ (b) $P^{\theta}_{S_\gamma} < P^{*}_{S_\alpha}$, $\pi^{\theta}_{S_\gamma} < \pi^{*}_{S_\alpha}$ (c) $P^{\theta}_{S_\gamma} > P^{*}_{S_\alpha}$

Figure 3.2: The three different states that can happen for the server; based on the values of δ and θ, sometimes paying the tax still leads to a higher profit. On the other hand, whenever $P^{\theta}_{S_\gamma}$ is greater than $P^{*}_{S_\alpha}$, it will always be better for the server to offer $P^{*}_{S_\alpha}$

As can be seen in the above figure, when the tax constraint is added to the profit function, it will not affect the overall behavior of the profit function and only creates a gap in the discontinuity point (when $\rho_e = \theta$); the reason is that δ, the same as Δ, is a constant parameter.

Based on what has been said, we may notice that a server can have two strategies if she is not in the conditions of *Lemma 3.4*:

- **Strategy 1**: offer $P^{\theta}_{S_\gamma}$ in order to escape from paying the tax by offering a lower price

- **Strategy 2**: offer $P^{*}_{S_\alpha}$ while not considering the tax penalty

Thus, according to the above strategies, we have to find out when $\pi^{*}_{S_\alpha}$ will be higher than $\pi^{\theta}_{S_\gamma}$ despite paying the vacancy in the former one. For this purpose, we have to compare the following values:

$$\begin{aligned}
&\pi^{\theta}_{S_\gamma} < \pi^{*}_{S_\alpha} \Leftrightarrow \rho^{\theta}_{e_\gamma} P^{\theta}_{S_\gamma} - u < \rho^{*}_{e_\alpha} P^{*}_{S_\alpha} - u - \delta \Leftrightarrow \left(\tfrac{\rho^{\theta}_{e_\gamma}}{1+\rho^{\theta}_{e_\gamma}}\right) P^{\theta}_{S_\gamma} < \left(\tfrac{\rho^{*}_{e_\alpha}}{1+\rho^{*}_{e_\alpha}}\right) P^{*}_{S_\alpha} - \delta \\
&\Leftrightarrow \theta \left(\tfrac{1-\theta(1+\mu)}{1-\theta}\right) < \left(\mu + 1 - \sqrt{\mu(\mu+1)} - \sqrt{\mu(\mu+1)} + \mu\right) - \delta \\
&\Leftrightarrow \delta < \left(2\mu + 1 - 2\sqrt{\mu(\mu+1)} - \tfrac{\theta - \theta^2(1+\mu)}{1-\theta}\right)
\end{aligned} \tag{3.7}$$

Now, we can define the following lemma with the above inequality:

Lemma 3.5. *In the single-host model when there is a tax based on utilization, if the utilization threshold is larger than* $1 - \sqrt{\frac{\mu}{\mu+1}}$*: if the condition (3.7) is met, Strategy 2 is the winner. Otherwise, Strategy 1 is the winner.*

Proof. Please see the appendix. □

Finally, using *Lemma 3.4* and *Lemma 3.5*, we can define the following proposition:

Proposition 3.1. *In the M/M/1/1 loss system when there is a tax based on utilization, the optimal price function (e.g., the price which leads to the best possible profit) and the best possible profit gained by the server based on different values of* θ *and* δ *will be as below:*

The optimal price function:

$$P^{*}_{S_\gamma} = \begin{cases} P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}, & \theta > 1 - \sqrt{\frac{\mu}{\mu+1}}, \quad \textit{and} \\ & \delta > 2\mu + 1 - 2\sqrt{\mu(\mu+1)} - \frac{\theta-\theta^2(1+\mu)}{1-\theta} \\ P^{*}_{S_\alpha} = \mu + 1 - \sqrt{\mu(\mu+1)}, & \textit{otherwise} \end{cases} \tag{3.8}$$

The best possible profit gained by the server:

$$\pi^*_{S_\gamma} = \begin{cases} 2\mu + 1 - 2\sqrt{\mu\,(\mu+1)} - u, & \theta \leq 1 - \sqrt{\frac{\mu}{\mu+1}} \\ \pi', & \theta > 1 - \sqrt{\frac{\mu}{\mu+1}} \end{cases} \tag{3.9}$$

where:

$$\pi' = \begin{cases} 2\mu + 1 - 2\sqrt{\mu\,(\mu+1)} - u - \delta, & \delta \leq 2\mu + 1 - 2\sqrt{\mu\,(\mu+1)} - \frac{\theta - \theta^2(1+\mu)}{1-\theta} \\ \theta\left(\frac{1-\theta(1+\mu)}{1-\theta}\right) - u, & \delta > 2\mu + 1 - 2\sqrt{\mu\,(\mu+1)} - \frac{\theta - \theta^2(1+\mu)}{1-\theta} \end{cases}$$

Proof. Please see the appendix. □

3.5 Impacts of the Tax Payment: An Application to Short-Term Rentals

In this section, we analyze the effects of paying tax by the servers on their pricing strategies. Moreover, for the next following two sections, we assume that the system we are analyzing the decisions for is the short-term rentals system which can be later generalized to other similar systems. Subsequently, the server in the system will be referred to as the *host*, the customers will be referred to as the *tourists*, the service rate will be equal to the length of the stay in the system, the tax in Model III will be referred to as the *vacancy tax*, and finally, the regulator will be referred to as the *government*.

3.5.1 Model II: Tax Sensitivity Analysis

For this model, since paying the tax is mandatory for every participant in the short-term rental system, the hosts cannot change their strategy in a way to improve their profit; even if they offer their price for free, they need to pay the tax. Therefore, the pricing strategies in Model II are the same as Model I, and the only difference is that the hosts are going to gain less profit in the former. On the other hand, the government can act strictly and set Δ in a way to make $\pi^*_{S_\beta}$ less than π_L. Under these circumstances, the hosts will opt for long-term rentals. Consider the following figure:

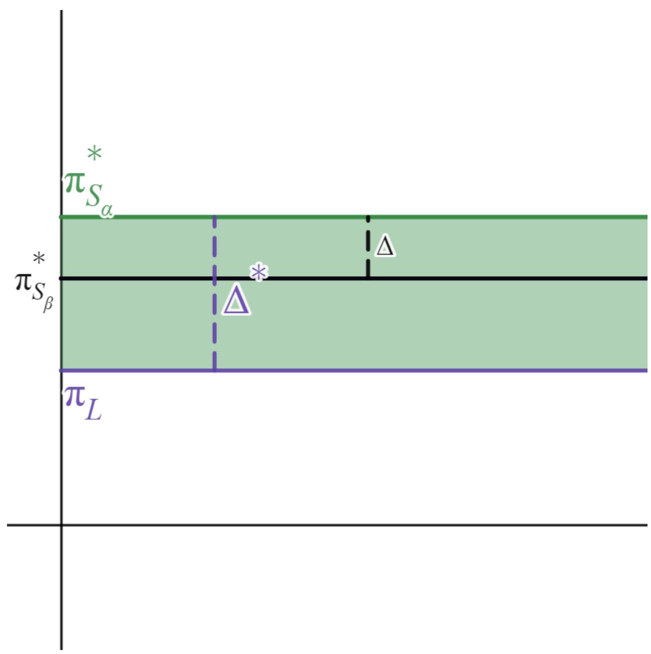


Figure 3.3: Tax analysis in Model II, Single Host

According to Figure 3.3, the difference between $\pi^*_{S_\alpha}$ and π_L, the green region, is the range for the tax value, called Δ^*, that the host will still choose the short-term rental, but she will have less profit than $\pi^*_{S_\alpha}$. As an example, if the tax amount is

Δ (the black dotted line), the corresponding best profit will be $\pi^*_{S_\beta}$ which is equal to $\pi^*_{S_\alpha} - \Delta$. Thus, when $\Delta < \Delta^*$, the hosts will remain in the short-term rentals, and when $\Delta^* \geq \Delta$, they will prefer long-term rentals. The disadvantage with this model is that the government does not have any control over changing the prices; hosts either enter the short-term rentals and offer $P^*_{S_\beta} = P^*_{S_\alpha}$ or do not enter the short-term rentals. Not being able to change the prices with this policy means not being able to change nor control the demand of tourists for the short-term rentals. In fact, this policy is more for controlling the supply of long-term rentals rather than controlling the demand for hotels and short-term rental systems. Nevertheless, if the government is too strict in this model, it will reduce the short-term supply. Therefore, it will decrease the competition for hotels as the most crucial competitor of hotel industries are short-term rentals, and their number have been decreased now.

3.5.2 Model III: Tax Sensitivity Analysis

In Model III, however, as can been seen in Figure 3.2, usually, the offered price by the host should be lowered if a host wants to gain the maximum profit from the short-term rentals system. Nonetheless, as we have noticed, for some specific values of θ, e.g., the utilization threshold for paying the vacancy tax, and δ, e.g., the tax amount in Model III, despite the existence of the vacancy tax penalty, the best strategy of the host will not differ from the case where there is not a penalty for the vacancy of the units (when the corresponding scenario to Figure 3.2c happens, similar to Model I). This means that a host will not need to reduce her offered price in order to set the utilization above the threshold as it is already so. Sometimes, even the maximum profit gained by that host will not decrease despite the vacancy tax limitation.

Based on these facts, we have named three different possible outcomes in Model III based on values of δ and θ: 1) *Effective Tax Region with Price Change*, 2) *Effective Tax Region without Price Change*, and 3) *Ineffective Tax Region*; all of which have been illustrated in Figure 3.4.

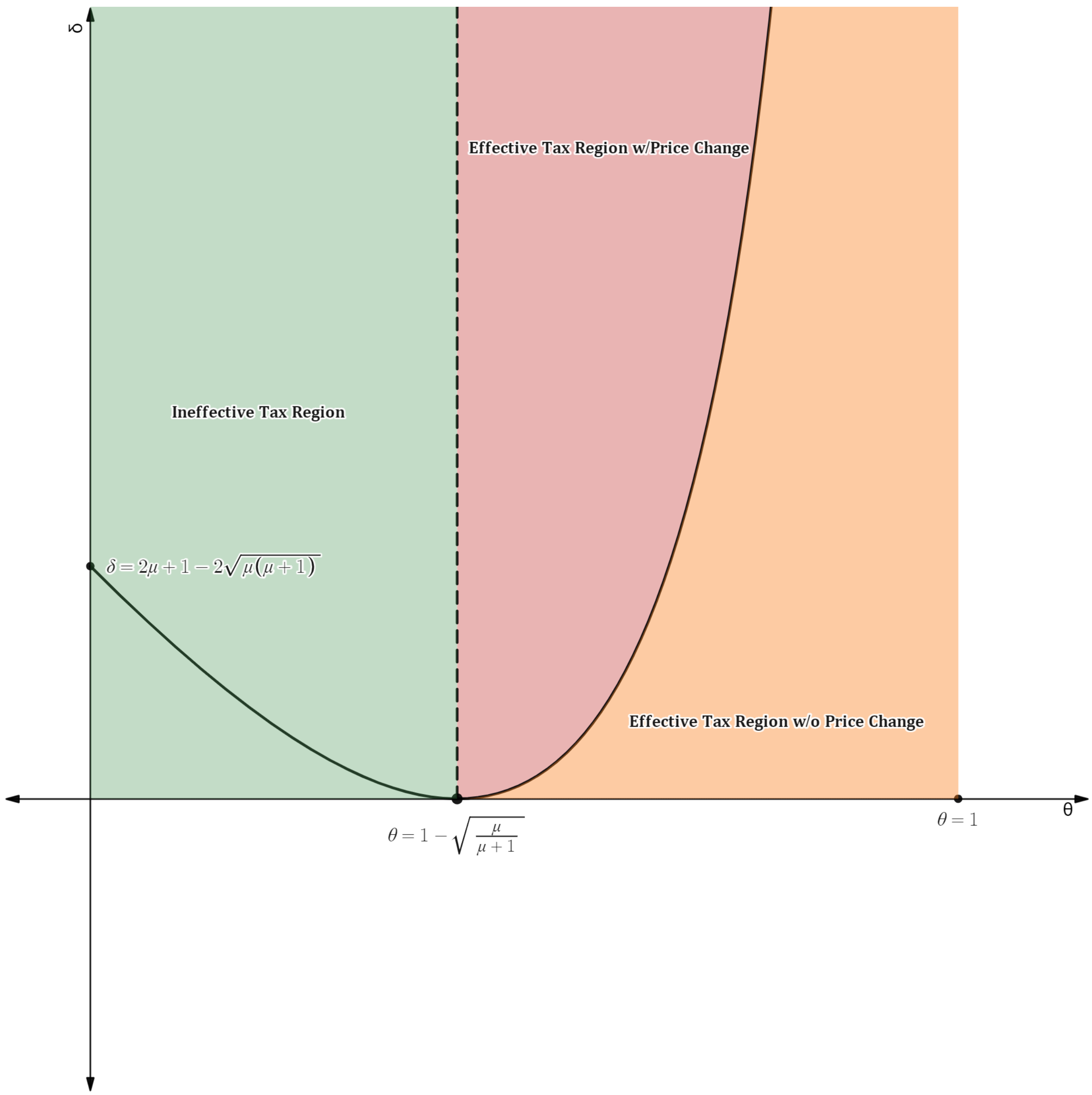


Figure 3.4: Various tax regions based on different values of θ, the utilization threshold and δ, the tax amount in Model III

According to the above graph, we have plotted $\delta = 2\mu + 1 - 2\sqrt{\mu(\mu+1)} - \frac{\theta}{1-\theta}(1 - \theta(1+\mu))$ which comes from the second condition of choosing $P_{S_\gamma}^{\theta}$ in equation

(3.8) while the x-axis is θ and the y-axis is δ. Moreover, the intersection of the graph with each axis has been calculated. By increasing the value of μ, the dotted black line will move towards the left-hand side meaning the effective tax region will expand which is in accordance to our previous observations; the shorter the average length of stay for tourists, the lower the utilization will be and in return, the easier it will be for the government to make its tax penalty effective. Finally, three different colors illustrate the three different potential policies which are going to be described in the following.

Effective Tax Region w/Price Change

If the goal of the government is to reduce the number of host(s) in the short-term rentals, or in other words, the government wants to channel supply in the short-term rental market to the long-term housing market, it should define the values of θ and δ as the following:

$$\begin{cases} \theta > 1 - \sqrt{\frac{\mu}{\mu+1}} \\ \delta > 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta}{1-\theta}\left(1 - \theta\left(1+\mu\right)\right) \end{cases} \tag{3.10}$$

Which refers to the red region in Figure 3.4 and its corresponding scenario in the host's point of view will be Figure 3.2a; in other words, the host, in this situation, even with the best possible strategy, is going to gain less profit than the time there is not a vacancy tax penalty (since we already showed that the profit function in Model I (without tax) is concave and has only one maximum point; thus, reducing a positive value of δ from this function will lead to a lower profit than $\pi^*_{S_\alpha}$). In this state, based on the value of π_L, the average yearly income in the long-term rentals, and the value of u, the average fixed costs of staying in the short-term rentals, one host may compare the value of $\pi_{S_\gamma}^{\theta}$ with π_L, and if the fixed costs are in a way that they make the profit in short-term rentals lower than the long-term rentals, the host will leave the former and utilize her unit in the latter. In this way, the government has reached its goal as it wanted to reduce the number of short-term rentals' hosts.

The other outcome for this policy is that since hosts have to reduce their offered price inevitably, from $P_{S_\alpha}^{*}$ to $P_{S_\gamma}^{\theta}$, based on the demand rate $\lambda = 1 - P_S$, the demand for short-term rentals will increase. In other words, tourists will be more encouraged to use short-term rentals. Thus, for hotels, it reduces competition from the peer-to-peer (P2P) rental market to a less extent, if at all. Nonetheless, the short-term rentals supply may decrease if for some host $\pi_{S_\gamma}^{\theta} < \pi_L$, and in return, the long-term rentals supply may increase.

Effective Tax Region w/o Price Change

The other, however less strict, policy for the government will be setting the values of θ and δ as the following:

$$\begin{cases} \theta > 1 - \sqrt{\frac{\mu}{\mu+1}} \\ \delta \leq 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta}{1-\theta}\left(1 - \theta\left(1+\mu\right)\right) \end{cases} \tag{3.11}$$

Which refers to the orange region in Figure 3.4 and its corresponding scenario in the host's point of view will be Figure 3.2b. The reason why we use the term *less strict* (or a *moderate*) tax condition is that the value of δ is going to be less than before this time; nevertheless, it will still lead to a profit less than the time there is no vacancy penalty (Model I). With this policy, since P_S will not change, the demand rate for the short-term rentals will not differ; thus, there will not be any specific encouragement for the tourists to join the system than when there were no tax penalties. However, the government is gaining some money from those hosts who decided to stay in the system despite paying vacancy taxes. Moreover, the chances of leaving short-term rentals for a host and opting for long-term rentals will be less than the previous policy as $\pi^*_{S_\alpha} > \pi^\theta_{S_\gamma}$; this means that hosts' profit in this state will be more than the previous policy. Hence, when a host wants to compare her profit in the short-term rentals with the average yearly income in the long-term rentals, she is more likely to stay in the short-term rentals than in the previous case. Therefore, there will be an increase in the supply of long-term rentals (but less increase in comparison to the previous policy).

As a result, with this policy, the regulator (e.g., the government) is increasing the long-term supply while also trying to protect the hotel industry from the peer-to-peer rental platforms by receiving vacancy tax from those P2P platforms who are still willing to stay in those systems.

Ineffective Tax Region

If the government chooses any threshold less than $1 - \sqrt{\frac{\mu}{\mu+1}}$, regardless of the value of δ, its policy is going to be ineffective as the threshold is not high enough to make the hosts reduce their price; in other words, the system's utilization is already above the threshold value; thus, setting θ below $1-\sqrt{\frac{\mu}{\mu+1}}$ will be useless. In Figure 3.4, this policy will lie in the green region; furthermore, in the host's perspective, this state will be Figure 3.2c.

This policy will have no effect on neither tourists nor the hosts' decision-making process as everything remains the same as the case when there is no vacancy tax penalty. Nevertheless, if we accept the fact that not all the hosts are going to act professionally as most of the units are usually managed by regular hosts lacking pricing tools to get help from based on (Hill, 2015), some hosts may reduce their offered price and set some smaller P_S than the time there were no vacancy tax penalties.

In the next chapter, we will try to work on larger scale, modified Erlang-loss systems, and find the utilization and comparison under different situations

3.5.3 Comparison of Models

As mentioned earlier, the pricing strategies in Model II are the same as Model I except for the last decision of either staying in the short-term rentals or the long-term rentals. For Model III, assuming $\delta = \Delta$, e.g., assuming the tax values are equal, when the government chooses the *effective tax without price change* policy, the effects on the pricing strategies of the hosts will be the same as the ones in Model II; in both models, the hosts have to offer $P^*_{S_\alpha}$ to gain the best possible profit while paying the tax amount of $\delta = \Delta$. This means the regulator can reach the exact desired outcomes and goals derived in Model II by utilizing Model III as well. However, Model III is more advantageous since the government is also able to control the price and subsequently, control the demand for the short-term rentals by following the *effective tax with price change* policy; something that is not possible in Model II.

In conclusion, we can say utilizing Model III in this model (single host) is more beneficial since more goals can be achieved based on different values of parameters of δ and θ in this model.

As a summary of all the three models described in this chapter, we have illustrated the following flowchart that summarizes the optimal pricing strategy in each model, based on different values of μ, the service rate, θ, the utilization threshold, Δ, the tax amount in Model II, and δ, the tax amount in Model III.

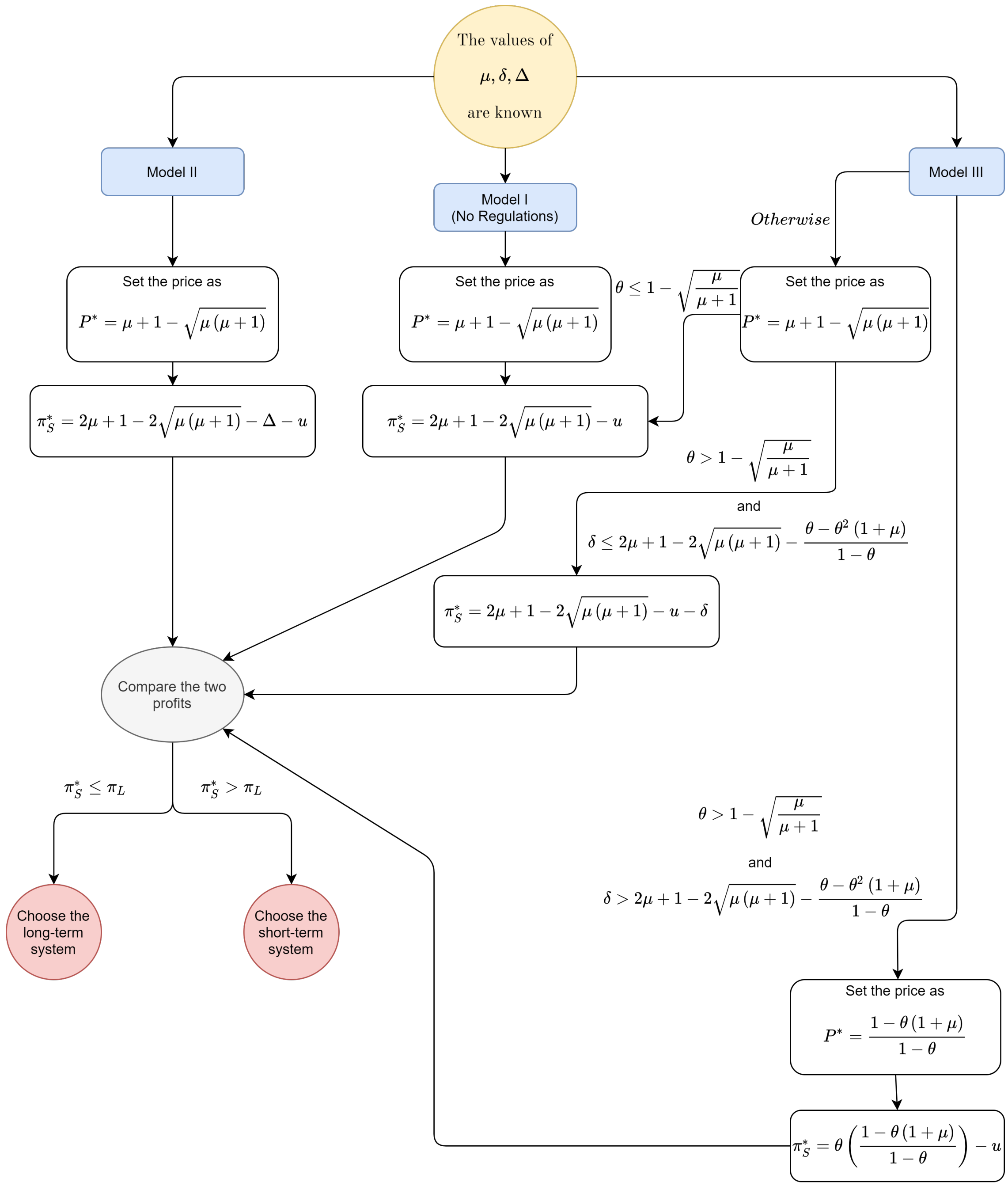


Figure 3.5: Summary of the optimal price strategies for each taxing regulation in an M/M/1/1 loss system

Chapter 4

A Modified Erlang Loss Model with Advance Reservation and Multinomial Logit Choice of Servers

In this chapter, we try to extend the model we discussed in the previous chapter to a more general model with N servers. We also add the two extensions mentioned previously to this system; advance reservation (for the arrival process) and the multinomial logit choice model (for the process of choosing among empty servers by customers). We develop a simulation in which we can find the utilization and, correspondingly, the profit for each server based on different queueing parameters. We will finally analyze the impacts of advance reservation on the productivity of the server.

4.1 M/M/N/N Erlang-Loss System

The M/M/N/N Erlang-Loss system generally works the same as M/M/1/1 systems stated previously. The only difference is that there are now N servers, and the system's maximum capacity is also N customers. The following figure illustrates this system:

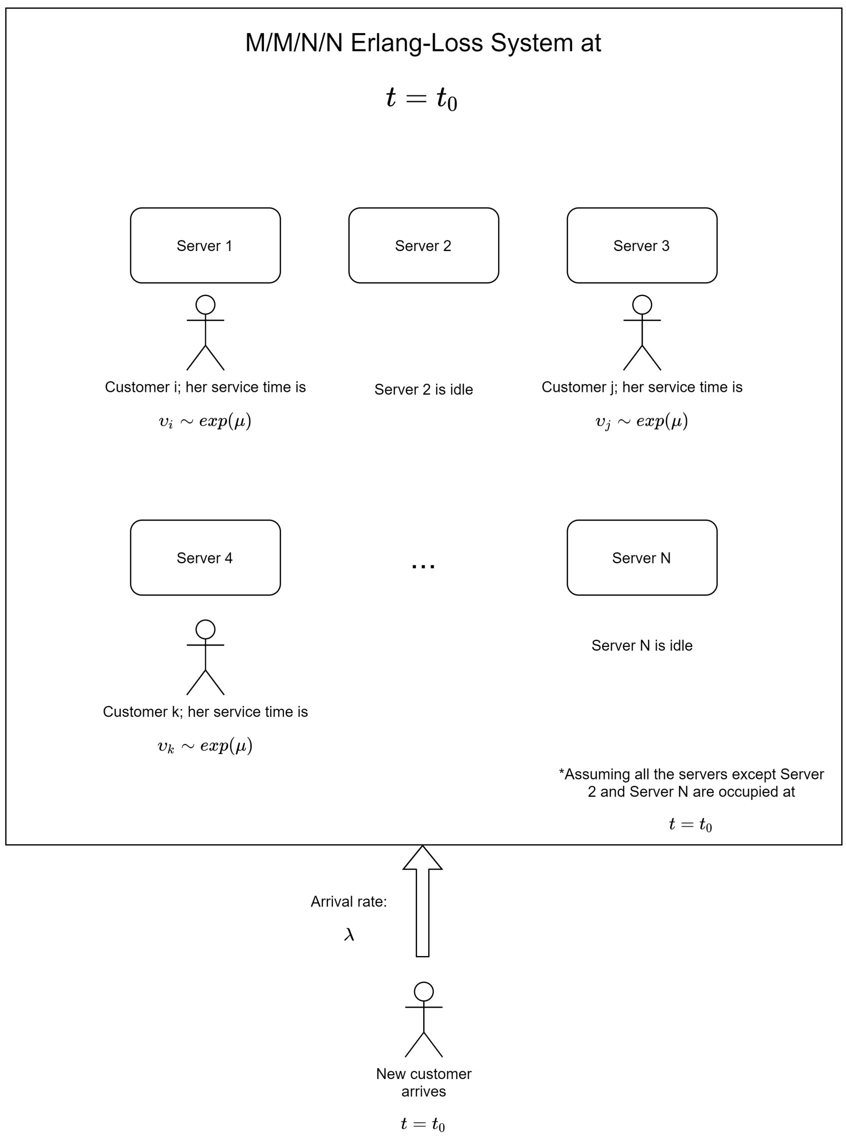


Figure 4.1: M/M/N/N Erlang-loss system illustration

Erlang's Loss Formula

One of the most important equations in the M/M/N/N loss system is Π_N: the steady state when there are N customers, independent of time, in the system. It will be derived as the following (see, e.g., equation 3.55 in Gross, 2008):

$$\Pi_N = \frac{\frac{\left(\frac{\lambda}{\mu}\right)^N}{N!}}{\sum_{k=0}^{N} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!}} \tag{4.1}$$

The above formula is equivalent to the customer loss probability in the system and is called *Erlang's Loss* formula, denoted by $B(N, \rho)$. Thus, if a customer arrives, there will be a Π_N chance that the system is fully occupied, so that customer cannot enter the system. That is why *effective* arrival rate and *effective* utilization are defined and equal to:

$$\rho_e = \frac{\lambda_e}{N\mu}, \quad \text{where } \lambda_e = (1 - \Pi_N)\,\lambda \tag{4.2}$$

Next, we will discuss the extensions to this current loss model.

4.2 Modified Erlang-Loss System

In this section, we consider two extensions to the classic Erlang-Loss system, and we will state our assumptions for each of them.

4.2.1 Advance Reservation

In the Erlang-Loss system, customers are limited to be served only upon their arrival. However, in real-world cases, there are many instances where people reserve in advance for their desired service. Almost any queueing system with real-world settings supports advance reservation; queueing systems of restaurants, short-term rentals, healthcare systems, and so forth. Hence, to include these types of customers with advance reservations, we need to categorize the arrivals into two types: 1) the ones who want to be served upon their arrival, 2) the ones who want to reserve for some time in the future. We assume that the customer of the second type arrives according to the Poisson process with the rate of $\lambda_{reserve}$. We then denote the arrival rate of the first type of customers by λ_{now} to be able to make a distinction between them. Moreover, for the former, we assume the number of days they reserve in advance follows an exponential distribution with the mean of τ.

Furthermore, since these two types of customers are independent, the overall arrival rate to the system will also be a Poisson process with the rate of $\lambda = \lambda_{now} + \lambda_{reserve}$.

4.2.2 Multinomial Logit Choice Model

In general, the servers in the Erlang-Loss system are identical, meaning that if a customer arrives, she will choose from the available servers uniformly at random. However, we want to consider a system in which each server is free to offer their desired price. In response to the prices, each upcoming customer needs to decide which available server will maximize her utility. To this end, we try to utilize the multinomial logit choice model (MNL) and define a utility function for each customer

i, and server j, inspired by Farhoodi (2019):

$$U_j^i = a_j + b_j P_j + \epsilon_j^i \tag{4.3}$$

Where P_j is the price of server j. Also, ϵ_j^i is customer i's individual specific utility of server j. In another point of view based on Train (2009), ϵ_j^i is a utility that is unknown to us but is observable by customer i and it follows the Gumbel distribution. Finally, the utility function of the outside option in customer i's point of view will be:

$$U_0^i = a_0 + b_0 P_0 + \epsilon_0^i \tag{4.4}$$

Based on different contexts for the queueing system, an outside option may refer to various alternatives. For example, in a short-term rentals system, an outside option can be hotel systems, whereas, in a food court queueing system, a restaurant outside that food court is considered the outside option.

With this set of assumptions, whenever a customer arrives at the system, she will calculate her utility derived from each available server while considering the outside option. She then chooses an available server that maximizes her utility.

The following figure illustrates the modified M/M/N/N Erlang-loss system, including advance reservation and MNL choice model:

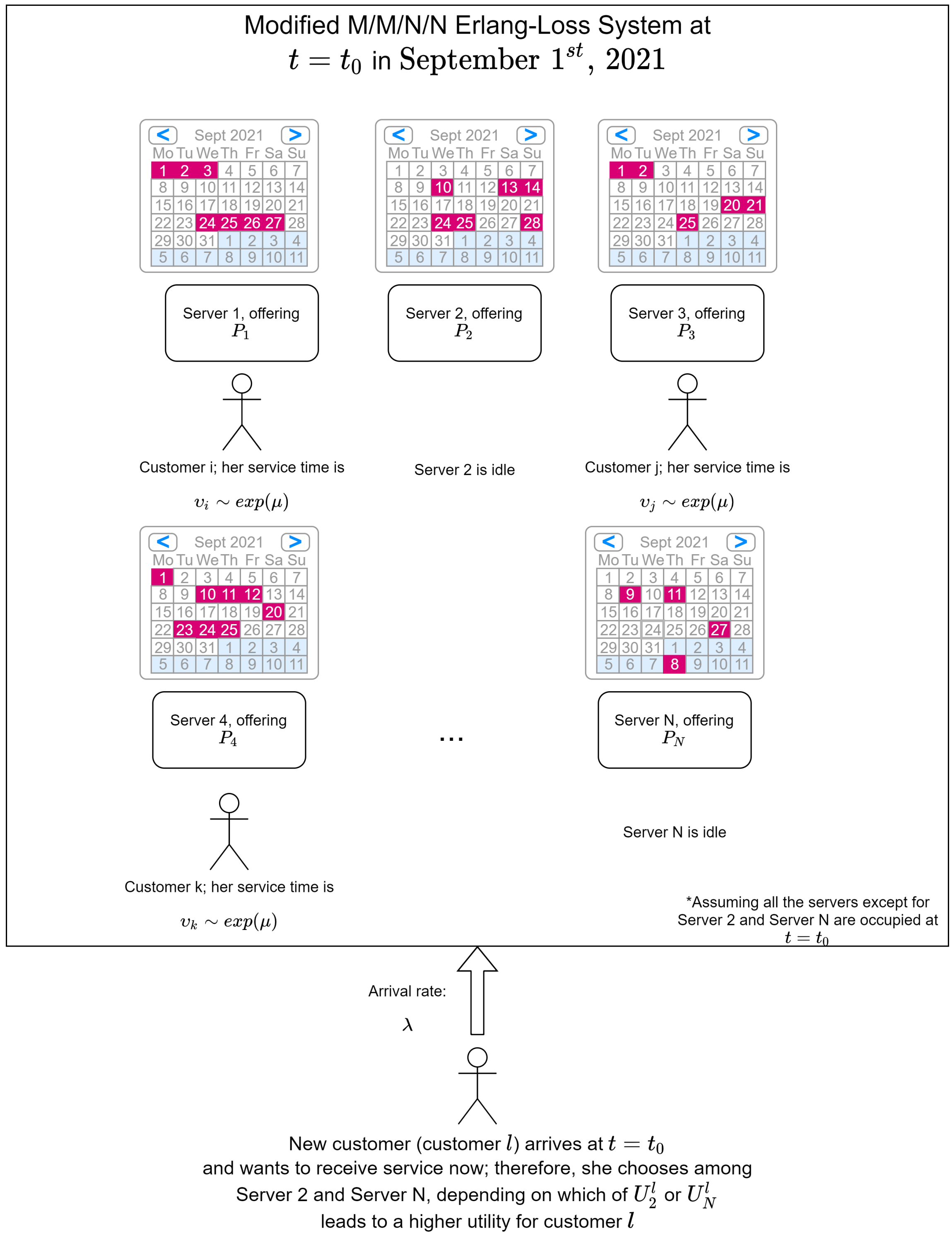


Figure 4.2: Modified M/M/N/N Erlang-loss system with advance reservation and MNL choice model

In Figure 4.2, the red spots show that those dates are already reserved, a feature that is not considered in the classic Erlang-loss system. Also, the upcoming customer chooses the unit which gives her the highest utility, which refers to the MNL choice model. Moreover, note that if **1)** all the prices are equal to P, **2)** for all values of i and j, $\epsilon^i_j = 0$ are i.i.d., **3)** for all values of j, a_j and b_j are equal, and **4)** $P_0 = P + \epsilon$, where ϵ is any small positive number, then the corresponding system will turn into the Erlang-Loss system since this leads to uniform choice over servers.

As one might notice, one of the most significant characteristics of queueing systems is finding the utilization and, subsequently, the profit gained by that system. Adding the two extensions of advance reservation and MNL choice model makes our queueing system intractable for some further analysis. Therefore, we decided to build a simulation model to find the utilization value of different servers based on various scenarios. In the next section, we will discuss more.

4.2.3 Applications in Reservation Systems

In some systems, such as Zipcar, the reservation is a core part of the system and cannot be removed; however, the question is how flexible the system should be regarding the maximum number of days to reserve in advance for their service. The same question can be asked when setting pricing strategies for short-term rental systems.

Moreover, policies could be slightly different for some other companies where the advance reservation is an add-on feature, such as ride-sharing systems (like Lyft or Uber), companies that recently have added the advance booking to their service. For example, as previously mentioned, Lyft let the customers reserve up to only seven days (Lyft Help, 2017) while its competitor, Uber, has the option to book a ride for one month ahead (Uber Help, 2016). Therefore, the first question in a price optimization problem for these kinds of systems is what percentage of the customers should be among the ones who want to reserve for some time in the future. The second will be how soon the customers are allowed to book a server (unit, ride, and a car in our case) to bring the maximum profit for the system. Therefore, we are interested in finding the answer to these questions, and we will try to explore more in the following sections.

4.3 Simulation of the Modified Erlang-Loss System

Due to the complexities that we mentioned earlier, we decided to simulate the modified Erlang-Loss system to find the utilization for each server; moreover, if we consider prices for each server, we will be able to find the profit for all of them.

For this purpose, we utilized SimPy (SimPy 4.0.1 Documentation, 2020) and NumPy 1.19.5 (Harris *et al.*, 2020) packages from Python on free version of Google Colaboratory notebook (Bisong, 2019). The CPU of Google Colaboratory platform that we have run the codes on is Intel(R) Xeon(R) CPU @ 2.30GHz with 13GB RAM . Our proposed simulation accepts the following as the inputs:

- λ: the overall arrival rate of the system
- λ_{now}: the arrival rate of the first type of customers to the loss system (e.g., the ones who want to be served upon their arrival)
- $\lambda_{reserve}$: the arrival rate of the second type of customers to the loss system (e.g.,

the ones who want to reserve a server for some time in the future)

- κ: arrival split rate - as we previously mentioned, $\lambda = \lambda_{reserve} + \lambda_{now}$; therefore, we can simply assume that κ percent of the overall arrivals are of type 1, e.g., $\lambda_{now} = \kappa\lambda$ and the rest are of type 2, e.g., $\lambda_{reserve} = (1 - \kappa)\lambda$
- μ: the service rate of the system
- N: the number of servers
- $P_i, \quad \forall i \in \{1, 2, ..., N\}$: the price of each server (if the system is not price-based, the user can input zero for each of them)
- P_0: the price of the outside option (if the system is not price-based, the user can input zero for this)
- T: the duration of the simulation in days
- t: the duration of the warm-up session in days
- τ: the average number of days that the customers book in advance for the reservation
- ε: the error threshold for terminating the simulation
- η: the maximum number of iterations if not reached the threshold

When all the inputs are given, simulation starts, and customers arrive based on the given arrival rates; however, the busy time of servers will not be collected during the warm-up period to let the system be adapted to the normal conditions based on the given data. After the warm-up period, the simulation will run for T more

Algorithm 1 Simulation of the Modified Erlang-Loss System

input: $(\lambda, \kappa, \tau, \mu, N, P_i, T, t, \varepsilon, \eta)$
output: $(u_1, u_2, ..., u_N)$
$itr \leftarrow 1$
while $itr < \eta$ **do**
  $t_0, \nu_j \leftarrow 0$
  while $t < t_0 < T + t$ **do** ▷ during warm-up, results are not collected
    for each incoming customer i, let them choose among the servers available for the requested time interval **do**
      if customer i chooses server j **then**
        $\nu_j \leftarrow \nu_j + \upsilon_i$
      end if
    end for
  end while
  for each server j **do**
    $u_j^{[itr]} \leftarrow \frac{\nu_j}{T}$
    $\bar{u}_j^{[itr]} \leftarrow \frac{\sum_k^{itr} u_j^{[k]}}{itr}$
    if for any server j, $|\bar{u}_j^{[itr]} - \bar{u}_j^{[itr-1]}| > \varepsilon$ **then**
      $itr \leftarrow itr + 1$
      break
    else
      for each server j **do**
        $\hat{u}_j \leftarrow \bar{u}_j^{itr}$
      end for
      return $(\hat{u}_1, \hat{u}_2, ..., \hat{u}_N)$ ▷ desired accuracy reached
      terminates
    end if
  end for
end while
return $(\bar{u}_1^{[\eta]}, \bar{u}_2^{[\eta]}, ..., \bar{u}_N^{[\eta]})$ ▷ desired accuracy not reached

units of time. Each upcoming customer chooses among available servers based on the MNL choice model, and their service time, υ_i, will be assigned to their corresponding server. Moreover, if we assume that the upcoming customer i chooses server j, υ_i will be added to ν_j, the time that server j has been busy up to now during the simulation. When t_0, the system's current time, reaches T, the simulation will stop, and the utilization of each server will be calculated. The whole cycle will be repeated for few more iterations. At the end of each iteration, the average utilization for each server (from the beginning of the simulation up to the last iteration) will be calculated. If the difference between two successive average utilization is smaller than the error threshold, ε, for all the servers, the simulation will terminate. Otherwise, it will continue until the number of iterations equals η, and then the final average for each server will be returned.

4.4 Simulation Experiments

Now, we try to solve a couple of problems to evaluate our proposed simulation and do some experiments to analyze the impacts of the different parameters on the utilization of the servers. First, we try to run an Erlang-Loss system in our simulation to evaluate the accuracy of the results. We can do so with the help of Erlang-Loss queueing systems formulas.

4.4.1 Problem 1: Evaluation of the Simulation with Erlang-Loss System

Consider the following input:

$\lambda = 2, \kappa = 1, \tau = 0, \mu = 0.1, N = 15, P_0 = 0.01, P_i = 0, T = 365, t = 30, \varepsilon = 10^{-4}, \eta = 500$. Since we wanted to compare our simulation with the Erlang-Loss system, we removed the MNL choice model by assuming $\epsilon_j^i = 0$ for all sets of i and j. With this setting, any nonzero price for the outside option will always lose. Furthermore, since we have assumed $\lambda_{reserve} = 0$, we will have $\lambda = \lambda_{now} + \lambda_{reserve} = \lambda_{now}$ which means that we are dealing with an Erlang-Loss system.

To be able to compare the results of the simulation, we used the iterative Erlang-Loss formula to calculate the utilization in the aforementioned system (see, e.g., equation 3.56 in Gross, 2008):

$$\Pi_N = B\left(N, \rho\right) = \frac{\rho B\left(N-1, \rho\right)}{N + \rho B\left(N-1, \rho\right)}, \quad N \geq 1, \quad \text{where } \rho = \frac{\lambda}{\mu} \tag{4.5}$$

The results of the comparison are brought in Table 4.1.

Utilization	Simulation	Erlang-Loss Formula	Absolute Error
Server 1	0.8922	0.8933	0.0012
Server 2	0.8930	0.8933	0.0004
Server 3	0.8913	0.8933	0.0020
Server 4	0.8922	0.8933	0.0011
Server 5	0.8914	0.8933	0.0020
Server 6	0.8944	0.8933	0.0011
Server 7	0.8928	0.8933	0.0005
Server 8	0.8962	0.8933	0.0028
Server 9	0.8953	0.8933	0.0020
Server 10	0.8944	0.8933	0.0011
Server 11	0.8916	0.8933	0.0018
Server 12	0.8916	0.8933	0.0017
Server 13	0.8921	0.8933	0.0012
Server 14	0.8934	0.8933	0.0000
Server 15	0.8955	0.8933	0.0022
Average	0.8932	0.8933	0.0002

Table 4.1: Evaluating our proposed simulation by comparing the utilization results with the Erlang-Loss formula

As can be seen in Table 4.1, the absolute error is close to zero (considering the error threshold is 10^{-4}), which validates the accuracy of our simulation. As a reminder, we have purposefully modified the inputs to the simulation to have an Erlang-Loss system; thus, we expected to see results similar to those from Erlang-Loss formulas.

The simulation terminated after 275 iterations (13.51 seconds). There is some other helpful information that can be derived from the simulation. For example, we can count the number of blocked customers upon their arrival, e.g., they arrived at the system while all the servers were busy; hence, they had to leave the system. The average number of customers who reneged is 241 in the duration of 365 days, while the number of total arrivals is 490. We have illustrated the customers who were blocked in each day in the following figure:

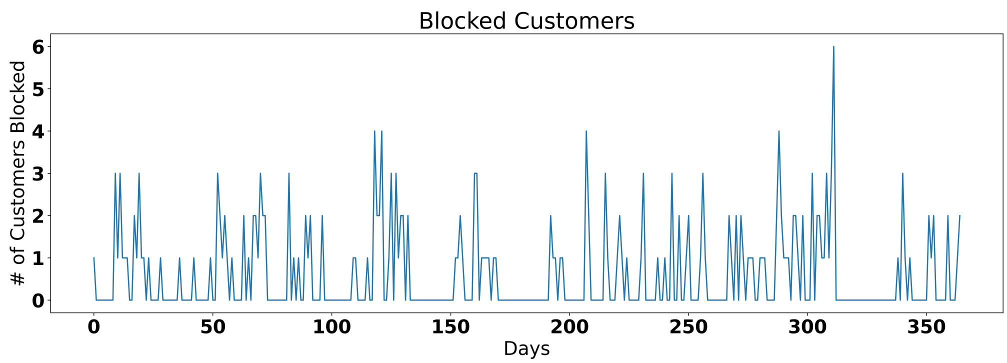
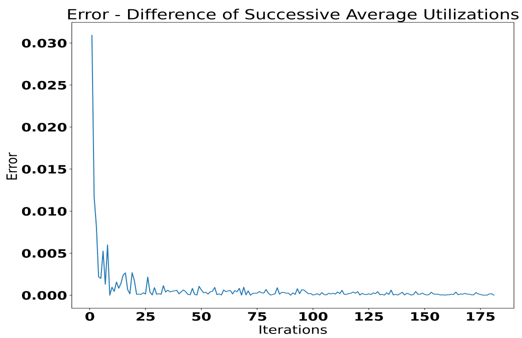
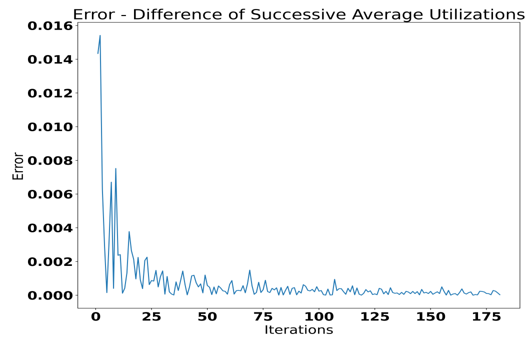


Figure 4.3: Number of customers who are blocked in each day

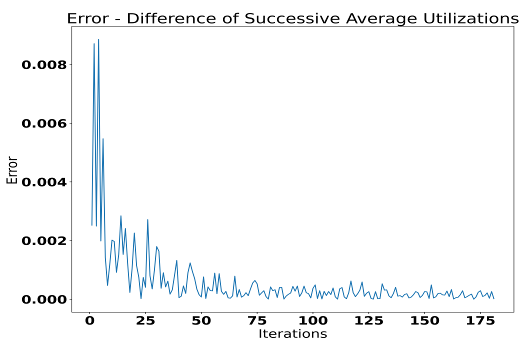
Furthermore, the error convergence for the first four server has been depicted in Figure 4.4; the rest of the servers follow the same pattern. In each graph, as one can see, the difference between two successive average utilization is comparably high and fluctuates often; however, it will be more stable after few iterations until the point where the difference between two successive average utilization for each server is less than ε.



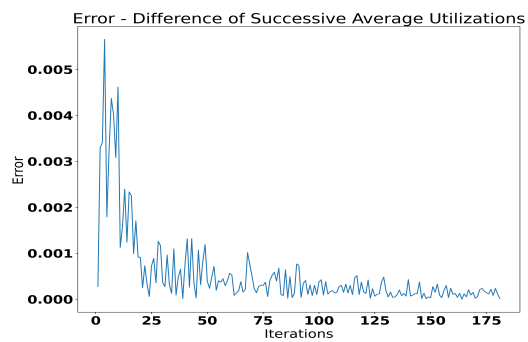
(a) Server 1



(b) Server 2



(c) Server 3



(d) Server 4

Figure 4.4: Error convergence for the first four servers in the simulation - Problem 1

Now that we have made sure our proposed simulation works with comparably high accuracy, as the next problem, we try to utilize the advantages of this simulation and compare the effects of advance reservation on servers' utilization.

4.4.2 Problem 2: The Impacts of $\lambda_{reserve}$ on Utilization

In this problem, we assume the overall arrival rate to the system is fixed and is equal to 10, e.g., $\lambda = \lambda_{now} + \lambda_{reserve} = 10$. We then run different simulations with various values of λ_{now} and $\lambda_{reserve}$. Thus, we can consider the input to be as the following:

$\lambda = 10, \kappa = k, \tau = 7, \mu = 1, N = 15, P_0 = 0.01, P_i = 0, T = 365, t = 30, \varepsilon =$

$10^{-4}, \eta = 500$ for $k = \{0, 0.1, 0.2, ..., 1\}$. As can be seen, we have once again assumed that all the prices are equal (i.e., zero) and $\tau = 7$, e.g., on average, the second type of customers reserve for one week ahead. We then run the simulation for 11 times and the following table is a summary of the results:

λ_{now}	$\lambda_{reserve}$	**# of Served Customers**	**# of Blocked Customers**	**# of Arrivals**	**Average System Utilization**	**Equivalent Erlang-Loss System**
10	0	3513	132	3645	0.6385	0.6423
9	1	3507	141	3648	0.6353	0.6423
8	2	3500	150	3650	0.6262	0.6423
7	3	3477	162	3639	0.6142	0.6423
6	4	3450	177	3627	0.6006	0.6423
5	5	3429	202	3631	0.5898	0.6423
4	6	3404	220	3624	0.5774	0.6423
3	7	3384	241	3625	0.5661	0.6423
2	8	3376	252	3628	0.5607	0.6423
1	9	3380	269	3649	0.5599	0.6423
0	10	3380	270	3650	0.5574	0.6423

Table 4.2: Comparison of λ_{now} and $\lambda_{reserve}$ effects on system utilization

The last column, equivalent Erlang-loss utilization, is the system in which $\lambda = 10$, and it does not support advance reservation. Moreover, as can be seen, although the overall arrival rate to the system is fixed, when the advance reservation increases, the utilization keeps decreasing. One can interpret it in this way: the more uncertainty we have in the loss system, the more loss we will have in the utilization of the system. The reason can be mainly due to the increase in the number of blocked customers, e.g., the ones who enter the system but all the servers are either already occupied or already reserved for that time. As an instance, consider a case in which one customer reserves for server j from day 10 to day 12. On the other hand, assume a customer arrives at the system at day 3 and wants to stay there until day 11. In this case, server j will not be available for this customer as it is already reserved for day 10 to day 12. That is one of the scenarios why the number of blocked customers will increase when the advance reservation increases. Based on different values of λ_{now} and $\lambda_{reserve}$, we depicted the decrease in the utilization and comparison of the served and blocked customers in Figure 4.5, and Figure 4.6, respectively:

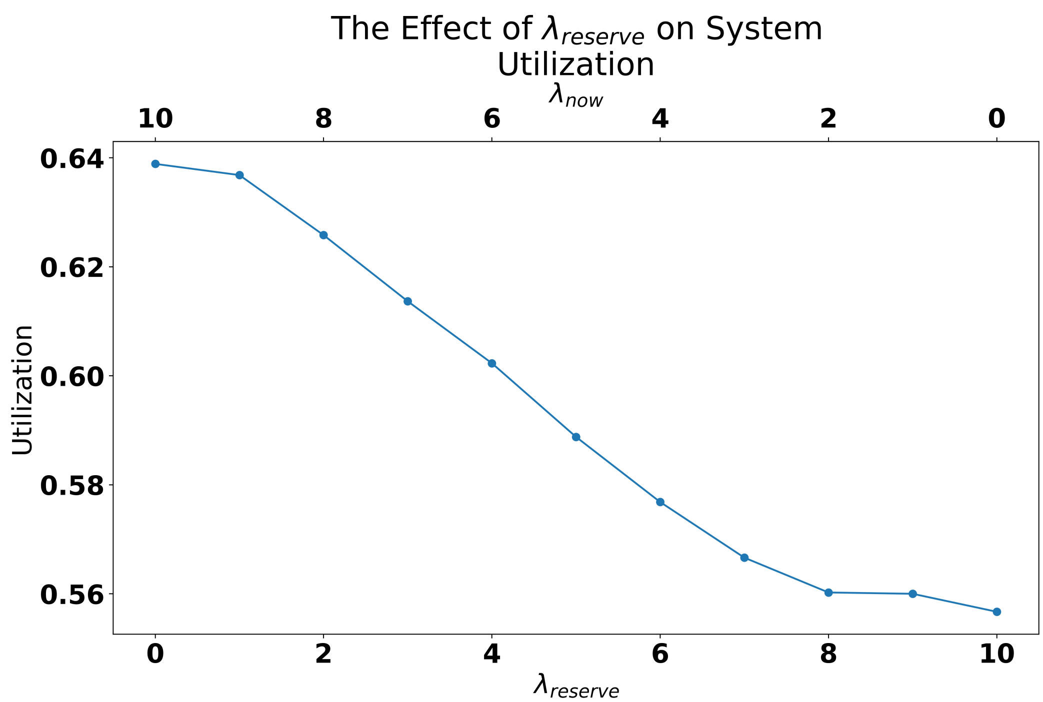


Figure 4.5: The decrease in utilization by increasing the value of $\lambda_{reserve}$ while keeping λ fixed (decreasing κ)

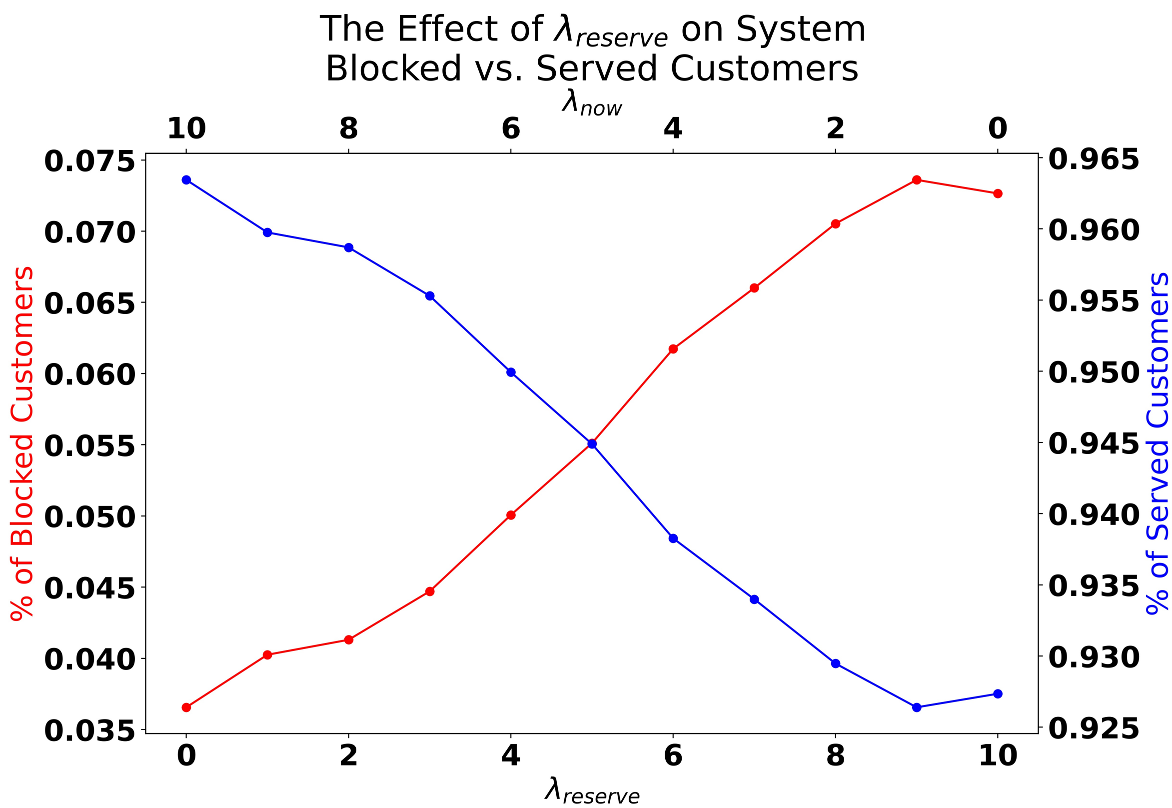


Figure 4.6: Comparison of percentage of the served customers with the percentage of the blocked customers based on different values of $\lambda_{reserve}$ and λ_{now} while λ is fixed

In the next problem, we try to analyze the impacts of τ in advance reservation Erlang-Loss systems.

4.4.3 Problem 3: Impacts of τ on Utilization

This time, we aim to see if there is a relation between the value of τ, e.g., the average number of days that the second type of customers tend to reserve in advance, and the utilization of the system. For this purpose, we modify the input to the simulation as below:

$\lambda = 10, \kappa = 0.5, \tau = k, \mu = 1, N = 15, P_0 = 0.01, P_i = 0, T = 365, t = 30, \varepsilon = 10^{-4}, \eta = 500$ for $k = \{1, 2, ..., 14\}$. Thus, we are going to run the simulation for 14 times, each time with a different value of τ. The results are brought in the following table:

τ	**# of Served Customers**	**# of Blocked Customers**	**# of Arrivals**	**Average System Utilization**	**Equivalent Erlang-Loss Utilization**
1	3473	170	3643	0.6236	0.6423
2	3462	182	3644	0.6112	0.6423
3	3455	193	3648	0.6047	0.6423
4	3446	198	3644	0.5992	0.6423
5	3435	199	3634	0.5941	0.6423
6	3430	202	3632	0.591	0.6423
7	3424	204	3628	0.5879	0.6423
8	3424	203	3627	0.5872	0.6423
9	3422	203	3625	0.5845	0.6423
10	3431	207	3638	0.5839	0.6423
11	3437	207	3644	0.5814	0.6423
12	3432	210	3642	0.5827	0.6423
13	3435	214	3649	0.5836	0.6423
14	3429	213	3642	0.5801	0.6423

Table 4.3: Comparison of τ effects on system utilization

We have also brought the graph of the utilization based on different values of in

Figure 4.7:

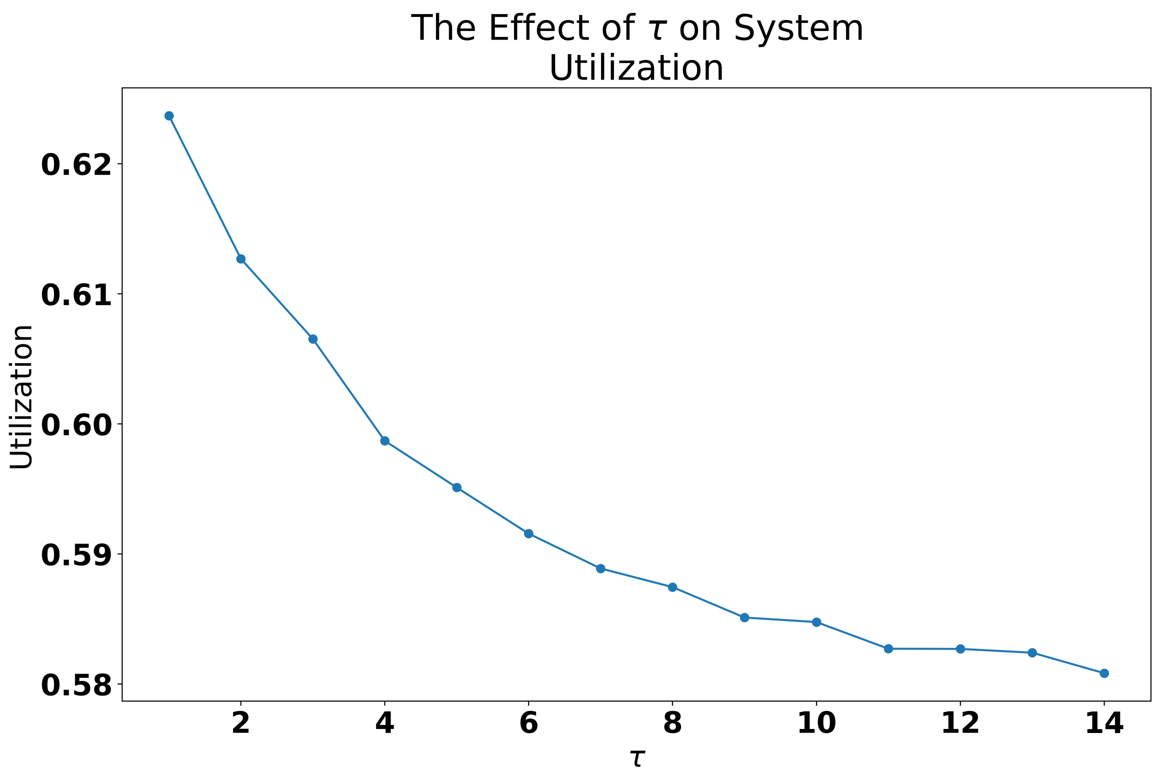


Figure 4.7: The decrease of the utilization by increasing the value of τ while all other parameters are fixed

As we can see, with the increase in the value of τ, the utilization tends to decrease for a fixed value of the arrival rate. This validates our previous observation regarding the impact of the reservation arrival rate; now, when the value of τ is higher, the advance reservation impact will be even bolder. In other words, when the value of τ is comparably high, for example, when $\tau = 12$ compared to when $\tau = 2$, the chances of losing customers are higher since in the former, there are more days from the date a customer reserves a unit in advance until the date she enters the system. Thus, more customers will arrive in this period and more losses we will probably have. This increase in the number of customer loss can be noticed in the following figure:

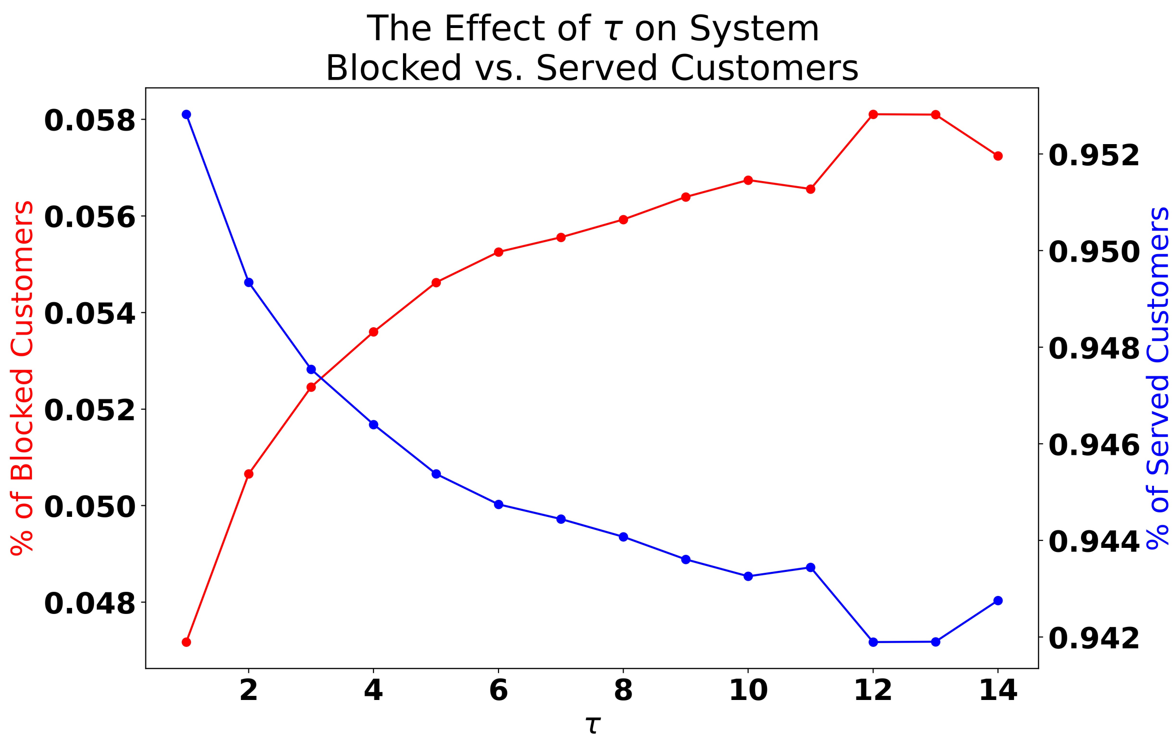


Figure 4.8: Comparison of number of served customers with the number of blocked customers based on different values of τ

Comparing Figure 4.5 with Figure 4.7, and also, Figure 4.6 with Figure 4.8, one can see that both parameters generally have a same effect on the utilization. However, there is a slight difference and that is τ's effect is less extreme than $\lambda_{reserve}$ on the amount of utilization. It can also be validated through Figure 4.7 that the decrease in the utilization is smoother than Figure 4.5. Nonetheless, for small values of τ, the utilization loss will be negligible. Therefore, a good policy could be letting a queueing system support advance reservation but limiting it for small periods of reservation, e.g., the time from the date they reserve a server until the date they enter it.

Now that we have analyzed the impacts of different parameters of advance reservation on the utilization, we will simulate another loss system in which both advance reservation and MNL choice model is included. Also, we consider a state in which prices are not exactly equal.

4.4.4 Problem 4: Comparison of the Erlang-Loss Model with our Proposed Modified Erlang-Loss Model

As the last solved problem with simulation, we consider a case where both advance reservation and MNL choice model are included. Therefore, we modify some parameters in Problem 1 and the input to the system will be as follows:

$\lambda = 2, \kappa = 0.5, \tau = 7, \mu = 0.1, N = 15, P_0 = 100, P_1 = 95, P_2 = 95, P_3 = 97, P_4 = 97, P_5 = 99, P_6 = 99, P_7 = 101, P_8 = 101, P_9 = 103, P_{10} = 103, P_{11} = 105, P_{12} = 105, P_{13} = 107, P_{14} = 107, P_{15} = 109, T = 365, t = 30, \varepsilon = 10^{-4}, \eta = 500$. Comparing the inputs of this current model with the inputs of Problem 1, we can see that we split the overall arrival rate equally between the two types of customers (now and reserve). Also, we assumed that there are some differences between prices of the servers as can be seen. Moreover, to include the MNL choice model, we considered the utility functions for each customer i and server j be according to equation (4.3) while assuming $a_j = 0$ and $b_j = -1$ for simplicity. Finally, for ϵ_j^i, we assumed based on Train (2009) that $\beta = 0.5772$ as the location parameter, and $\mu = 1$, as the scale parameter.

In the following table, we brought the summary of the results:

Server	**Price**	**Utilization**	**Profit**
Server 1	95	0.8023	27819.2376
Server 2	95	0.8042	27884.195
Server 3	97	0.8243	29183.0278
Server 4	97	0.8248	29201.5381
Server 5	99	0.8303	30002.3038
Server 6	99	0.8307	30019.0855
Server 7	101	0.7477	27564.7728
Server 8	101	0.7467	27527.2045
Server 9	103	0.3764	14150.9058
Server 10	103	0.3861	14516.97
Server 11	105	0.0832	3188.1444
Server 12	105	0.0775	2970.7551
Server 13	107	0.0126	492.9283
Server 14	107	0.013	505.9491
Server 15	109	0.0017	67.0675

Table 4.4: The summary of the utilization and profit for each server

In the above table, we also showed the profit for each server (defined as $\pi_j = P_j \times u_j \times 365$). As can be seen, since we have included the MNL choice model and thus, due to the existence of ϵ_j^i, which creates some randomness for the utility functions of each customer, the cheapest servers do not necessarily have the highest utilization. Moreover, if we calculate the average of the utilization values, we will have $\bar{u} = 0.4908$. Comparing with the results of Problem 1 (Table 4.1), we notice that the average utilization has decreased by around 0.4, which is a significant drop. One of the reasons may be the outside price; $P_0 = 100$ is cheaper than half of the servers; therefore, it will be preferable to those servers most of the time.

Nonetheless, because of the MNL choice model, servers 7 to 15 will not have a utilization precisely equal to zero; for some customers, their corresponding utility function may be even higher than the outside option. The other reason is that in this problem, we have assumed half of the arrivals are the ones who reserve for some time in future (instead of all the arrivals seeking service upon their arrival, Problem 1). Based on our previous discussion, the advance reservation will reduce the utilization of the servers, which is noticeable here.

Now that we have solved several cases and have some insights into the simulation, we will list its strengths and weaknesses in the following section.

4.5 Advantages and Disadvantages of the Simulation

Although the simulation works fine for small and simple instances (i.e., 13.51 seconds for Problem 1), when the scale of the problem gets larger, especially for large values of arrival rate, the computation will slow down. In Table 4.5, we show some various scenarios (inputs) and the corresponding time to run each of them in order to see how the number of servers affect the computation speed.

# of Servers	**Arrival Rate**	**Service Rate**	**Itr**	**Average Utilization**	**Total Time**
5	10	1	41	0.8721	9.00
5	100	2	13	0.9788	25.28
10	20	1	87	0.9238	38.88
10	100	10	47	0.7857	116.25
25	100	1	20	0.9870	42.29
50	100	2	110	0.8951	319.02

Table 4.5: Computation comparison with different inputs

We need to point out that the other parameters that have not been mentioned in the above table are assumed to be the same as the ones in Problem 1. As can be seen, when the arrival rate increases, the elapsed time will increase especially if it is the case where the utilization is not close to 1. On the other hand, when the utilization is so close to 1, it means that most of the arrivals are being blocked which is not a time-consuming process in the simulation; however, if the utilization is not close to 1, it means that most of the arrivals are being processed by the simulation (i.e., which available units each arrival has to choose, how long they are going to stay in those servers, and so forth) that takes some time. Moreover, when we add advance reservation to it, e.g., $\lambda_{reserve} \neq 0$, and $\lambda_{reserve}$ and τ are not too small, the computation will slow down more than before and it will take even more time to compute the results.

On the other hand, as mentioned before, one of the most significant parameters in a loss system is the utilization of each server. By knowing the utilization for each server, different strategies can be used based on various contexts in which the system is defined. That is why we have decided to train a machine learning (ML) model to predict the utilization in the modified Erlang-Loss system to overcome the slow computation barrier. In the next chapter, we will discuss more.

Chapter 5

Predicting the Utilization in Modified Erlang-Loss Systems with Machine Learning

To build a machine learning model for predicting the utilization of each server, we first need to create a dataset. We will then try different algorithms to train the ML model and choose the best among them in terms of accuracy.

5.1 Creating the Dataset

We created 13000 random input tuples, ran the simulation for each, and kept the outputs as the label for each input. Also, we specified an interval for all the parameters and picked the random values from those intervals. Table 5.6 gives a brief description of the inputs:

	λ	κ	τ	μ	P_0	P_1	P_2	P_3	P_4	P_5
Mean	5.01	0.5	10.07	5.01	272.95	273.24	274.3	272.77	274.88	275.66
STD	2.9	0.29	5.77	2.88	128.53	129.68	129.12	129.79	129.76	129.67
Min	0	0	0	0	46.71	45.56	47.39	49.41	46.96	46.93
25%	2.52	0.25	5.06	2.53	161.46	160.99	163.17	159.5	162.15	163.16
50%	4.98	0.5	10.07	5.01	272.63	272.94	273.18	272.4	275.87	275.59
75%	7.53	0.75	15.1	7.49	382.77	385.37	385.66	385.76	387.85	388.65
Max	10	1	20	10	504.39	503.4	503.66	503.08	503.47	504.68

Table 5.6: Description of the dataset created for building ML models to predict the utilization for each server in an M/M/5/5 loss system

As it can be seen, we have fixed the number of servers, $N = 5$, to make the ML modeling simpler. Among these 13000 samples, we took 3000 of them with closer offered prices; in other words, in those 3000 samples, a randomly chosen server, say P_i was selected randomly from 50 to 500; however, the other prices were chosen randomly from the interval $(P_i - 5, P_i + 5)$. The reason behind that was to train the model to be familiar with the cases that the prices are somehow closer to each other as well; the case that can be more realistic than the time when one server offers $P_i = 50$ and the other offers $P_j = 500$. For the other 10000 samples, each price was chosen uniformly and separately at random from the range of 50 to 500.

To illustrate the correlation between different parameters in this dataset, we used the *pairplot* feature in *Seaborn* package (Waskom, 2021) with the help of *matplotlib* (Hunter, 2007). We brought the pairwise plots of λ, κ, τ, μ and P_1 in Figure 5.9 (the rest of the prices will follow the same pattern as P_1).

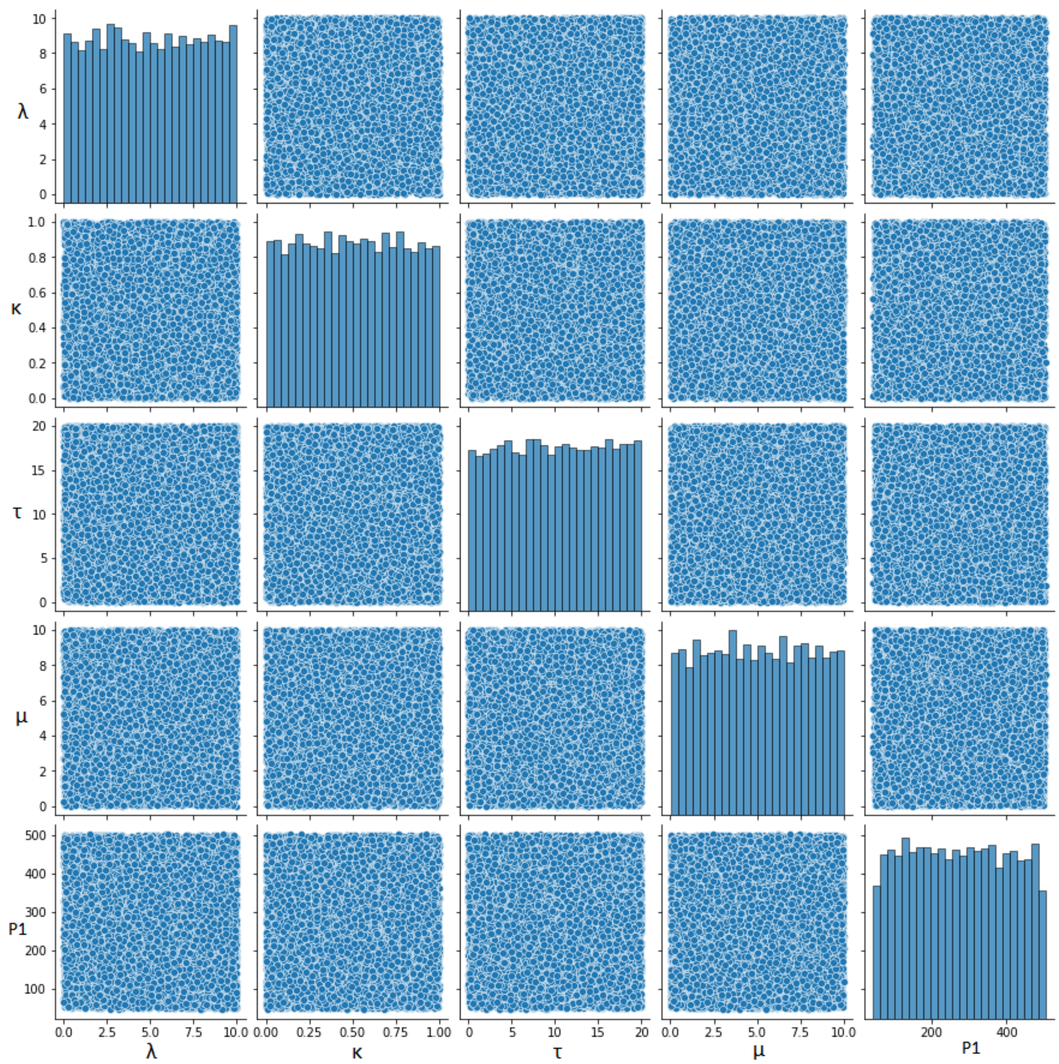


Figure 5.9: Pair plots of the features of the dataset including the first server's price

As can be seen, for each pair of parameters, there is full coverage for both x and y axes. This shows that the input data has been chosen uniformly at random and they are independent. On the diagonal, on the other hand, we can see the distribution plots of each of these parameters which have almost been distributed uniformly over the specified range.

Moreover, for some further analysis on the dataset, we brought Figure 5.10, the pair plots related to P_1 and service rate. The P_1-related plots show that although almost every price can reach high utilization in the system, the dots are denser for smaller prices. It makes sense for when a server is offering a smaller price in a specified range (in our case, 50 to 500), chances will be higher to be the winner in the price competition among all the servers. For the relation between service rate and utilization, we can see that when the service rate is close to zero, the utilization is almost equal to 1. Moreover, the corresponding profit will also be at its highest. It makes sense since, for example, when $\mu = 0.001$, an upcoming customer may stay in the system for 1000 days on average, which is not realistic but happens due to randomness in the data collection.

Lastly, for the relation between price and the corresponding profit, when price increases, the profit generally increases; however, the occurrences drop drastically. It means that setting the highest price does not necessarily lead to the highest profit.

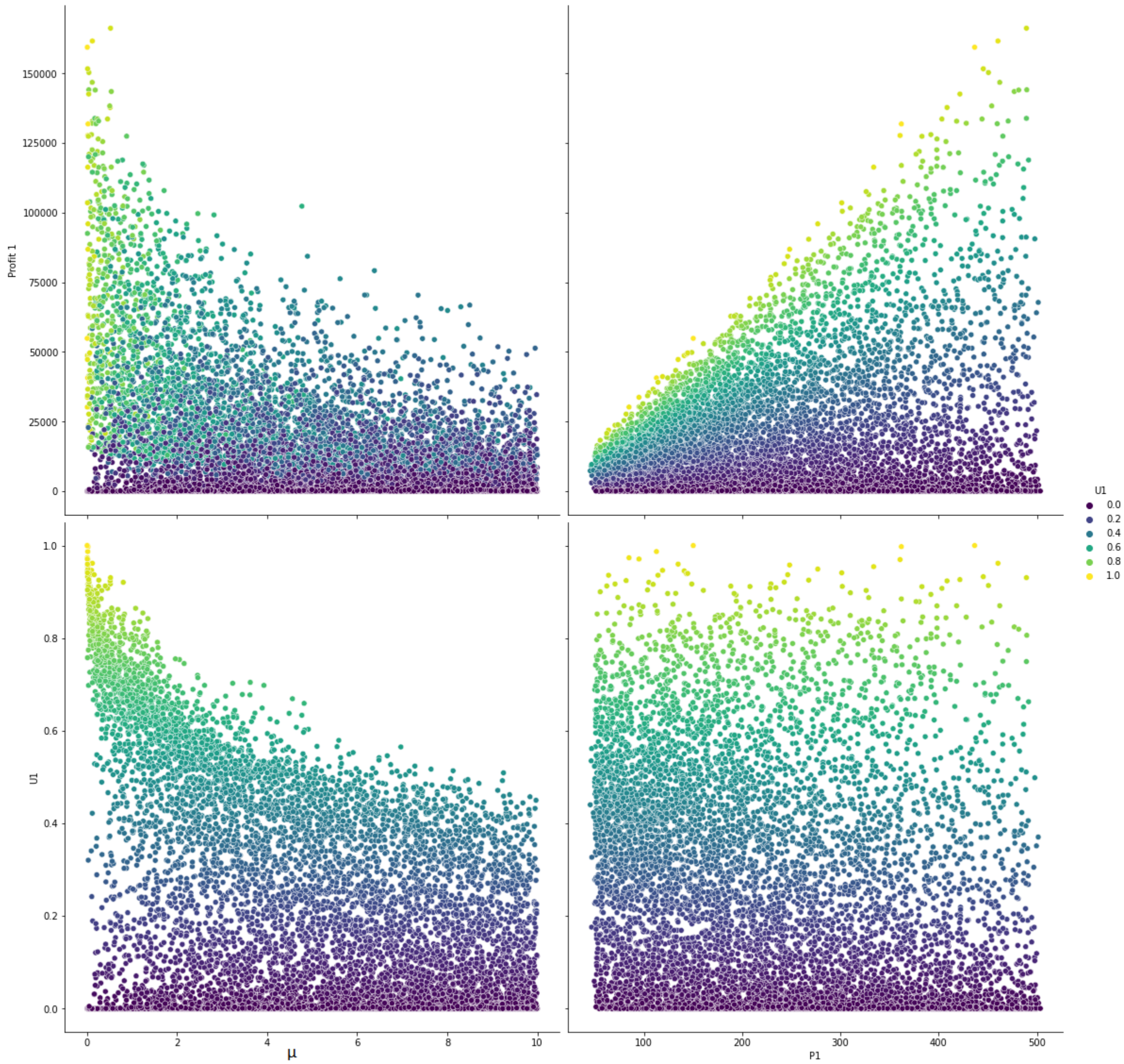


Figure 5.10: Pair plots of the service rate, first server's price and profit including its utilization

Now that we have some insights regarding our dataset, we can start working on some ML models in the following sections.

5.2 Predicting the Utilization: Neural Network

In this section, we will start training our model with a neural network algorithm using *TensorFlow 2.5* library (Abadi *et al.*, 2015). We start with assigning 80% of the data as the training set and the rest as the test set. The feature set has 10 columns, and the label set has 5 columns: utilization of each server. Since the input data have different scales (rates, days, prices), we decided to normalize the data by subtracting the mean values and dividing the result by the standard deviation.

Next, we defined two hidden layers of 64 neurons with the sigmoid activation function and RMSprop optimizer (learning rate = 0.001). The following figure shows a straightforward illustration of the model:

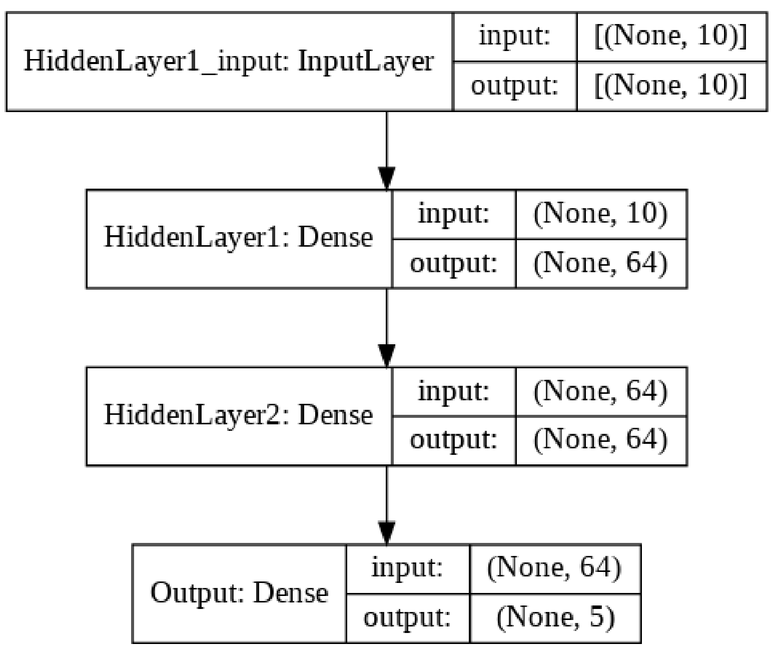


Figure 5.11: The setting of the layers in the proposed neural network model for predicting the utilization of each server

According to Figure 5.11, the input layer has the size of 10 for both its input and output since we have 10 features. Then, for the first hidden layer, the input has the same shape as the previous layer's output (input layer), and this pattern continues until the last layer where the output is 5, e.g., the number of labels to be predicted. Moreover, the *None* in each layer is the batch size which can vary; that is why it is indicated as *None*.

After designing the model, we trained it with 4000 epochs (each sample from the training set went through the whole network for 4000 iterations) and 20% of the validation split. The following figure shows the mean absolute error at each epoch for both the training set and the validation set:

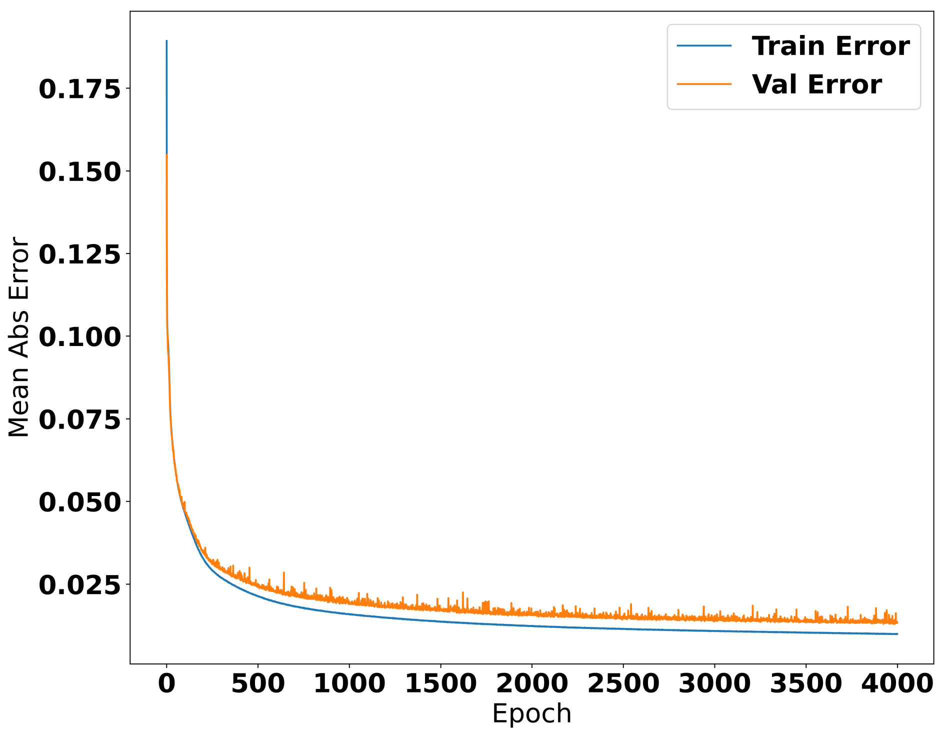


Figure 5.12: MAE in training and validation set

As can be seen, the MAE decreases when the number of epochs increase. There is still an improvement for the validation set; however, it is insignificant to be considered. On the other hand, since the validation error does not increase, we do not have overfitting for this model.

Moreover, we have also brought the errors at the last five epochs of training in the following table:

	Loss	**MAE**	**MSE**	**Validation Loss**	**Validation MAE**	**Validation MSE**	**Epoch**
3995	0.0004	0.0097	0.0004	0.001	0.0141	0.001	3995
3996	0.0004	0.0098	0.0004	0.001	0.0136	0.001	3996
3997	0.0004	0.0098	0.0004	0.001	0.0135	0.001	3997
3998	0.0004	0.0098	0.0004	0.001	0.0132	0.001	3998
3999	0.0004	0.0097	0.0004	0.001	0.0135	0.001	3999

Table 5.7: Loss, MAE, and MSE in training and validation sets for the last five epochs

We then tried our model to predict the test data. For this purpose, we illustrated the graph of model predictions compared with true values for each instance in test set:

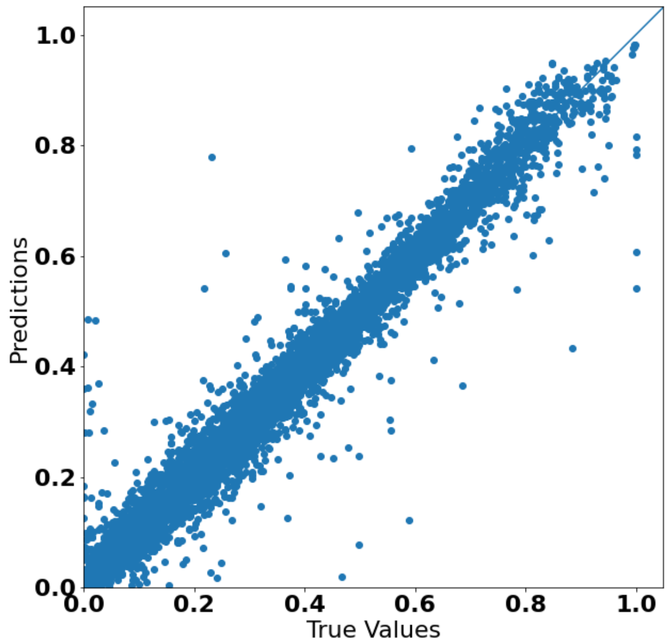


Figure 5.13: Comparison of the model prediction and the true value of the test set

Based on Figure 5.13, the closer the dots are to the $y = x$ line, the more accurate the model will be. As can be seen, there are a few outilers which have slightly more error than the ones around the line; nevertheless, they happen rarely.

Finally, in the following table, we brought the errors of predicting the test set:

Loss	**MAE**	**MSE**
0.0009	0.0138	0.0009

Table 5.8: The errors for the test set

In the next section, we will try to train few more algorithms and compare the results.

5.3 Predicting the Utilization: Random Forest, Extra Trees and Gradient Boosting

In this section, we will train a model with three multi-output regression algorithms: random forest, extra trees, and gradient boosting. Since all these three methods are based on decision trees, we decided to work on them together. Moreover, to build and compare our proposed models in this section, we use *Scikit-learn* library (Pedregosa *et al.*, 2011). For this purpose, same as before, we first split the data into training and test set with the split of %80. On the contrary, we did not normalize the data as in decision trees; the normalization does not impact the functionality of the algorithm.

Now, we can train the three above mentioned algorithms and choose the better values for hyperparameters by tuning. For this purpose, we set a list for some of the more essential hyperparameters and brought them in the following table:

Hyperparameter/Algorithm	**Multi-Output Random Forest**	**Multi-Output Extra Trees**	**Multi-Output Gradient Boosting**
n_estimators	[200, 400, ..., 2000]	[200, 400, ..., 2000]	[200, 400, ..., 2000]
max_features	[auto, sqrt]	[auto, sqrt]	[auto, sqrt, None]
max_depth	[3, 6, ..., 33]	[3, 6, ..., 33]	[3, 6, ..., 33]
min_samples_split	[2, 5, 10]	[2, 5, 10]	[2, 5, 10]
min_samples_leaf	[1, 2, 4]	[1, 2, 4]	[1, 2, 4]
bootstrap	[True, False]	[True, False]	-
learning_rate	-	-	[0.1, 0.2, 0.5]

Table 5.9: Hyperparameter tuning - intervals for each hyperparameter

We have two different alternatives here; either using the GridSearchCV or using the RandomizedSearchCV. In our case, since we have assumed long intervals for each hyperparameter, using the GridSearchCV will not computationally be a good idea. The reason is that there will be too many combinations to be trained. Therefore, we use the RandomizedSearchCV method for each of these algorithms with the above set of hyperparameters. We consider 5 iterations for all three algorithms, each with the 5-fold cross-validation. It means that each algorithm will be trained $5 \times 5 = 25$ times, and the best score among all of those will be chosen. The following table summarizes the results:

Algorithm /Best Hyperparameter	**Multi-Output Random Forest**	**Multi-Output Extra Trees**	**Multi-Output Gradient Boosting**
n_estimators	400	400	1800
max_features	auto	auto	sqrt
max_depth	27	18	24
min_samples_split	2	5	10
min_samples_leaf	2	2	4
bootstrap	True	True	-
learning_rate	-	-	0.1
Score	0.8221	0.8067	0.8002

Table 5.10: Hyperparameter tuning - best scores parameters

Thus, based on the results, we will choose the random forest method with the suggested hyperparameters in Table 5.10. We then train the model with these settings with a 5-fold cross-validation and calculate the value of MAE which is 0.0528

5.4 Choosing the Best Model

Comparing the MAE of the random forest method (the best algorithm among the two other algorithms in the previous section) with the MAE derived from the neural network model (Table 5.8), we can see that the latter slightly excels the former one. Therefore, we can utilize the trained neural network method to predict the utilization in the specific system mentioned at the beginning of this chapter.

Now that we have built a successful model to predict the utilization, one of the essential parameters in loss systems, we will utilize different pricing strategies based on various queueing contexts and optimization goals. The next chapter focuses on a specific price optimization problem for an M/M/N/N loss system where N is larger than 1.

Chapter 6

Price Optimization in Modified Erlang-Loss System

We begin this chapter by first describing the specific systems for which we want to find the optimal price. We then build another ML model to find the profit of those aforementioned loss systems. Finally, we state our proposed algorithm to find the optimal price.

6.1 Assumptions

Throughout the whole chapter, we will follow these assumptions: **1) Advance Reservation and MNL Choice Model** - the same as the previous chapter, we include these two extensions, e.g., customers can reserve in advance for the next τ days (on average) and they will choose among the available servers based on their utility function, U_j^i, equation (4.3), **2) Price Equality** - prices are all set equal together, e.g., $P_j = P, \forall j \in \{1, 2, ..., N\}$; in fact, inspired by Zhu *et al.* (2019b), we assume that a third party could be the decision maker for this system and he could be the one who sets all the pricing strategies with a goal of maximizing the whole system's revenue; however, we should mention that this pricing tool does not dictate any prices but it only recommends prices and in our problem, we are assuming all the servers are going to choose the offered price, **3) Linear Demand** almost the same as our assumption in Chapter 3 for the demand rate, $\lambda = D - P$ where D is some constant and not necessarily is equal to 1 and P is the price of each server. The other assumptions will be the same as what we stated in the previous chapter.

From now on, we will refer to loss systems with the above assumptions as equally priced modified Erlang-loss systems. In the following section, we will work on building ML models to find the optimal pricing strategy.

6.2 Predicting the Profit in Equally Priced Modified Erlang-Loss System

The same as how we defined the profit equation in Chapter 3, equation (3.1), we define the modified profit function as below:

$$\pi = \bar{\rho}_e \times P \times T \tag{6.1}$$

Where T is the time during which the simulation has been run, i.e., 365 days in our case (indicating a year) and $\bar{\rho}_e$, is the average effective utilization of the queueing system derived from the simulation (since we do not know the exact value).

As mentioned previously, due to computational barriers for using our simulation in large problems, we utilized machine learning to predict the utilization instead of calculating it directly. Here, the same thing will be true for finding the utilization in an equally priced system too. Assuming the demand rate for the system is $\lambda = D - P$, if we are dealing with large values of D such as $D = 100$ or $D = 1000$, to be able to find the price leading us to the optimal profit, we need to try large values of λ in the simulation. However, according to our earlier discussion, it will be time-consuming for our simulation to run those instances. Therefore, we still need an ML model to predict the utilization in the first step and then find the profit in the next one.

To this end, we will create another dataset, the same as what we made before (Table 5.6) with the same range of intervals for parameters. The only difference is that we set the prices all equal together, meaning that the first price will be chosen randomly from the range of 50 to 500, and the other servers' price will be the same as the first one. Moreover, there will be no outside option in the model either due to the assumption of $\lambda = D - P$. In fact, the outside option is already modeled in the demand function in this new model; when the price is set higher, lambda will decrease under this demand function, which means that more customers are pushed to the outside option. It is also good to mention that now, the label in the dataset is the average system utilization, and it is a single value rather than a five-tuple that we used to deal with in the previous chapter.

For training the model, since we have already worked with quite the same dataset and the neural network method outperformed, in this chapter, we only train the models with neural network algorithm and then utilize the *Keras Tuner* for hyper-parameter tuning. We have run this tuning process for 20 trials and each trial for 2 executions. We then put the objective function of our network to minimize the validation set's MAE. We also assumed that the input layer has 64 neurons (the same as the previous chapter), and the output layer has the size of 1 as it is supposed to predict only a single value. For hidden layers, we randomly chose from 2 to 5 number of hidden layers in each trial. Moreover, each hidden layer will have a random number of neurons within a range of 32 to 512 with the step of 32. All the activation functions are sigmoid, and the optimizer is RMSprop with the learning rate of 0.001. Moreover, we set the default value of epochs equal to 1000; however, to prevent overfitting, we used the callback of early stopping function with the patience of 10 to terminate the learning if considerable improvement has not been noticed. Figure 6.1 shows the best trained network structure in this hyperparameter tuning session.

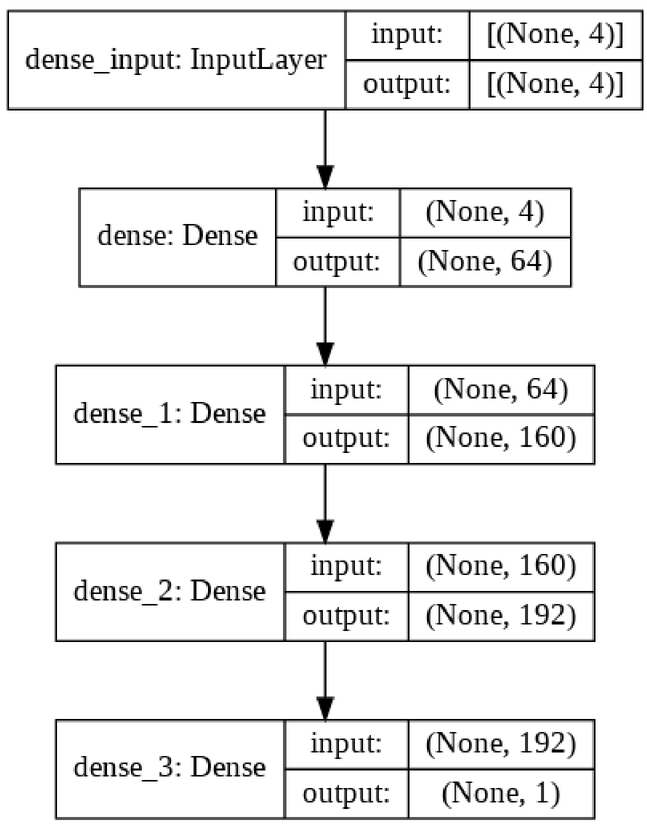


Figure 6.1: The best hypermodel derived from the keras tuner library; a model for predicting the utilization in an equally priced modified Erlang-loss system

Moreover, the errors are as follows:

Loss	**MAE**	**MSE**
0.0074	0.0074	1.2165×10^{-4}

Table 6.1: The errors for the test set; hypermodel for predicting the utilization in an equally priced modified Erlang-loss system

6.3 Price Optimization: Ternary Search

Now that we have a trained ML model which works acceptably well, we can utilize this model and run a ternary search algorithm to find the optimal price (the price which leads to the highest possible profit) inspired by Grigoriadis, M. D., & Khachiyan (1994) who proposed the ternary search algorithm as one of the fast approximation methods for solving convex optimization problems. Based on Bajwa *et al.* (2015), time complexity in this method is $O(\log_3 n)$ which is fairly acceptable in our case.

For this purpose, according to the definition of the profit function in equation (6.1), and according to the fact that the increase in price reduces the demand (arrival rate) as in $\lambda = D - P$, there will be a maximum point for the profit. Our goal here is to find that optimal point; hence, we define the modified ternary search algorithm to be compatible with our ML model as stated in Algorithm 2.

In Algorithm 2, $\hat{f}$ is the trained ML model to predict the utilization, and ε^* is the absolute precision of the ternary search.

Now, we can compare the optimal price and subsequently, the optimal profit in the equally priced modified Erlang-loss system with the optimal profit in the equally

Algorithm 2 Ternary Search Algorithm

input: $(\lambda_{now}, \lambda_{reserve}, \tau, \mu, P, T, \varepsilon^*, D, \kappa, \hat{f})$
output: P^*
$P_L \leftarrow 0$, $P_R \leftarrow D$
while TRUE **do**
    if $|P_L - P_R| < \varepsilon^*$ **then**
        return $P^* \leftarrow \frac{P_L+P_R}{2}$
    end if
    $P_{L/3} \leftarrow \frac{2P_L+P_R}{3}$, $P_{R/3} \leftarrow \frac{2P_R+P_L}{3}$
    $f_{L/3} \leftarrow \hat{f}(\kappa, \frac{1}{\mu}, D - P_{L/3}) \times P_{L/3} \times T$
    $f_{R/3} \leftarrow \hat{f}(\kappa, \frac{1}{\mu}, D - P_{R/3}) \times P_{R/3} \times T$
    if $f_{L/3} < f_{R/3}$ **then**
        $P_R \leftarrow P_{R/3}$
    else
        $P_L \leftarrow P_{L/3}$
    end if
end while

priced classic Erlang-loss system and investigate the impacts of the advance reservation and MNL choice model on pricing strategies.

6.4 Comparing the Optimal Price and Profit

To compare the optimal price and profit in these two aforementioned systems, we can utilize Algorithm 2 for the equally priced Erlang-loss system too. The only difference will be the function of $\hat{f}$; we have to use the Erlang-loss formula for finding the effective utilization, e.g., equation (4.5).

6.4.1 Comparison of the Profit and Price Value: κ Effect

In this part, we tried to find the optimal profit and the corresponding price based on different values of D in the classic Erlang-loss system, and then, we compared the results with two different scenarios: 1) $\tau = 7$ and $\kappa = 0.1$, e.g., 10% of the arrival rate are the ones who seek for service upon their arrival and the other 90% are for advance reservation; 2) $\tau = 7$ and $\kappa = 0.9$. For all three scenarios, we assumed $\mu = 1$ and $T = 365$, e.g., we are interested in comparing the profits gained in a year. We then plotted the yearly optimal profit in terms of D for these three scenarios:

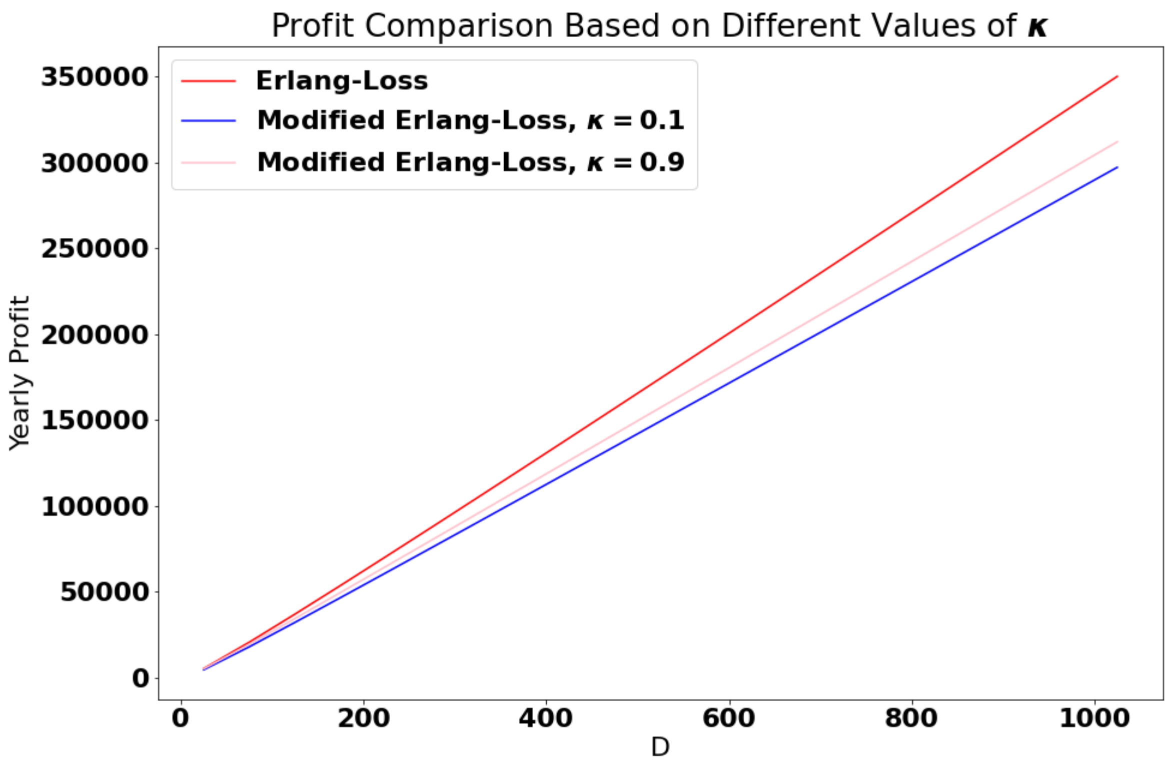


Figure 6.2: The comparison of the yearly optimal profit in modified Erlang-loss system with the Erlang-loss system based on two different values of κ

A more detailed summary can be found in Table 6.2.

D	P^*	π^*	$P^*_{\kappa=0.1}$	Absolute Profit Loss $\kappa = 0.1$	Relative Profit Loss $\kappa = 0.1$	$P^*_{\kappa=0.9}$	Absolute Profit Loss $\kappa = 0.9$	Relative Profit Loss $\kappa = 0.9$
25	18.79	5325.382	18.93	737.1098	0.1384	19.24	158.1367	0.0297
75	64.53	20699.5302	64.17	2708.8845	0.1309	65.91	1191.1576	0.0575
125	111.79	37037.0862	112.09	4819.4855	0.1301	114.19	2554.8635	0.069
175	159.61	53750.6409	160.75	7080.9387	0.1317	163.03	4098.5529	0.0763
225	207.76	70680.1759	209.77	9452.3454	0.1337	212.16	5761.4336	0.0815
275	256.11	87754.4235	258.99	11906.5279	0.1357	261.45	7511.1612	0.0856
325	304.62	104934.3134	308.34	14425.6591	0.1375	310.86	9328.0154	0.0889
375	353.24	122195.6943	357.79	16997.452	0.1391	360.33	11198.878	0.0916
425	401.96	139522.4068	407.3	19613.0646	0.1406	409.9	13114.4927	0.094
475	450.75	156903.011	456.87	22265.8818	0.1419	459.49	15068.0408	0.096
525	499.61	174329.0576	506.49	24950.8312	0.1431	509.13	17054.2882	0.0978
575	548.52	191794.0968	556.12	27663.8725	0.1442	558.82	19069.1589	0.0994
625	597.48	209293.0724	605.8	30401.7959	0.1453	608.51	21109.3254	0.1009
675	646.49	226821.9345	655.48	33161.914	0.1462	658.25	23172.1318	0.1022
725	695.53	244377.379	705.22	35942.065	0.1471	707.98	25255.3144	0.1033
775	744.6	261956.669	754.96	38740.3225	0.1479	757.75	27356.9863	0.1044
825	793.7	279557.5069	804.69	41555.1318	0.1486	807.52	29475.5838	0.1054
875	842.84	297177.9418	854.46	44385.1176	0.1494	857.31	31609.6712	0.1064
925	891.99	314816.3005	904.25	47229.1438	0.15	907.12	33758.0796	0.1072
975	941.17	332471.135	954.02	50086.0936	0.1506	956.93	35919.7857	0.108
1025	990.37	350141.1822	1003.83	52955.0886	0.1512	1006.77	38093.7936	0.1088

Table 6.2: Summary of the comparison of optimal profit and price: κ effect

Based on Table 6.2, we can see that the absolute profit loss and, subsequently, the relative profit loss will increase when κ decreases; this validates our observations in previous chapters. When κ decreases, the advance reservation will appear more often in the system, and thus the profit received will drop. In Figure 6.2, one can observe that the blue line lies below the pink line and conveys the same conclusion. Moreover, the difference between the red line (Erlang-loss system) and the pink line (modified Erlang-loss) comes from two reasons: a small effect of advance reservation (since κ is comparable high) and also the existence of the MNL choice model. Moreover, from Table 6.2, we can also notice that the optimal price happens in a larger value for the time when we consider the MNL choice model and advance reservation. With the increase in κ, the optimal price gets higher while the profit will be smaller.

6.4.2 Comparison of the Profit and Price Value: τ Effect

In this part, we assumed two different scenarios: 1) $\tau = 1$, and 2) $\tau = 20$ while in both scenarios we assumed $\kappa = 0.5$ and the other parameters are the same as before. In the following figure, we showed the impacts of τ in the profit loss comparing to the optimal profit received in the Erlang-loss system.

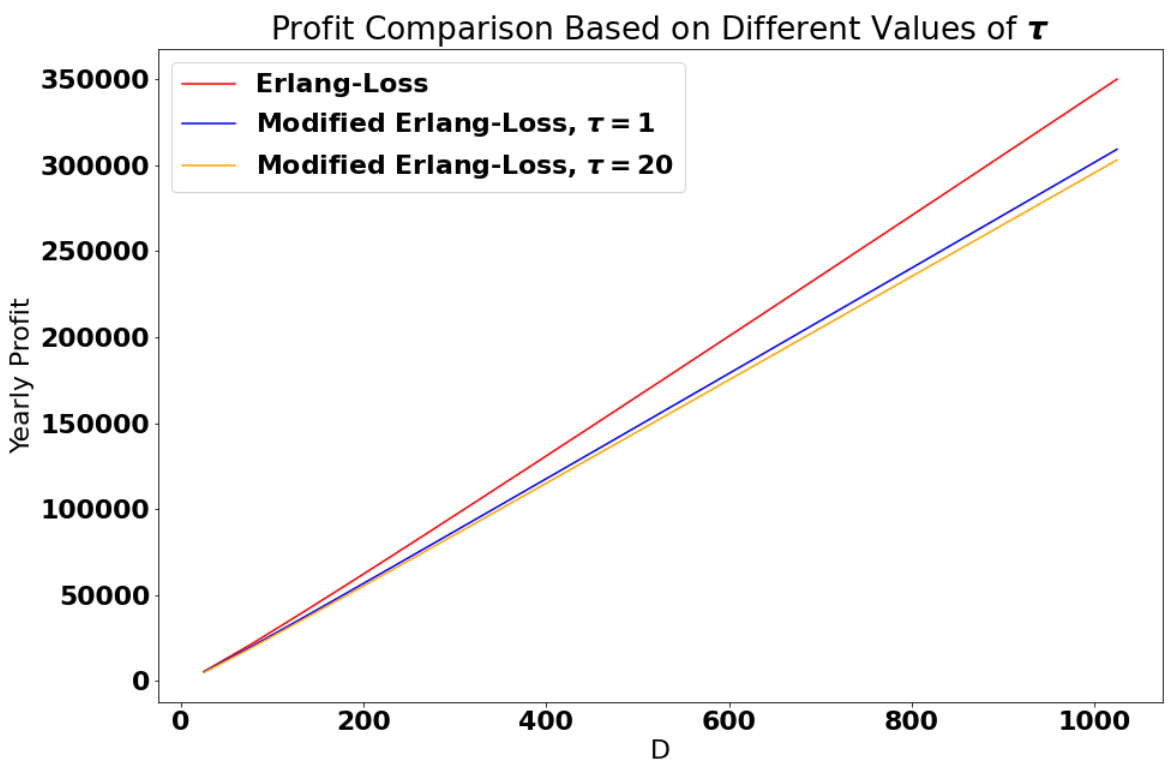


Figure 6.3: The comparison of the optimal profit in modified Erlang-loss system based on two different values of τ with the Erlang-loss system

As can be seen in Figure 6.3, the difference of profit loss between different values of τ is less than what we observed in Figure 6.2 for the values of κ. This validates our previous observation that τ has lower effects on the utilization of the system, and hence, lower impacts on the profit. In the following table, we showed more details of these two scenarios comparing to the Erlang-loss system.

D	P^*	π^*	$P^*_{\tau=20}$	Absolute Profit Loss $\tau = 20$	Relative Profit Loss $\tau = 20$	$P^*_{\tau=1}$	Absolute Profit Loss $\tau = 1$	Relative Profit Loss $\tau = 1$
25	18.79	5325.382	18.97	534.4759	0.1004	19.41	189.2917	0.0355
75	64.53	20699.5302	64.75	2133.5199	0.1031	65.9	1400.026	0.0676
125	111.79	37037.0862	112.8	3936.2197	0.1063	114.09	2913.6346	0.0787
175	159.61	53750.6409	161.55	5900.9718	0.1098	162.86	4597.4422	0.0855
225	207.76	70680.1759	210.64	7981.094	0.1129	211.95	6396.1599	0.0905
275	256.11	87754.4235	259.92	10147.602	0.1156	261.2	8279.5719	0.0943
325	304.62	104934.3134	309.32	12381.8034	0.118	310.59	10228.9591	0.0975
375	353.24	122195.6943	358.82	14670.9023	0.1201	360.05	12231.7218	0.1001
425	401.96	139522.4068	408.39	17005.719	0.1219	409.6	14278.8785	0.1023
475	450.75	156903.011	458.0	19379.3953	0.1235	459.21	16363.79	0.1043
525	499.61	174329.0576	507.67	21786.6414	0.125	508.83	18481.317	0.106
575	548.52	191794.0968	557.36	24223.2951	0.1263	558.5	20627.4443	0.1075
625	597.48	209293.0724	607.08	26686.0109	0.1275	608.22	22798.9162	0.1089
675	646.49	226821.9345	656.83	29172.0173	0.1286	657.94	24993.0368	0.1102
725	695.53	244377.379	706.59	31679.0352	0.1296	707.69	27207.6185	0.1113
775	744.6	261956.669	756.37	34205.101	0.1306	757.46	29440.8062	0.1124
825	793.7	279557.5069	806.19	36748.594	0.1315	807.25	31690.9516	0.1134
875	842.84	297177.9418	856.0	39308.0806	0.1323	857.04	33956.7293	0.1143
925	891.99	314816.3005	905.82	41882.3154	0.133	906.84	36236.8719	0.1151
975	941.17	332471.135	955.67	44470.231	0.1338	956.66	38530.3838	0.1159
1025	990.37	350141.1822	1005.51	47070.8424	0.1344	1006.5	40836.3395	0.1166

Table 6.3: Summary of the comparison of optimal profit and price: τ effect

The summary in Table 6.3 also validates our previous observation. Also, comparing to the results of Table 6.2, we can see that the reduction in profit is comparably lower. To illustrate this idea better, we showed all the relative errors in the following figure:

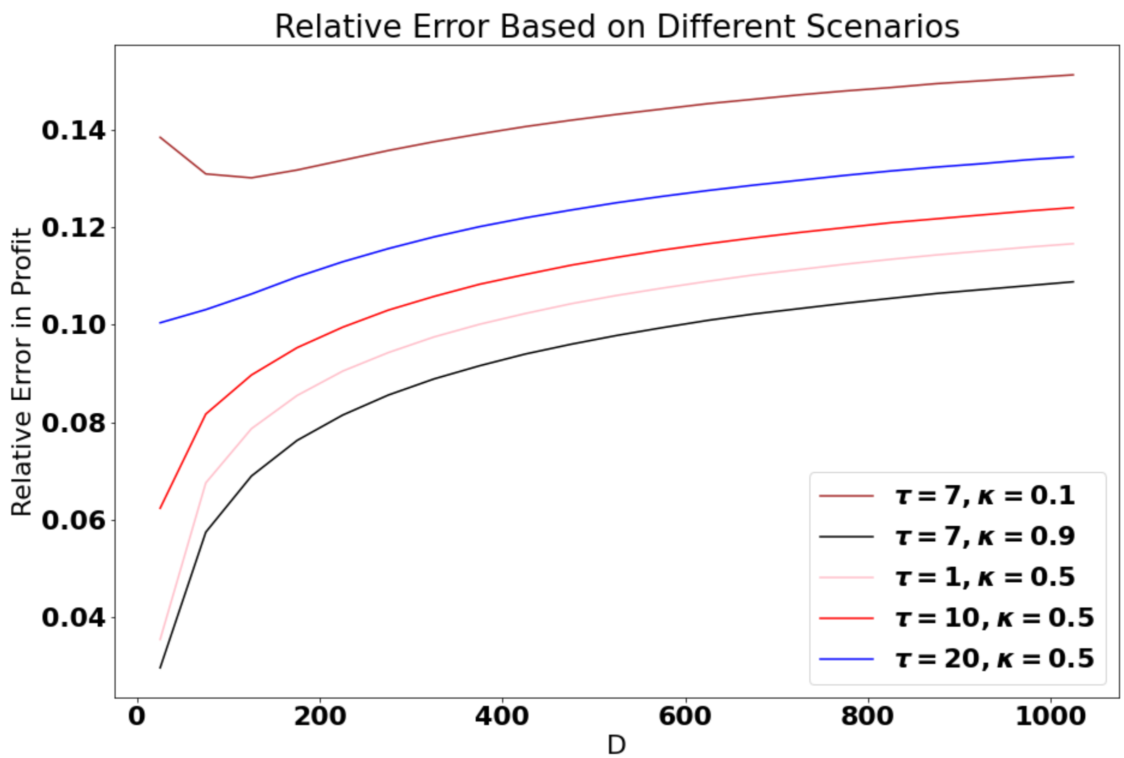


Figure 6.4: The relative error of profit loss in modified Erlang-loss system based on different values of κ and τ

As can be seen in Figure 6.4, the relative errors generally have the same pattern; they will increase when D increases. However, the slope gets smoother. Also, regarding the brown line, the reduction in the relative error for some small values of D may come from errors in the prediction of the value of utilization. Nonetheless, in larger

D values, it follows the same pattern. Finally, as can be noticed, the levels of the lines' occurrence mainly depend on the value of κ rather than τ which again indicates the higher impacts of κ than τ.

6.5 Final Remarks

Based on the conclusions that we had on the impacts of κ and τ parameters on the utilization and profit, we can state that for some companies like Zipcar where riders can reserve a car up to 14 days ahead (Zipcar Support, 2021), although the company is giving the customers more satisfaction by increasing the value of τ, it will lose some profit. On the other hand, as earlier mentioned, there is no option of removing the advance reservation in such systems meaning that κ is always almost equal to 1 there. Thus, the alternative of reducing κ is not possible here, which is close to the case of short-term rental systems. κ is comparably higher in short-term rentals than a system such as a ride-sharing platform; therefore, the focus should be on limiting the value of τ and increasing the price for the customers who want to reserve for a relatively long time ahead.

For ride-sharing companies, the reason why there is a difference between the τ value in Uber and Lyft, as an example, could be decreasing the value of τ to gain more profit in Lyft's point of view. On the other hand, both these companies can charge more if a user wants to reserve for a longer time in advance; with this policy, both τ and κ tend to drop while the system is still supporting advance reservation for their service. Finally, based on the relative profit loss that was depicted in Figure 6.4, the price for the reserved service should be higher than the price for non-reserved service to obtain the optimal profit.

Chapter 7

Conclusion

This thesis presents a price optimization analysis in stochastic loss systems while considering other features such as scheduling in advance or the more sophisticated customer choice model. Accordingly, the proposed simulation can be applied for various loss systems with different parameters and results in the utilization of each server. Furthermore, the utilization values can help any analyst set their goal and optimize their problem by correctly identifying their decision variable. Finally, different ML models can enhance the computations speed and lead to the final answers in a much shorter amount of time.

Our results in the first chapter showed that among the two different tax policies that government imposes for short-term rentals, the vacancy and speculation tax policy gives more flexibility to the regulator. In other words, with this policy, the government can control the hosts' pricing strategies and control the supply for different streams of housing systems. However, this is not the case in the MAT tax, where the government cannot encourage the hosts to change their offered price. We also showed that speculation and vacancy tax already include the MAT tax under specific circumstances. As a result, we concluded that the government or the regulator should use the vacancy tax (the tax based on the utilization).

On the other hand, from the hosts' point of view, we showed that although some policies can make the hosts change their offered price, they can still choose short-term rentals over long-term rentals. However, there will be some cases that offering their unit in the short-term rental will lead to less profit than long-term rentals. Based on the summary that we illustrated at the end of this chapter, one can choose the best possible price based on different values of parameters.

We then proposed a simulation for larger a larger scale, modified M/M/N/N Erlang-loss systems, including advance reservation and Multinomial Logit (MNL) choice model. The development of this simulation let us run numerical experiments based on which one can realize that there are two parameters in the advance reservation that impact the utilization of the system: the arrival rate of the customers who reserve in advance and the average number of days they reserve. Both these factors reduce the utilization while the impacts of the former are much more noticeable.

Finally, different Machine Learning (ML) models helped us reach our goals in a shorter time while bearing a little error value in their prediction. For example, the price optimization for an M/M/5/5 system shows that the optimal profit in a modified Erlang-loss system happens with a higher offered price than the classic Erlang-loss system. In contrast, the profit itself will be reduced in the former.

There are various ways to extend this current work for future research. One could be focusing on using general distribution for the arrival process and the service time of the customers entering the system. It will make the model more applicable for real-world uses. The other way would be training an ML model with a broader range of input data and more servers. With the help of more advanced computing techniques, using more recent simulation environments, and devoting more time to compile the simulation, one can extend the range for each queueing parameter to include more problems. Lastly, for finding the optimal price in price optimization part of this thesis, one can implement more robust methods such as stochastic gradient descent instead of ternary search algorithm based on the suggestion of Bertsimas and De Boer (2005).

Appendix: Proofs

Lemma 3.1

In the M/M/1/1 model loss system when there is no regulation, the profit function is concave and it has one maximum over the feasible values of the offered price; that happens when the server chooses $P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu(\mu+1)}$ *to offer. The optimal profit (the profit based on* $P^*_{S_\alpha}$*) will be* $\pi^*_{S_\alpha} = 2\mu + 1 - 2\sqrt{\mu(\mu+1)} - u$*.*

Proof. We first start the proof with reviewing some queueing theory principles. Denoting the steady-state probability of the two states in M/M/1/1, e.g., either the server is idle or is occupied, by Π_0 and Π_1, respectively, we can draw the birth-death diagram for this system as the following:

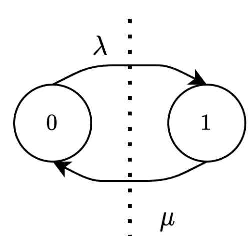


Figure 7.1: Birth-Death transition diagram in M/M/1/1 loss system

In Figure 7.1, there should be a balance on the dotted line in the steady-state;

hence: $\lambda\Pi_0 = \mu\Pi_1$. Therefore:

$$\Pi_0 + \Pi_1 = 1 \rightarrow \Pi_0 + \frac{\lambda}{\mu}\Pi_0 = 1 \rightarrow \Pi_0 = \frac{\mu}{\lambda + \mu} \rightarrow \begin{cases} \Pi_0 = \frac{1}{1+\rho} \\ \Pi_1 = \frac{\rho}{1+\rho} \end{cases} \tag{7.1}$$

From which the effective utilization can be derived:

$$\rho_e = (1 - \Pi_1)\,\rho = \Pi_0\rho = \frac{\rho}{1+\rho} = \frac{\lambda}{\lambda + \mu}$$

Getting back to the profit function, the critical point can be derived by taking a derivative from it in terms of the price variable and put it equal to zero:

$$\begin{aligned} &\tfrac{d\pi_{S_\alpha}}{dP_S} = \tfrac{d}{dP_S}\left(\rho_e P_S - u\right) = 0 \rightarrow \tfrac{d}{dP_S}\left(\tfrac{P_S - {P_S}^2}{\mu+1-P_S}\right) = 0 \\ &\rightarrow \left(\tfrac{(1-2P_S)(\mu+1-P_S)+\left(P_S-{P_S}^2\right)}{(\mu+1-P_S)^2}\right) = 0 \end{aligned} \tag{7.2}$$

Solving the equation above, we get two results for P_S. The optimal price value is either equal to $\mu + 1 + \sqrt{\mu\,(\mu+1)}$ or $\mu + 1 - \sqrt{\mu\,(\mu+1)}$. However, the first root is bigger than 1 which is not feasible in our model (the arrival rate is defined as $\lambda = 1 - P_S$; therefore, the largest possible amount for P_S is 1 which results in zero arrival rate for short-term rentals). As a result, the only critical point over the feasible range of price variables will be the latter. On the other hand, in order to show this point is a maximum, we have to show that the function is concave, e.g., its second derivative in terms of the price variable is negative over the feasible values.

Thus:

$$\frac{d^2\pi_{S_\alpha}}{dP_S{}^2} = \frac{d}{dP_S}\left(\frac{(1-2P_S)(\mu+1-P_S)+\left(P_\alpha-P_S{}^2\right)}{(\mu+1-P_S)^2}\right) =$$
$$\frac{((-2P_S(\mu+1-P_S)-(1-2P_S))+(1-2P_S))(\mu+1-P_S)^2-(-2(\mu+1-P_S))\left((1-2P_S)(\mu+1-P_S)+\left(P_S-P_S{}^2\right)\right)}{(\mu+1-P_S)^4} \rightarrow$$
$$\frac{d^2\pi_{S_\alpha}}{dP_S{}^2} = -\frac{2\mu(\mu+1)}{(\mu+1-P_S)^3} < 0 \tag{7.3}$$

Above equation shows that the profit function is concave and hence, $P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu\,(\mu+1)}$ will lead to a maximum profit for the host. The corresponding profit based on this price will be as follows:

$$\begin{aligned}
&\pi^*_{S_\alpha} = \rho_e P^*_{S_\alpha} - u \rightarrow \pi^*_{S_\alpha} = \left(\frac{1-P^*_{S_\alpha}}{\mu+1-P^*_{S_\alpha}}\right) P^*_{S_\alpha} - u \rightarrow \\
&\pi^*_{S_\alpha} = \left(\frac{1-\left(\mu+1-\sqrt{\mu(\mu+1)}\right)}{\mu+1-\left(\mu+1-\sqrt{\mu(\mu+1)}\right)}\right)\left(\mu+1-\sqrt{\mu\,(\mu+1)}\right) - u \rightarrow \\
&\pi^*_{S_\alpha} = \left(\frac{\sqrt{\mu(\mu+1)}-\mu}{\sqrt{\mu(\mu+1)}}\right)\left(\mu+1-\sqrt{\mu\,(\mu+1)}\right) - u \rightarrow \\
&\pi^*_{S_\alpha} = \frac{(\mu+1)\sqrt{\mu(\mu+1)}-\mu(\mu+1)-\mu(\mu+1)+\mu\sqrt{\mu(\mu+1)}}{\sqrt{\mu(\mu+1)}} - u \rightarrow \\
&\pi^*_{S_\alpha} = \frac{(2\mu+1)\sqrt{\mu(\mu+1)}-2\mu(\mu+1)}{\sqrt{\mu(\mu+1)}} - u \rightarrow \pi^*_{S_\alpha} = 2\mu+1-2\sqrt{\mu\,(\mu+1)} - u
\end{aligned} \tag{7.4}$$

□

Lemma 3.2

In the M/M/1/1 model loss system when there is regulation by a fixed amount of tax, the profit function is concave and it has one maximum over the feasible values of the offered price; that happens when the server chooses $P^*_{S_\beta} = P^*_{S_\alpha} = \mu+1-\sqrt{\mu\,(\mu+1)}$ to offer. The optimal profit (the profit based on $P^*_{S_\beta}$) will be $\pi^*_{S_\beta} = 2\mu+1-2\sqrt{\mu\,(\mu+1)}-$

$u - \Delta$ *or in other words, it will be* $\pi^*_{S_\beta} = \pi^*_{S_\alpha} - \Delta$.

Proof. Based on *Lemma 3.1*, the profit function in Model II is exactly the same but also includes the fixed cost parameter which is a constant. Therefore, it does not appear in any derivatives, and the final solutions will not change. Therefore, the same price as what stated in *Lemma 3.1* maximize the profit function. □

Lemma 3.3

In the M/M/1/1 model loss system when there is a tax based on utilization, if a server is willing not to pay the vacancy tax, she should set her price in a way that $P_S \leq P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}$

Proof. Assuming the threshold is θ, the server, in order not to pay the vacancy tax, should have her unit's utilization higher than θ.

$$\rho_e \geq \theta \Leftrightarrow \frac{\rho}{1+\rho} \geq \theta \Leftrightarrow \frac{\frac{\lambda}{\mu}}{1+\frac{\lambda}{\mu}} \geq \theta \Leftrightarrow \frac{\frac{\lambda}{\mu}}{\frac{\mu+\lambda}{\mu}} \geq \theta \Leftrightarrow \frac{\lambda}{\mu+\lambda} \geq \theta \Leftrightarrow \lambda \geq (\mu+\lambda)\,\theta \Leftrightarrow$$
$$\lambda\,(1-\theta) \geq \mu\theta \Leftrightarrow 1 - P_S \geq \frac{\mu\theta}{1-\theta} \Leftrightarrow P_S \leq 1 - \frac{\mu\theta}{1-\theta} \Leftrightarrow P_S \leq \frac{1-\theta-\mu\theta}{1-\theta} \Leftrightarrow P_S \leq \frac{1-\theta(1+\mu)}{1-\theta} \tag{7.5}$$

We can learn from above that if the price offered by the server is smaller than or equal to $P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}$, the utilization will be higher than the threshold meaning that there will be no need to pay the vacancy tax. □

Lemma 3.4

In the single-host model when there is a tax based on utilization, if the utilization threshold is smaller than or equal to $1 - \sqrt{\frac{\mu}{\mu+1}}$, *the best strategy will be offering* $P^*_{S_\gamma} =$

$P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu(\mu+1)}$ *to gain the maximum profit. Under these conditions, the host does not need to pay the vacancy tax either. However, if* $\theta > 1 - \sqrt{\frac{\mu}{\mu+1}}$ *then* $P^*_{S_\alpha} > P^\theta_{S_\gamma}$.

Proof. Based on the results in *Lemma 3.3*, in Model III, if the host offers a price smaller than or equal to $P^\theta_{S_\gamma}$, she does not need to pay the vacancy tax. On the other hand, based on *Lemma 3.1*, the profit function in Model I is concave. Therefore, if $P^*_{S_\alpha}$ happens to be smaller than or equal to $P^\theta_{S_\gamma}$, the host gains the maximum profit, e.g., the results of Model I and Model III will be the same. As a result:

$$\begin{aligned}&\mu+1-\sqrt{\mu(\mu+1)} \le \tfrac{1-\theta(\mu+1)}{1-\theta} \Leftrightarrow (1-\theta)(\mu+1) - (1-\theta)\sqrt{\mu(\mu+1)} \le 1-\theta(\mu+1) \Leftrightarrow \\ &\mu+1-(1-\theta)\sqrt{\mu(\mu+1)} \le 1 \Leftrightarrow \mu \le (1-\theta)\sqrt{\mu(\mu+1)} \Leftrightarrow \tfrac{\mu^2}{\mu(\mu+1)} \le (1-\theta)^2 \Leftrightarrow \\ &\sqrt{\tfrac{\mu}{\mu+1}} \le 1-\theta \Leftrightarrow \theta \le 1-\sqrt{\tfrac{\mu}{\mu+1}}\end{aligned} \tag{7.6}$$

All the steps are reversible; hence, we can conclude that if the very last term in the above inequality holds true, e.g., $\theta \le 1 - \sqrt{\frac{\mu}{\mu+1}}$, the host has to offer $P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu(\mu+1)}$, she will gain the maximum profit without the need to pay the vacancy tax.

□

Proposition 3.1

In the M/M/1/1 loss system when there is a tax based on utilization, the optimal price function (e.g., the price which leads to the best possible profit) and the best possible profit gained by the server based on different values of θ *and* δ *will be as below:*

The optimal price function:

$$P^*_{S_\gamma} = \begin{cases} P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}, & \theta > 1 - \sqrt{\frac{\mu}{\mu+1}}, \quad \textit{and} \\ & \delta > 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta-\theta^2(1+\mu)}{1-\theta} \\ P^*_{S_\alpha} = \mu + 1 - \sqrt{\mu\left(\mu+1\right)}, & \textit{otherwise} \end{cases} \tag{7.7}$$

The best possible profit gained by the server:

$$\pi^*_{S_\gamma} = \begin{cases} 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - u, & \theta \leq 1 - \sqrt{\frac{\mu}{\mu+1}} \\ \pi^{'}, & \theta > 1 - \sqrt{\frac{\mu}{\mu+1}} \end{cases} \tag{7.8}$$

where:

$$\pi^{'} = \begin{cases} 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - u - \delta, & \delta \leq 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta-\theta^2(1+\mu)}{1-\theta} \\ \theta\left(\frac{1-\theta(1+\mu)}{1-\theta}\right) - u, & \delta > 2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta-\theta^2(1+\mu)}{1-\theta} \end{cases}$$

Proof. The proof comes directly from the results in *Lemma 3.4* and *Lemma 3.5*. Based on *Lemma 3.4*, if $\theta \leq 1 - \sqrt{\frac{\mu}{\mu+1}}$, the optimal price will be the same as the one in the benchmark model. On the other hand, based on *Lemma 3.5*, if both conditions of $\theta > 1 - \sqrt{\frac{\mu}{\mu+1}}$ and $\delta > \left(2\mu + 1 - 2\sqrt{\mu\left(\mu+1\right)} - \frac{\theta-\theta^2(1+\mu)}{1-\theta}\right)$ are met, the optimal price will be less than the one in the benchmark model and is equal to $P^{\theta}_{S_\gamma} = \frac{1-\theta(1+\mu)}{1-\theta}$. The optimal profit will be calculated correspondingly.

□

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., and Others (2015). Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems, Software. *available from tensorflow. org*.

Abbasi, B., Babaei, T., Hosseinifard, Z., Smith-Miles, K., and Dehghani, M. (2020). Predicting solutions of large-scale optimization problems via machine learning: A case study in blood supply chain management. *Computers and Operations Research*, **119**, 104941.

Afeche, P., Liu, Z., and Maglaras, C. (2018). Ride-Hailing Networks with Strategic Drivers: The Impact of Platform Control Capabilities on Performance. *SSRN Electronic Journal*, pages 1–56.

Ammirato, S., Felicetti, A. M., Linzalone, R., Volpentesta, A. P., and Schiuma, G. (2020). A systematic literature review of revenue management in passenger transportation. *Measuring Business Excellence*, **24**(2), 223–242.

Australian Government (2021). Vacancy fee for foreign owners.

Bajwa, M. S., Agarwal, A. P., and Manchanda, S. (2015). Ternary search algorithm:

Improvement of binary search. *2015 International Conference on Computing for Sustainable Global Development, INDIACom 2015*, pages 1723–1725.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, **13**, 281–305.

Bertsimas, D. and De Boer, S. (2005). Simulation-based booking limits for airline revenue management. *Operations Research*, **53**(1), 90–106.

Better Dwelling (2017). Vacant Homes Are A Global Epidemic, And Paris Is Fighting It With A 60% Tax. *Better Dwelling*.

Bilali, A., Dandl, F., Fastenrath, U., and Bogenberger, K. (2019). An Analytical Model for On-Demand Ride Sharing to Evaluate the Impact of Reservation, Detour and Maximum Waiting Time. *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pages 1715–1720.

Bisong, E. (2019). *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress.

Bose, S. K. (2013). *An introduction to queueing systems*. Springer Science & Business Media.

Brumelle, S. (1978). Generalization of Erlang'S Loss System To State Dependent Arrival and Service Rates. *Mathematics of Operations Research*, **3**(1), 10–16.

Cachon, G. P. and Feldman, P. (2011). Pricing services subject to congestion: Charge per-use fees or sell subscriptions? *Manufacturing and Service Operations Management*, **13**(2), 244–260.

Chen, Y., Levi, R., and Shi, C. (2017). Revenue Management of Reusable Resources with Advanced Reservations. *Production and Operations Management*, **26**(5), 836–859.

Chong, V. (2020). Vacancy Tax In Malaysia: Should We Or Shouldn't We? *StarProperty*.

City of Toronto (2021). Short-Term Rental Operators/Hosts.

Corgozinho, M. C. B., Ferreira, V. A. S., and de Freitas Almeida, J. F. (2018). Scaling Hospital Laboratory Queues via Discrete Event Simulation Using Simpy. In *International Conference on Production and Operations Management Society*, pages 119–128. Springer.

Cowdrey, K. W., De Lange, J., Malekian, R., Wanneburg, J., and Jose, A. C. (2018). Applying queueing theory for the optimization of a banking model. *Journal of Internet Technology*, **19**(2), 381–389.

Cross, R. G., Higbie, J. A., and Cross, Z. N. (2011). Milestones in the application of analytical pricing and revenue management. *Journal of Revenue and Pricing Management*, **10**(1), 8–18.

Dunlop, J., Girma, D., and Irvine, J. (1999). *Digital mobile communications and the TETRA system*. John Wiley & Sons.

Erlang, A. K. (1917). Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Post Office Electrical Engineer's Journal*, **10**, 189–197.

Farhoodi, A. (2019). Welfare Estimation in Peer-to-Peer Markets with Heterogenous Agents: The Case of Airbnb. *SSRN Electronic Journal*, pages 1–44.

Garnett, O., Mandelbaum, A., & Reiman, M. I. (1998). Designing a telephone call center with impatient customers. *Manuf. Serv. Oper. Manag*, **4**.

Government of British Columbia (2018). Speculation and vacancy tax.

Government of British Columbia (2020). Frequently asked questions about the speculation and vacancy tax.

Government of Ontario (2017). O. Reg. 435/17: TRANSIENT ACCOMMODATION TAX.

Greenstein-Messica, A. and Rokach, L. (2020). Machine learning and operation research based method for promotion optimization of products with no price elasticity history. *Electronic Commerce Research and Applications*, **40**, 100914.

Grigoriadis, M. D., & Khachiyan, L. G. (1994). Fast approximation schemes for convex programs with many blocks and coupling constraints. *SIAM Journal on Optimization*, **4**(1), 86–107.

Gross, D. (2008). *Fundamentals of queueing theory*. John Wiley & Sons.

Guillet, B. D. and Mohammed, I. (2015). Revenue management research in hospitality and tourism: A critical review of current literature and suggestions for future research. *International Journal of Contemporary Hospitality Management*, **27**(4), 526–560.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., and Others (2020). Array programming with NumPy. *Nature*, **585**(7825), 357–362.

Hassin, R. and Koshman, A. (2017). Profit maximization in the M/M/1 queue. *Operations Research Letters*, **45**(5), 436–441.

Haviv, M. and Randhawa, R. S. (2014). Pricing in queues without demand information. *Manufacturing and Service Operations Management*, **16**(3), 401–411.

Helseth, A. and Sveen, E. B. (2020). Combining Machine Learning and Optimization for Efficient Price Forecasting. *International Conference on the European Energy Market, EEM*, **2020-Septe**, 0–4.

Hill, D. (2015). How much is your spare roomworth? *IEEE Spectrum*, **52**(9), 32–58.

Holden, L. (2017). *Inventory Optimization Using a SimPy Simulation Model*. Ph.D. thesis, East Tennessee State University.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, **9**(3), 90–95.

Insua, D., Ruggeri, F., and Wiper, M. (2012). *Bayesian analysis of stochastic process models*, volume 978. John Wiley & Sons.

Isguder, H. O. and Uzunoglu-Kocer, U. (2014). Analysis of GI/M/n/n queueing system with ordered entry and no waiting line. *Applied Mathematical Modelling*, **38**(3), 1024–1032.

Jacob, J. and Roet-Green, R. (2021). Ride solo or pool: Designing price-service menus for a ride-sharing platform. *European Journal of Operational Research*, **295**(3), 1008–1024.

Jain, R. (1990). *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons.

Jun, J. B., Jacobson, S. H., and Swisher, J. R. (1999). Application of discrete-event simulation in health care clinics: A survey. *Journal of the Operational Research Society*, **50**(2), 109–123.

Kambli, A., Sinha, A. A., and Srinivas, S. (2020). Improving campus dining operations using capacity and queue management: A simulation-based case study. *Journal of Hospitality and Tourism Management*, **43**(January), 62–70.

Kendall, D. G. . (1953). Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. *The Annals of Mathematical Statistics*, **24**(3), 338–354.

Kim, J. and Randhawa, R. S. (2018). The value of dynamic pricing in large queueing systems. *Operations Research*, **66**(2), 409–425.

Kingman, J. F. (2009). The first Erlang century-and the next. *Queueing Systems*, **63**(1), 3–12.

Lakshmi, C., & Iyer, S. A. (2013). Application of queueing theory in health care: A literature review. *Operations Research for Health Care*, **2**(1-2), 25–39.

Lyft Help (2017). Scheduled rides for riders.

Mariello, A., Dalcastagné, M., and Brunato, M. (2020). HotelSimu: Simulation-Based Optimization for Hotel Dynamic Pricing. In *International Conference on Learning and Intelligent Optimization*, pages 341–355. Springer.

Matloff, N. (2008). Introduction to discrete-event simulation and the simpy language. *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, **2**(2009), 1–33.

Medhi, J. (2002). Stochastic models in queueing theory. *Elsevier*.

Medhi, J. (2006). The evergreen erlang loss function. *Opsearch*, **43**(3), 309–319.

Mendelson, H., & Whang, S. (1990). Optimal incentive-compatible priority pricing for the M/M/1 queue. *Operations research*, **38**(5), 870–883.

Miller, B. L., & Buckman, A. G. (1987). Cost allocation and opportunity costs. *Management Science*, **33**(5), 626–639.

Naor, P. (1969). The Regulation of Queue Size by Levying Tolls. *The Econometric Society Stable*, **37**(1), 15–24.

Oliveira, B. B., Carravilla, M. A., and Oliveira, J. F. (2017). Fleet and revenue management in car rental companies: A literature review and an integrated conceptual framework. *Omega (United Kingdom)*, **71**, 11–26.

Palm, C. (1943). Intensitatsschwankungen im fernsprechverker. *Ericsson Technics*, **44**.

Palvannan, R. K. and Teow, K. L. (2012). Queueing for healthcare. *Journal of Medical Systems*, **36**(2), 541–547.

Paschalidis, I. C. and Tsitsiklis, J. N. (2000). Congestion-dependent pricing of network services. *IEEE/ACM Transactions on Networking*, **8**(2), 171–184.

Paschalidis, I. C., & Liu, Y. (2002). Pricing in multiservice loss networks: static pricing, asymptotic optimality and demand substitution effects. *IEEE/ACM Transactions On Networking*, **10**(3), 425–438.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **12**(85), 2825–2830.

Pedro Aznar, J., Maria Sayeras Maspera, J., and Quer, X. (2019). A game theory approach to Airbnb and hotels competition. *European Journal of Tourism Research*, **21**, 119–123.

Pender, J., Rand, R., and Wesson, E. (2020). A Stochastic Analysis of Queues with Customer Choice and Delayed Information. *Mathematics of Operations Research*, **45**(3), 1104–1126.

Petricek, M., Chalupa, S., and Melas, D. (2021). Model of price optimization as a part of hotel revenue management—stochastic approach. *Mathematics*, **9**(13).

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*, pages 35–39.

Rocca, R. (2021). Toronto city council approves plan to implement vacant home tax.

Segú, M. (2020). The impact of taxing vacancy on housing markets: Evidence from France. *Journal of Public Economics*, **185**, 104079.

Shajin, D. and Krishnamoorthy, A. (2021). On a queueing-inventory system with impatient customers, advanced reservation, cancellation, overbooking and common life time. *Operational Research*, **21**(2), 1229–1253.

Shajin, D., Krishnamoorthy, A., Dudin, A. N., Joshua, V. C., and Jacob, V. (2020). On a queueing-inventory system with advanced reservation and cancellation for the next K time frames ahead: the case of overbooking. *Queueing Systems*, **94**(1-2), 3–37.

Shortle, J. F., Thompson, J. M., Gross, D., & Harris, C. M. (2018). *Fundamentals of queueing theory*. John Wiley & Sons.

SimPy 4.0.1 Documentation (2020). SimPy: Discrete event simulation for Python.

Strauss, A. K., Klein, R., and Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*, **271**(2), 375–387.

Tekin, P. and Erol, R. (2017). A Literature Review of Revenue Management in Retail Sectors and the Categorization of Solution Methods. *Proceedings - UKSim-AMSS 2016: 10th European Modelling Symposium on Computer Modelling and Simulation*, **7**, 91–96.

Thomopoulos, N. T. (2012). *Fundamentals of queuing systems: statistical methods for analyzing queuing models*. Springer Science & Business Media.

Train, K. E. (2009). *Discrete choice methods with simulation, second edition*, volume 9780521766.

Uber Help (2016). Scheduling a ride in advance.

Van Woensel, T. and Vandaele, N. (2007). Modeling traffic flows with queueing models: A review. *Asia-Pacific Journal of Operational Research*, **24**(4), 435–461.

Vives, A., Jacob, M., and Payeras, M. (2018). Revenue management and price optimization techniques in the hotel sector: A critical literature review. *Tourism Economics*, **24**(6), 720–752.

Wang, K., Li, N., and Jiang, Z. (2010). Queueing system with impatient customers: A review. *Proceedings of 2010 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2010*, pages 82–87.

Wang, Y. B., Qian, C., & Cao, J. D. (2010). Optimized M/M/c model and simulation for bank queuing system. *2010 IEEE International Conference on Software Engineering and Service Sciences*, pages 474–477.

Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, **6**(60), 3021.

Xiao, H. and Zhang, G. (2010). The queuing theory application in bank service optimization. *2010 International Conference on Logistics Systems and Intelligent Management, ICLSIM 2010*, **2**(072300410110), 1097–1100.

Zhu, H., Gumus, M., and Ray, S. (2019a). Short-Term Housing Rentals and Corporatization of Platform Pricing. *SSRN Electronic Journal*.

Zhu, H., Gumus, M., and Ray, S. (2019b). Short-Term Housing Rentals and Corporatization of Platform Pricing. *SSRN Electronic Journal*.

Zipcar Support (2021). Reservation Timeframes.

Ziya, S., Ayhan, H., and Foley, R. D. (2006). Optimal prices for finite capacity queueing systems. *Operations Research Letters*, **34**(2), 214–218.

Ziya, S., Ayhan, H., and Foley, R. D. (2008). A note on optimal pricing for finite capacity queueing systems with multiple customer classes. *Naval Research Logistics*, **55**(5), 412–418.